Realizations and LP

Melvin Fitting
Dept. Mathematics and Computer Science
Lehman College (CUNY), 250 Bedford Park Boulevard West
Bronx, NY 10468-1589
e-mail: melvin.fitting@lehman.cuny.edu
web page: comet.lehman.cuny.edu/fitting

March 25, 2007

Abstract

LP can be seen as a logic of knowledge with justifications. Artemov's Realization Theorem says justifications can be extracted from validities in the more conventional Hintikka-style logic of knowledge S4, in which they are not explicitly present. Justifications, however, are far from unique. There are many ways of realizing each theorem of S4 in the logic LP. If the machinery of justifications is to be applied to artificial intelligence, or better yet, to everyday reasoning, we will need to work with whatever justifications we may have at hand—one version may not be interchangeable with another, even though they realize the same S4 formula. In this paper we begin the process of providing tools for reasoning about justifications directly. The tools are somewhat complex, but in retrospect this should not be surprising. Among other things, we provide machinery for combining two realizations of the same formula, and for replacing subformulas by equivalent subformulas. (The second of these is actually weaker than just stated, but this is not the place for a detailed formulation.) The results are algorithmic in nature—semantics for LP plays no role. We apply our results to provide a new algorithmic proof of Artemov's Realization Theorem itself.

1 Introduction

The logic LP (logic of proofs) was introduced by Artemov. It is a propositional modal-like logic that was created to help complete a research program originating with Gödel, to provide a natural arithmetic foundation for intuitionistic logic, [7]. LP plays a central role in the solution of that problem. But it also proves to be an extremely interesting logic for its own sake, a logic of justifications that is closely related to a standard logic of knowledge, S4. Further, it is generalizable to justification versions of the usual variety of logics of knowledge. It is, however, a difficult logic to work with. Semantics, see [4], is more complex than that of standard modal logics, and the same applies to its proof theory. The advantage that LP provides lies in its representation of justifications, intended to be explicit proofs in the mathematical sense. From the point of view of logics of knowledge, this provides a way of dealing with the well-known logical omniscience problems—justifications grow more complex as we continue to reason, and so they supply a measure of how hard it is to know something. With this technical machinery available, it should be possible to reason about justifications themselves in useful ways. It turns out that doing so presents considerable technical difficulties. In this paper we provide some tools. More on what these tools accomplishe will have to wait until the end of this section by which point sufficient terminology will have been introduced.

For a thorough presentation of LP including the original motivation, see [2], and for a discussion of its evolution into a family of logics of knowledge, see [6]. Here things must be restricted to a brief sketch of the basic ideas. LP has a modal-like language, but instead of a single modal operator it has an infinite family of them—they are called *proof polynomials*, or *justifications* depending on context. If t is a proof polynomial and X is a formula, $t{:}X$ is another formula which can be read informally as "t is a proof of X", or "t is a justification for X". Part of the formal machinery of the logic is a calculus on proof polynomials. LP is intended to have a close relationship with the normal modal logic S4, which in turn has a close relationship with intuitionistic logic and, as Gödel observed, S4 necessity has the properties one associates with an informal notion of provability. The formal connection between LP and arithmetic is Artemov's Arithmetic Completeness Theorem, which is not considered in this paper. The formal connection between LP and S4 is Artemov's Realization Theorem, which is given a new proof here. Loosely it says that each theorem of S4 can be converted into a theorem of LP—a *realization* of the S4 theorem—that expresses the constructive content of the S4 theorem. Negative occurrences of $\Box$ become proof variables and positive occurrences of $\Box$ become proof polynomials that may involve those variables. A statement can be found below in Theorem 2.1, and another, making use of the special machinery introduced here, in Theorem 3.6.

One of the new things in this paper is that realizations are turned into first-class mathematical objects—essentially they become functions defined on occurrences of modal operators. In order to do this conveniently, an *annotated* version of S4 is introduced, in which distinct occurrences of the necessity operator are syntactically different symbols. All this makes it easier to reason about realizations and their behavior. We think it will be found to be a useful tool generally—it certainly has been one here.

We said that LP is a difficult logic to work with. Here is an example. Suppose A, B, and C are formulas in a standard modal language, and we interpret the $\Box$ operator as a knowledge operator; $\Box P$ is read as 'P is known'. Now, suppose we have established that $A \supset C$ and $B \supset C$ in S4, a common logic of knowledge—assume these formulas contain modal operators. Of course we can conclude $(A \vee B) \supset C$ by standard classical reasoning. But suppose instead that the formulas have been replaced by realizations of them—$\Box$ occurrences have become explicit evidence, proof polynomials. And suppose we have established that a particular realization of $A \supset C$ and a particular realization of $B \supset C$ are the case in LP. Via an indirect argument, passing through Artemov's Realization Theorem, we can conclude that some realization of $(A \vee B) \supset C$ is provable in LP. The difficulty is, this realization may have little, if any, connection with the ones for $A \supset C$ and $B \supset C$, and so our original pieces of evidence, embodied in the original realizations, may be lost in the process. Can this be avoided? Can we build directly on the information we have in the realizations for $A \supset C$ and for $B \supset C$ to get a realization for $(A \vee B) \supset C$ that is in some way naturally related? This question will be addressed in Section 8 after a Realization Merging Theorem has been proved.

Here is another example. We commonly replace, in a formula $Z(A)$, a subfomula A with another formula B that has been proved to be equivalent to A, getting a new formula $Z(B)$, and conclude that $Z(A)$ and $Z(B)$ are themselves equivalent. This is something we do all the time—it works in classical propositional logic, in intuitionistic logic, and in normal modal logics. But again things are not so simple in LP, where the subformula A of $Z(A)$ may be embedded in a structure of justifications as represented by proof polynomials. If A is replaced with B, justifications must be updated to reflect the original ones combined with ones embodying the passage from A to B. Further, A may be present in different parts of $Z(A)$ with different supporting justifications. This can lead to considerable complexities. Another LP tool provided is a replacement for the usual replacement theorem, appropriate for LP.

The general plan of this paper is as follows. After a background presentation of LP we get into

new material. The LP presentation is entirely proof-theoretic—semantics plays no role here. An *annotated* version of S4 is given, in which distinct occurrences of modal operators are syntactically distinguished. This is a fundamental tool for what follows. Then *realization functions* are defined, as functions on formulas of the annotated logic. A Realization Modification Theorem is proved. This is, perhaps, best understood via the corollaries that follow it in subsequent sections. One of them provides a solution to the first problem discussed above, about extracting a realization for $(A \vee B) \supset C$ from a realization for $A \supset C$ and another for $B \supset C$. A different corollary is an analog of the classical Replacement Theorem—replacement of subformulas by equivalent subformulas produces an equivalent formula. It applies in a more restricted setting than the corollary allowing the combining of realization functions. Finally, some of these results are combined to supply a new proof of Artemov's Realization Theorem—or perhaps it can be viewed as Artemov's proof unfolded.

Since Artemov's original work, the ideas of LP have been extended to a family of similar logics, now called *Justification Logics*. Thus LP is one among many—indeed various multi-modal logics of knowledge have also been brought into the picture. I do not address the whole family of justification logics in detail here—things are complicated enough. But it is clear that my methods do extend, and I comment briefly on this at the end of the paper.

I want to thank Roman Kuznets for very careful reading of an early draft of part of this paper, and for catching several errors.

2 The Logic LP

This section contains a brief formulation of LP axiomatically. A semantics will not be needed in this paper. The language of LP, denoted L_{LP} here, is built from the following basic machinery, which comes from [2].

1. propositional variables, P, Q, P_1, P_2, ...
2. propositional constant, $\bot$
3. logical connective, $\supset$
4. proof variables, x, y, x_1, x_2, ...
5. proof constants, c, d, c_1, c_2, ...
6. function symbols ! (monadic), $\cdot$, + (binary)
7. operator symbol of the type $\langle term \rangle{:}\langle formula \rangle$

Proof polynomials are built up from proof variables and proof constants, using the function symbols. *Ground* proof polynomials are those without variables. *Formulas* are built up from propositional variables and the propositional constant $\bot$ using $\supset$ (with other connectives defined in the usual way), and an extra rule of formation: if t is a proof polynomial and X is a formula then $t{:}X$ is a formula.

The formula $t{:}X$ can be read: "t is a proof of X." Proof constants intuitively represent proofs of basic, assumed truths. Proof variables in a formula can be thought of as implicitly universally quantified over proofs. If t is a proof of $X \supset Y$ and u is a proof of X, we should think of $t \cdot u$, the application of t to u, as a proof of Y. The operation ! is a proof-checker: if t is a proof of X then $!t$ is a verification that t is such a proof. The operation + combines proofs in the sense that $t + u$ proves all the things that t proves plus all the things that u proves.

The following axiom system for LP is from [1, 2]. Axioms are specified by giving axiom schemas, and these are:

A0.	**Classical**	Enough classical propositional axiom schemes
A1.	**Application**	$t{:}(X \supset Y) \supset (s{:}X \supset (t \cdot s){:}Y)$
A2.	**Reflexivity**	$t{:}X \supset X$
A3.	**Proof Checker**	$t{:}X \supset !t{:}(t{:}X)$
A4.	**Sum**	$s{:}X \supset (s+t){:}X$
		$t{:}X \supset (s+t){:}X$

Rules of inference are modus ponens, and a version of the necessitation rule, for axioms only.

R1.	**Modus Ponens**	$\vdash Y$ provided $\vdash X$ and $\vdash X \supset Y$
R2.	**Axiom Necessitation**	$\vdash c{:}X$ where X is an axiom *A0* – *A4* and c is a proof constant.

As usual, a proof is a finite sequence of formulas each of which is an axiom or comes from earlier terms by one of the rules of inference. A notion of *derivation* can be introduced either directly, or indirectly by defining $\Gamma \vdash X$ to mean that $(G_1 \wedge \ldots \wedge G_n) \supset X$ is a theorem for some finite subset $\{G_1, \ldots, G_n\}$ of Γ.

The specification of which constants are associated with which axioms for rule *R2* applications is called a *constant specification*. A constant specification is *injective* if each proof constant is used for at most one axiom. Injective constant specifications suffice, but are not required. If a proof uses an injective constant specification, we will say the proof is *injective*, and what it proves is *injectively provable*. In [4] constant specifications were assumed to be given beforehand, and their properties were investigated in some detail. Computational complexity is dependent on details of the constant specification. In [2] things were more flexible, and constants were generally assigned during the course of a proof. In this paper we use the flexible version.

The Artemov Realization Theorem plays a fundamental role for LP. If Z is any theorem of LP, and we replace every proof polynomial by $\Box$ (the *forgetful* projection), the result is a theorem of S4. This much is easy to see: it is clearly the case for each axiom of LP, and is preserved by the LP rules of derivation. The Realization Theorem, [2], is a converse to this—an alternative formulation more in keeping with the methodology of this paper can be found in Theorem 3.6.

Theorem 2.1 (Realization Theorem) *If Z is a theorem of S4, there is some way of replacing $\Box$ symbols with proof polynomials to produce an injectively provable theorem of LP. Moreover this can be done so that negative occurrences of $\Box$ in Z are always replaced with distinct proof variables, and positive occurrences by proof polynomials that may involve those variables.*

Negative occurrences of proof variables can be thought of as inputs, and the proof polynomials involving them as outputs. Thus theorems of S4, in a sense, carry implicit constructive functional content which their embeddings into LP make explicit.

Definition 2.2 A *substitution* is a mapping from proof variables to proof polynomials. If σ is a substitution and X is a formula, we write $X\sigma$ for the result of replacing each proof variable x in X with the proof polynomial $x\sigma$. Similarly for substitution in proof polynomials.

The following is shown in [2].

Theorem 2.3 (Substitution Lemma) *If X is a theorem of LP, so is $X\sigma$. Further, if X has an injective proof, so does $X\sigma$.*

The constant specification used for proving X and that used for proving $X\sigma$ will, in general, be different, but this fact can be safely ignored for what we do here.

A fundamental result that will be used over and over in this paper is the Lifting Lemma, also from [2], which says that proofs and derivations in LP can be internalized.

Theorem 2.4 (Lifting Lemma) *Suppose*

$$s_1{:}X_1, \ldots, s_n{:}X_n, Y_1, \ldots, Y_k \vdash Z$$

then there is a proof polynomial $t(s_1, \ldots, s_n, y_1, \ldots, y_k)$ (where the y_i are variables) such that

$$s_1{:}X_1, \ldots, s_n{:}X_n, y_1{:}Y_1, \ldots, y_k{:}Y_k \vdash t(s_1, \ldots, s_n, y_1, \ldots, y_k){:}Z.$$

Moreover, if the original derivation was injective, the same is the case for the later derivation.

Corollary 2.5 *If Z has an LP proof, then for some ground proof polynomial t, $t{:}Z$ will have an LP proof, injective if the proof of Z was injective.*

The proof polynomial t in the Corollary above can always be taken so that it does not involve the operator $+$. The standard proof, by induction on axiomatic derivation length, constructively produces such a polynomial. See [2] for details.

3 Annotations and Realizations

In this section some simple machinery is introduced to keep track of modal operator *occurrences*. But first modal operators in the standard sense are needed. Let $L_\Box$ be the usual language of propositional modal logic, built up from propositional letters using $\bot$, $\supset$, and $\Box$, with other connectives and $\Diamond$ taken as defined in the usual way, if needed.

The LP Realization Theorem treats positive and negative occurrences of modal operators differently; negatives are replaced by proof variables while positives need not be. Further, different negative occurrences in a formula are replaced with distinct variables. Generally all this has been done somewhat informally, but formal machinery for it is straightforward. We introduce an annotated version, $L^a_\Box$, of the language $L_\Box$, intermediate between $L_\Box$ and L_{LP}. As will be seen, it amounts to syntactic, and not semantic, machinery.

Definition 3.1 The language $L^a_\Box$ and its features are introduced as follows.

1. Instead of a single modal operator $\Box$, there is an infinite family, $\Box_1$, $\Box_2$, ... These will be called *indexed* modal operators. Formulas of $L^a_\Box$ are built up as in $L_\Box$, but using indexed modal operators instead of $\Box$. Formulas of $L^a_\Box$ will generally be referred to as *annotated formulas*.

2. If X is an annotated formula, and X' is the result of replacing all indexed modal operators, $\Box_n$, with $\Box$, then X' is a formula of $L_\Box$. We say X is an *annotated version* of X', and X' is an *unannotated version* of X.

3. A *properly* annotated formula is an annotated formula meeting the conditions that: no indexed modal operator occurs twice; and if $\Box_n$ occurs in a negative position n is even, and if it occurs in a positive position n is odd.

Example 3.2 Here is an example of a properly annotated formula.

$$\Box_2(\Box_1 U \supset \Box_4(\Box_3 P \supset \Box_6 V)) \supset \Box_5 W \tag{1}$$

One can think of a properly annotated formula as a bookkeeping device to keep track of occurrences of modal operators and their polarities—negative occurrences are even, positive occurrences are odd. Properly annotated formulas play a fundamental role, but it is important to note that formulas that are annotated but not properly so also arise naturally. For instance, if X is properly annotated and Y is a subformula, it may not itself be properly annotated—it will not be if Y is a negative subformula of X because polarities have been reversed in passing from X to Y. Generally we will fix a properly annotated formula X and work with subformulas of it, all of which are annotated, and properly so in context as subformulas of X.

Semantically, annotations are simply ignored. That is, in an S4 model $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \Vdash \rangle$ we use the following rule of evaluation:

$$\mathcal{M}, \Gamma \Vdash \Box_n X \Longleftrightarrow \mathcal{M}, \Delta \Vdash X \text{ for every } \Delta \in \mathcal{G} \text{ with } \Gamma \mathcal{R} \Delta$$

Then in a model, an annotated formula X and its unannotated version X' behave alike at each world. As we remarked earlier, annotations are for syntactical and not for semantical purposes.

Now that we have annotated formulas, realizations can be defined functionally in a natural way. Before giving the definition, it might be useful to comment on one item that is about to come up in it. Let us say the displayed occurrence of term t in the formula $t{:}Z$ is *self-referential* if t also has an occurrence in Z. It is shown in [8] that one must admit self-referential proof constants in order to have the Realization Theorem hold. In this paper we will sometimes need to exercise control over the self-referentiality of proof variables, hence item 3 below.

Definition 3.3 Realization functions and related notions are defined as follows.

1. A *realization function* is a mapping from positive integers to proof polynomials that maps even integers to LP variables. Moreover it is assumed that all realization functions behave the same on the even integers; specifically, if r is any realization function, $r(2n) = x_n$, where $x_1, x_2, \ldots$ is the list of proof variables arranged in a standardized order.

2. If X is a formula of $L^a_\Box$, an annotated formula, and r is a realization function, by $r(X)$ is meant the result of replacing each modal operator $\Box_i$ in X with the proof polynomial $r(i)$. More precisely, for subformulas of X, $r(A \supset B) = r(A) \supset r(B)$, $r(P) = P$ for P atomic, and $r(\Box_i Z) = r(i){:}r(Z)$. Of course $r(X)$ is formula of LP.

3. Let X be an annotated formula. We say the realization function r is *non self-referential on variables over* X provided, for each subformula $\Box_{2n} Y$ of X the variable $r(2n) = x_n$ does not occur in $r(Y)$.

Example 3.4 Let r be a realization function such that the following holds, where f, g, h, and k are particular proof polynomials that need not be fully specified for present purposes, except that the only variables are those explicitly shown, and the behavior of r on other inputs is not needed.

$$\begin{array}{lcllcl} r(1) & = & g(x_2, x_3, x_5) & r(4) & = & x_2 \\ r(2) & = & x_1 & r(5) & = & h(x_1, x_2, x_3) \\ r(3) & = & f(x_3) & r(6) & = & x_3 \end{array} \tag{2}$$

Let $X = X(P)$ be formula (1) from Example 3.2 (in Section 6 P will play a special role, though it does not at the moment). Then we have the following. Note that r is non self-referential on variables over $X(P)$.

$$r(X(P)) = x_1{:}[g(x_2, x_3, x_5){:}U \supset x_2{:}(f(x_3){:}P \supset x_3{:}V)] \supset h(x_1, x_2, x_3){:}W \tag{3}$$

Finally, here is our official definition of a realization.

Definition 3.5 If X is a formula of $L_\Box$, a conventional modal formula, a *realization* of X is any formula of LP of the form $r(X')$ where r is a realization function and X' is any properly annotated version of X.

Artemov's Realization Theorem can now be given the following formulation.

Theorem 3.6 *If Z is a theorem of S4, there is a realization of Z that is an injectively provable theorem of LP. In fact if Z is a theorem of S4, then for any properly annotated version X of Z there is a realization function r such that $r(X)$ is injectively provable in LP.*

4 Restricted Realization Modification

We will be considering some general ways of modifying realization functions. One is the merging of several realization functions into a single one that is related to the originals in useful ways. Another arises in connection with an LP analog of the familiar Replacement Theorem. Originally, establishing basic properties of these modification techniques required separate but similar proofs, see [5] for one of them. Eventually the proofs were combined, making the overall work considerably simpler. The cost, however, is that we must formulate a theorem that is general enough, and such a theorem is less easy to grasp intuitively. Before reading this section you might take a look at sections 6–8 in which various corollaries are derived. These corollaries have a simpler nature, and can be used as a lead-in to the present section.

All results in this paper are algorithmic. Verification that the algorithms are correct is generally more complicated than the algorithms themselves, so in each case an algorithm is stated fully first, then its correctness is established. We hope this makes following the work somewhat easier.

The Restricted Realization Modification Theorem, below, is called 'restricted' for two reasons. First, there is a non self-referentiality on variables condition, *H–2*. (Definition 3.3 should be recalled here.) Later, in Section 7, we will see that this can sometimes be dropped (though perhaps not always). Second, a pair $\langle r_\varphi, \sigma_\varphi \rangle$, consisting of a realization function and a substitution, is constructed for each subformula φ of a given annotated formula. We will see that this can always be simplified—there is a single pair that will work uniformly for every subformula. But we must go through the restricted version before we can establish a uniform one.

It was mentioned that there will be particular modification techniques that follow from the work of this section. One concerns the merging of multiple realization functions; you can see the beginnings of this with the presence of r_1, r_2, ..., multiple realization functions, in condition *H–2* below. The other primary modification technique has to do with the replacement of a subformula of a given formula by another subformula; you can see this beginning to emerge too, with two formulas A and B being mentioned in conditions *H–1* and *H–3*.

Definition 4.1 Let X be an annotated formula (not necessarily properly annotated) and let σ be a substitution.

1. σ meets the *no new variable* condition provided that for each variable x the proof polynomial $x\sigma$ contains no variables other than x.

2. An even indexed modal operator $\Box_{2n}$ that occurs in X is said to be an *input operator* in X (whether $\Box_{2n}$ is in a negative position or not). If $\Box_{2n}$ is an input operator we say x_n is an *input variable* of $r(X)$, where r is any proper realization function (they all behave the same on even integers).

3. σ *lives on input positions* in X provided the only variables x_n for which $x_n\sigma \neq x_n$ are such that $\Box_{2n}$ occurs in X .

There is a certain complication that should be discussed before launching into the formal details. We care about *proper* annotation but, as noted earlier, subformulas of a properly annotated formula need not be properly annotated. In order to deal with this we fix a formula $X(P)$ that is properly annotated and work with subformulas of $X(P)$, which are properly annotated within $X(P)$, but may not be when considered on their own.

In what follows, if $\psi(P)$ is an annotated formula and P is a propositional letter, $\psi(A)$ is the result of replacing all occurrences of P in $\psi(P)$ with occurrences of the annotated formula A.

Definition 4.2 Let $X(P)$ be an annotated formula in which the propositional letter P has at most one positive occurrence, let A and B be annotated formulas, and let r_0 be a realization function.

1. For a subformula $\varphi(P)$ of $X(P)$, we say a realization/substitution pair $\langle r, \sigma\rangle$ *replaces* $r_0(A)$ *with* $r_0(B)$ *at* P *in* $\varphi(P)$ *within* $X(P)$ provided:

   (a) if $\varphi(P)$ is a positive subformula of $X(P)$ then $r_0(\varphi(A))\sigma \supset r(\varphi(B))$ has an injective LP proof;

   (b) if $\varphi(P)$ is a negative subformula of $X(P)$ then $r(\varphi(B)) \supset r_0(\varphi(A))\sigma$ has an injective LP proof.

2. We say $\langle r, \sigma\rangle$ *hereditarily replaces* $r_0(A)$ with $r_0(B)$ at P in $X(P)$ provided, for each subformula $\varphi(P)$ of $X(P)$, $\langle r, \sigma\rangle$ replaces $r_0(A)$ with $r_0(B)$ at P in $\varphi(P)$ within $X(P)$.

The following theorem allows for N realization functions. In this paper at most two will be needed for the Realization Theorem, but applications not considered here can involve more.

Theorem 4.3 (Restricted Realization Modification) *Assume the following.*

H–1. $X(P)$ *is a properly annotated formula in which the propositional letter* P *has at most one positive occurrence,* A *and* B *are properly annotated formulas,* A *and* $X(P)$ *share no indexes, and* B *and* $X(P)$ *share no indexes.*

H–2. r_1, r_2, ..., r_N *are realization functions, all non self-referential on variables over* $X(A)$.

H–3. For each $k = 1, \ldots, N$, $r_k(A) \supset r_k(B)$ *has an injective proof.*

H–4. $r_1(B) = r_2(B) = \ldots = r_N(B)$.

Then for each subformula $\varphi(P)$ *of* $X(P)$ *there is some realization/substitution pair* $\langle r_\varphi, \sigma_\varphi\rangle$ *such that:*

C–1. $\langle r_\varphi, \sigma_\varphi\rangle$ *replaces* $r_k(A)$ *with* $r_k(B)$ *at* P *in* $\varphi(P)$ *within* $X(P)$, *for* $k = 1, \ldots, N$.

C–2. σ_φ *lives on input positions in* $\varphi(P)$*;*

C–3. σ_φ *meets the no new variable condition;*

C–4. If r_k *is non self-referential on variables over* $X(B)$ *for every* $k = 1, \ldots, N$ *then* r_φ *is non self-referential on variables over* $X(B)$*.*

First the algorithm for constructing $\langle r_\varphi, \sigma_\varphi \rangle$ is given, then it is proved correct.

Begin Algorithm Assume the conditions *H–1* through *H–4* hold. The algorithm proceeds recursively, on the structure of $\varphi(P)$, a subformula of $X(P)$. Here are the cases.

Base Case: $\varphi(P)$ is atomic. Set $r_\varphi = r_1$ and σ_φ to be the identity substitution.

Modal Case: $\varphi(P)$ is $\Box_i\theta(P)$, and $\langle r_\theta, \sigma_\theta \rangle$ has been constructed.

$\varphi(P)$ *positive:* By hypothesis, each of the following is provable.

$$\begin{gathered} r_1(\theta(A))\sigma_\theta \supset r_\theta(\theta(B)) \\ r_2(\theta(A))\sigma_\theta \supset r_\theta(\theta(B)) \\ \vdots \\ r_N(\theta(A))\sigma_\theta \supset r_\theta(\theta(B)) \end{gathered}$$

Use the Lifting Lemma to produce ground proof polynomials $u_1, \ldots, u_N$ that "prove" the respective formulas above, that is, the following are injective LP theorems.

$$\begin{gathered} u_1{:}[r_1(\theta(A))\sigma_\theta \supset r_\theta(\theta(B))] \\ u_2{:}[r_2(\theta(A))\sigma_\theta \supset r_\theta(\theta(B))] \\ \vdots \\ u_N{:}[r_N(\theta(A))\sigma_\theta \supset r_\theta(\theta(B))] \end{gathered}$$

Then set $\sigma_\varphi = \sigma_\theta$ and define r_φ as follows.

$$r_\varphi(n) = \begin{cases} [u_1 \cdot r_1(n) + u_2 \cdot r_2(n) + \ldots + u_N \cdot r_N(n)]\sigma_\theta & \text{if } n = i \\ r_\theta(n) & \text{otherwise} \end{cases}$$

The sum displayed in the $n = i$ case needs parentheses since $+$ is not assumed to be associative, but any grouping will work in this case.

$\varphi(P)$ *negative:* In this case i must be even. By hypothesis, the following are provable.

$$\begin{gathered} r_\theta(\theta(B)) \supset r_1(\theta(A))\sigma_\theta \\ r_\theta(\theta(B)) \supset r_2(\theta(A))\sigma_\theta \\ \vdots \\ r_\theta(\theta(B)) \supset r_N(\theta(A))\sigma_\theta \end{gathered}$$

Use the Lifting Lemma to produce proof polynomials $u_1, \ldots, u_N$ that "prove" the respective formulas above, so the following are LP theorems.

$$\begin{gathered} u_1{:}[r_\theta(\theta(B)) \supset r_1(\theta(A))\sigma_\theta] \\ u_2{:}[r_\theta(\theta(B)) \supset r_2(\theta(A))\sigma_\theta] \\ \vdots \\ u_N{:}[r_\theta(\theta(B)) \supset r_N(\theta(A))\sigma_\theta] \end{gathered}$$

Then set $r_\varphi = r_\theta$ and define σ_φ as follows, where $i = 2j$.

$$x_n\sigma_\varphi = \begin{cases} (u_1 + u_2 + \ldots + u_N) \cdot x_j & \text{if } n = j \\ x_n\sigma_\theta & \text{otherwise} \end{cases}$$

Implication Case: $\varphi(P)$ is $\theta(P) \supset \eta(P)$ and $\langle r_\theta, \sigma_\theta\rangle$ and $\langle r_\eta, \sigma_\eta\rangle$ have been constructed. It is shown that the two substitutions commute. Set $\sigma_\varphi = \sigma_\theta\sigma_\eta = \sigma_\eta\sigma_\theta$, and define r_φ as follows.

$$r_\varphi(n) = \begin{cases} r_\theta(n)\sigma_\eta & \text{if } \Box_n \text{ in } \theta(B) \\ r_\eta(n)\sigma_\theta & \text{if } \Box_n \text{ in } \eta(B) \\ r_1(n) & \text{otherwise} \end{cases}$$

End Algorithm

Begin Correctness Proof Assume hypotheses *H–1* through *H–4* of Theorem 4.3 hold. Note that, because of *H–1*, both $X(A)$ and $X(B)$ are properly annotated. We proceed by induction on the complexity of the subformula $\varphi(P)$. Call a subformula $\varphi(P)$ of $X(P)$ *good* provided there is some $\langle r_\varphi, \sigma_\varphi\rangle$ such that *C–1* to *C–4* hold; we also say $\langle r_\varphi, \sigma_\varphi\rangle$ is a *witness* to the goodness of $\varphi(P)$. We will show every subformula of $X(P)$ is good, and that the algorithm just described produces witnesses.

Let $\varphi(P)$ be a subformula of $X(P)$ and as an induction hypothesis, suppose all its proper subformulas are good—we show $\varphi(P)$ itself is good. There are several cases to consider.

Case: $\varphi(P)$ is atomic. The algorithm sets $r_\varphi = r_1$ and σ_φ to be the identity substitution. σ_φ trivially lives on input positions in $\varphi(P)$ and meets the no new variable condition. And if r_1 is non self-referential on variables over $X(B)$, of course so is r_φ. To finish the verification that $\langle r_\varphi, \sigma_\varphi\rangle$ is a witness to the goodness of $\varphi(P)$ there are two subcases to consider.

Subcase: $\varphi(P)$ is not P; say it is Q, which might occur positively or negatively. Then for each k, both $r_k(\varphi(A))\sigma_\varphi \supset r_\varphi(\varphi(B))$ and $r_\varphi(\varphi(B)) \supset r_k(\varphi(A))\sigma_\varphi$ are simply $Q \supset Q$, which certainly has an injective LP proof.

Subcase: $\varphi(P)$ is P. This must be a positive subformula of $X(P)$ since the occurrence of P in $X(P)$ is positive. For each k we need the injective provability of $r_k(\varphi(A))\sigma_\varphi \supset r_\varphi(\varphi(B))$. This is $r_k(A) \supset r_1(B)$, but $r_1(B) = r_k(B)$ by *H–4*, so we need injective provability of $r_k(A) \supset r_k(B)$, which we have by *H–3*.

Case: $\varphi(P)$ is $\Box_i\theta(P)$, and this is a positive subformula of $X(P)$ (hence i is odd). By the induction hypothesis there is $\langle r_\theta, \sigma_\theta\rangle$ that witnesses the goodness of $\theta(P)$. In particular, for each $k = 1, \ldots, N$ the following is an injective theorem of LP: $r_k(\theta(A))\sigma_\theta \supset r_\theta(\theta(B))$. By the Lifting Lemma, Corollary 2.5, for each $k = 1, \ldots, N$ there is a ground proof polynomial u_k such that $u_k{:}[r_k(\theta(A))\sigma_\theta \supset r_\theta(\theta(B))]$ is an injective theorem of LP. Then, using Application and Modus Ponens, the following is an injective LP theorem for $k = 1, \ldots, N$.

$$(r_k(i)\sigma_\theta){:}r_k(\theta(A))\sigma_\theta \supset (u_k \cdot (r_k(i)\sigma_\theta)){:}r_\theta(\theta(B))$$

We have $(r_k(i)\sigma_\theta){:}[r_k(\theta(A))\sigma_\theta] = [r_k(i){:}r_k(\theta(A))]\sigma_\theta = r_k(\Box_i\theta(A))\sigma_\theta = r_k(\varphi(A))\sigma_\theta$. Then, for $k = 1, \ldots, N$ we have injective provability of the following.

$$r_k(\varphi(A))\sigma_\theta \supset (u_k \cdot (r_k(i)\sigma_\theta)){:}r_\theta(\theta(B))$$

It follows by LP axiom *A4*, Sum, that for each $k = 1, \ldots, N$, the following is injectively provable:

$$r_k(\varphi(A))\sigma_\theta \supset \{u_1 \cdot (r_1(i)\sigma_\theta) + u_2 \cdot (r_2(i)\sigma_\theta) + \ldots + u_N \cdot (r_N(i)\sigma_\theta)\}{:}r_\theta(\theta(B))$$

or equivalently, since each u_k contains no variables,

$$r_k(\varphi(A))\sigma_\theta \supset \{[u_1 \cdot r_1(i) + u_2 \cdot r_2(i) + \ldots + u_N \cdot r_N(i)]\sigma_\theta\}{:}r_\theta(\theta(B))$$

The algorithm has us set $\sigma_\varphi = \sigma_\theta$, so of course σ_φ meets the no new variable condition. Also σ_θ lives on the input positions in $\theta(P)$, and hence σ_φ lives on the input positions in $\varphi(P)$ as well, since i must be odd.

The algorithm also has us set r_φ to be like r_θ, except that

$$r_\varphi(i) = (u_1 \cdot r_1(i) + u_2 \cdot r_2(i) + \ldots + u_N \cdot r_N(i))\,\sigma_\theta.$$

Note that, since $\Box_i\theta(B)$ is a subformula of $X(B)$ which is properly annotated, i cannot occur as an index in $\theta(B)$, and hence $r_\theta(\theta(B)) = r_\varphi(\theta(B))$. Putting all this together, we have the following, injectively, for each $k = 1, \ldots, N$.

$$\begin{aligned} r_k(\varphi(A))\sigma_\varphi &= r_k(\varphi(A))\sigma_\theta \\ &\supset \{[u_1 \cdot r_1(i) + u_2 \cdot r_2(i) + \ldots + u_N \cdot r_N(i)]\sigma_\theta\}{:}r_\theta(\theta(B)) \\ &= r_\varphi(i){:}r_\theta(\theta(B)) \\ &= r_\varphi(i){:}r_\varphi(\theta(B)) \\ &= r_\varphi(\Box_i\theta(B)) \\ &= r_\varphi(\varphi(B)) \end{aligned}$$

Finally, suppose each r_k is non self-referential on variables over $X(B)$. We show the same is true of r_φ. Since r_φ is the same as r_θ except on i, which is odd, and r_θ is non self-referential on variables over $X(B)$, if r_φ failed to be so it must be that $\Box_i : \theta(B)$ is part of some subformula $\Box_{2n}Z(B)$ of $X(B)$ and the variable x_n occurs in $r_\varphi(i)$. But $r_\varphi(i) = (u_1 \cdot r_1(i) + u_2 \cdot r_2(i) + \ldots + u_N \cdot r_N(i))\,\sigma_\theta$ where each u_k is ground. Since σ_θ meets the no new variable condition, each $r_k(i)\sigma_\theta$ has the same variables as $r_k(i)$. But this violates the condition that each r_k is non self-referential on variables over $X(B)$.

Case: $\varphi(P)$ is $\Box_i\theta(P)$, where this is a negative subformula of $X(P)$. By the induction hypothesis there is $\langle r_\theta, \sigma_\theta\rangle$ that witnesses the goodness of $\theta(P)$. Since $X(P)$ is properly annotated, i must be even, say it is $2j$. Then every realization function will replace $\Box_i$ with the proof variable x_j. Since $\theta(P)$ is a negative subformula, for each $k = 1, \ldots, N$ the following are injectively provable: $r_\theta(\theta(B)) \supset r_k(\theta(A))\sigma_\theta$. By the Lifting Lemma, Corollary 2.5, there are ground proof polynomials u_k such that $u_k{:}[r_\theta(\theta(B)) \supset r_k(\theta(A))\sigma_\theta]$ is an injective theorem of LP, for $k = 1, \ldots, N$. Then by the Sum axioms, for each $k = 1, \ldots, N$, the following is injectively provable.

$$(u_1 + u_2 + \ldots + u_N){:}[r_\theta(\theta(B)) \supset r_k(\theta(A))\sigma_\theta]$$

From these using Application and Modus Ponens, the following is an injective LP theorem for each $k = 1, \ldots, N$.

$$x_j{:}r_\theta(\theta(B)) \supset ((u_1 + u_2 + \ldots + u_N) \cdot x_j){:}r_k(\theta(A))\sigma_\theta$$

The algorithm has us set $r_\varphi = r_\theta$. If each r_k is non self-referential on variables over $X(B)$ so is r_θ by the induction hypothesis, and then so r_φ since $r_\varphi = r_\theta$. Note that $r_\varphi(\varphi(B)) = r_\theta(\Box_i\theta(B)) = x_j : r_\theta(\theta(B))$. Thus we have injective provability of the following, for each $k = 1, \ldots, N$.

$$r_\varphi(\varphi(B)) \supset ((u_1 + u_2 + \ldots + u_N) \cdot x_j) : r_k(\theta(A))\sigma_\theta$$

The algorithm has us set σ_φ to be the same as σ_θ except that $x_j\sigma_\varphi = (u_1 + u_2 + \ldots + u_N) \cdot x_j$. Then we must have injective provability of the following, for each $k = 1, \ldots, N$.

$$r_\varphi(\varphi(B)) \supset (x_j\sigma_\varphi) : r_k(\theta(A))\sigma_\theta$$

By hypothesis, σ_θ meets the no new variable condition, and since $(u_1 + u_2 + \ldots + u_N) \cdot x_j$ contains only x_j as a variable, σ_φ also meets this condition. Also σ_θ lives on the input positions in $\theta(P)$. Since σ_φ differs from σ_θ only on x_j, then σ_φ lives on the input positions in $\Box_i\theta(P) = \Box_{2j}\theta(P) = \varphi(P)$.

Since $\Box_i\theta(A)$, that is $\Box_{2j}\theta(A)$, is a subformula of $X(A)$, and each r_k is non self-referential on variables over $X(A)$, then x_j cannot occur in $r_k(\theta(A))$ for any $k = 1, \ldots, N$. Consequently for each $k = 1, \ldots, N$, $(x_j\sigma_\varphi) : r_k(\theta(A))\sigma_\theta = (x_j\sigma_\varphi) : r_k(\theta(A))\sigma_\varphi = [r_k(\Box_i\theta(A))]\sigma_\varphi = [r_k(\varphi(A))]\sigma_\varphi$. Then we have the injective provability of the following, for each $k = 1, \ldots, N$.

$$r_\varphi(\varphi(B)) \supset r_k(\varphi(A))\sigma_\varphi$$

Case: $\varphi(P)$ is $\theta(P) \supset \eta(P)$. By the induction hypothesis both $\theta(P)$ and $\eta(P)$ are good. Let $\langle r_\theta, \sigma_\theta\rangle$ witness the goodness of $\theta(P)$ and $\langle r_\eta, \sigma_\eta\rangle$ witness the goodness of $\eta(P)$. We combine these into something that witnesses the goodness of $\theta(P) \supset \eta(P)$.

We first show we have commutativity of substitutions. That is, $\sigma_\theta\sigma_\eta = \sigma_\eta\sigma_\theta$. We do this by a case analysis.

Suppose $\Box_{2i}$ is in input position in $\theta(P)$; we show $x_i(\sigma_\theta\sigma_\eta) = x_i(\sigma_\eta\sigma_\theta)$. The substitution σ_θ lives on input positions in $\theta(P)$, while σ_η lives in input positions in $\eta(P)$, and the input positions of $\theta(P)$ and $\eta(P)$ are not shared since $\theta(P) \supset \eta(P)$ is a subformula of the properly annotated formula $X(P)$. Consequently we have $x_i\sigma_\eta = x_i$. Then of course $x_i(\sigma_\eta\sigma_\theta) = x_i\sigma_\theta$. Also, since σ_θ meets the no new variable condition, $x_i\sigma_\theta$ only has x_i as a variable, so $x_i(\sigma_\theta\sigma_\eta) = x_i\sigma_\theta$. So for $\Box_{2i}$ in input position in $\theta(P)$ we have $x_i(\sigma_\theta\sigma_\eta) = x_i(\sigma_\eta\sigma_\theta)$.

In a similar way $x_i(\sigma_\theta\sigma_\eta) = x_i(\sigma_\eta\sigma_\theta)$ if $\Box_{2i}$ is in input position in $\eta(P)$. Hence $x_i(\sigma_\theta\sigma_\eta) = x_i(\sigma_\eta\sigma_\theta)$ if $\Box_{2i}$ is in input position in $\theta(P) \supset \eta(P)$.

Finally, if $\Box_{2i}$ is not in input position in $\theta(P) \supset \eta(P)$, $x_i\sigma_\theta = x_i\sigma_\eta = x_i$ since σ_θ lives on input positions in $\theta(P)$ and σ_η lives on input positions in $\eta(P)$. Thus in all cases, $x_i(\sigma_\theta\sigma_\eta) = x_i(\sigma_\eta\sigma_\theta)$. Hence commutativity of substitutions has been shown.

The algorithm has us set $\sigma_\varphi = \sigma_\theta\sigma_\eta = \sigma_\eta\sigma_\theta$. Since both σ_θ and σ_η meet the no new variable condition, it follows that σ_φ also does. And because $x_i\sigma_\theta = x_i\sigma_\eta = x_i$ if x_i is not in input position in $\theta(P) \supset \eta(P)$, σ_φ lives on input positions in $\theta(P) \supset \eta(P)$.

Next the algorithm has us define a realization function as follows.

$$r_\varphi(n) = \begin{cases} r_\theta(n)\sigma_\eta & \text{if } \Box_n \text{ is in } \theta(B) \\ r_\eta(n)\sigma_\theta & \text{if } \Box_n \text{ is in } \eta(B) \\ r_1(n) & \text{otherwise} \end{cases}$$

Since $\theta(B) \supset \eta(B)$ is a subformula of the *properly* annotated formula $X(B)$, the antecedent and the consequent cannot share an index, so r_φ is well-defined. It is a realization function, that is, $r_\varphi(2i) = x_i$, by the following argument. Suppose that $\Box_{2i}$ occurs in $\theta(B)$; then it cannot occur in $\eta(B)$. Since r_θ is a realization function, $r_\theta(2i) = x_i$, and since σ_η lives on input positions in $\eta(P)$, and $\Box_{2i}$ is not one of them, $x_i\sigma_\eta = x_i$. Thus $r_\varphi(2i) = x_i$ in this case. Similarly if $\Box_{2i}$ occurs in $\eta(B)$. Obviously if things fall into the 'otherwise' case the result is immediate since r_1 is a realization function.

Provability of certain implications must be shown. The proof now divides into two cases, depending on whether $\varphi(P) = \theta(P) \supset \eta(P)$ is a positive or a negative subformula of $X(P)$. We'll cover the positive case; the other is similar. In the positive case, $\theta(P)$ is a negative subformula of $X(P)$, and $\eta(P)$ positive. Since $\langle r_\theta, \sigma_\theta\rangle$ witnesses the goodness of $\theta(P)$ and $\langle r_\eta, \sigma_\eta\rangle$ witnesses the goodness of $\eta(P)$, the following are injective theorems of LP for each $k = 1, \ldots, N$:

$$r_\theta(\theta(B)) \supset r_k(\theta(A))\sigma_\theta$$
$$r_k(\eta(A))\sigma_\eta \supset r_\eta(\eta(B))$$

What must be shown is the injective provability of the following, for $k = 1, \ldots, N$.

$$r_k(\varphi(A))\sigma_\varphi \supset r_\varphi(\varphi(B))$$

By definition of φ, we need injective provability of

$$r_k(\theta(A) \supset \eta(A))\sigma_\varphi \supset r_\varphi(\theta(B) \supset \eta(B))$$

or equivalently,

$$[r_k(\theta(A))\sigma_\varphi \supset r_k(\eta(A))\sigma_\varphi] \supset [r_\varphi(\theta(B)) \supset r_\varphi(\eta(B))],$$

so it is enough to show the injective provability of the following.

$$r_\varphi(\theta(B)) \supset r_k(\theta(A))\sigma_\varphi$$
$$r_k(\eta(A))\sigma_\varphi \supset r_\varphi(\eta(B))$$

We have injective provability of $r_\theta(\theta(B)) \supset r_k(\theta(A))\sigma_\theta$, hence we also have injective provability of $r_\theta(\theta(B))\sigma_\eta \supset r_k(\theta(A))\sigma_\theta\sigma_\eta$ by Theorem 2.3, but this is $r_\varphi(\theta(B)) \supset r_k(\theta(A))\sigma_\varphi$. Likewise we have injective provability of $r_k(\eta(A))\sigma_\eta \supset r_\eta(\eta(B))$, so we also have injective provability of $r_k(\eta(A))\sigma_\eta\sigma_\theta \supset r_\eta(\eta(B))\sigma_\theta$, and this is $r_k(\eta(A))\sigma_\varphi \supset r_\varphi(\eta(B))$.

Finally, suppose r_k is non self-referential on variables over $X(B)$ for each $k = 1, \ldots, N$, and so by the induction hypothesis, r_θ and r_η also are. Suppose r_φ failed to be non self-referential on variables over $X(B)$; we derive a contradiction. By the supposition, there must be a subformula $\Box_{2n}W(B)$ of $X(B)$ such that x_n occurs in $r_\varphi(W(B))$. Then x_n occurs in $r_\varphi(j)$ for some $\Box_j$ in $W(B)$. There are three cases to consider: $\Box_j$ is part of $\theta(B)$, $\Box_j$ is part of $\eta(B)$, and $\Box_j$ is in $X(B)$ but outside both $\theta(B)$ and $\eta(B)$.

Suppose first that $\Box_j$ is part of $\theta(B)$. From the definition of r_φ in this case, $r_\varphi(j) = r_\theta(j)\sigma_\eta$, and x_n occurs in this. Since σ_η meets the no new variable condition, x_n must occur in $r_\theta(j)$. But this contradicts the assumption that r_θ is non self-referential on variables over $X(B)$. The argument is similar if $\Box_j$ is part of $\eta(B)$. And finally if $\Box_j$ is not part of either $\theta(B)$ or $\eta(B)$, then $r_\varphi(j) = r_1(j)$, and this contradicts the condition that r_1 is non self-referential on variables over $X(B)$.

End Correctness Proof

5 Uniform Realization Modification

The theorem proved in the previous section establishes the existence of a pair $\langle r_\varphi, \sigma_\varphi\rangle$ that combines and alters realizations in certain ways, but there is one such pair for each subformula φ of a given formula X. Now we show this can be done uniformly: there is a single realization/substitution pair that will serve for every subformula of X.

Theorem 5.1 (Uniform Realization Modification) *Assume hypotheses H–1 to H–4 of Theorem 4.3. Then there is some single realization/substitution pair $\langle r, \sigma\rangle$ such that:*

U–1. $\langle r, \sigma\rangle$ hereditarily replaces $r_k(A)$ with $r_k(B)$ at P in $X(P)$ for $k = 1, \ldots, N$;

U–2. σ lives on input positions in $X(P)$;

U–3. σ meets the no new variable condition;

U–4. If r_k is non self-referential on variables over $X(B)$ for $k = 1, \ldots, N$ then r is non self-referential on variables over $X(B)$.

Begin Algorithm Use the algorithm of Theorem 4.3 to construct $\langle r_\varphi, \sigma_\varphi\rangle$ for each subformula φ of X. Then set $r = r_X$ and $\sigma = \sigma_X$.
End Algorithm

Begin Correctness Proof
For each subformula $\varphi(P)$ of $X(P)$ let $\langle r_\varphi, \sigma_\varphi\rangle$ be the realization/substitution pair constructed according to the algorithm of Theorem 4.3. We will show that if we take $r = r_X$ and $\sigma = \sigma_X$, this provides an appropriate $\langle r, \sigma\rangle$. Conditions *U–2* to *U–4* are immediate by *C–2* to *C–4* of Theorem 4.3. To show condition *U–1* here is a useful bit of temporary terminology. For a subformula $\varphi(P)$ of $X(P)$:

> $\langle r, \sigma\rangle$ meets the *Hereditary Condition* on $\varphi(P)$ if, for each subformula $\psi(P)$ of $\varphi(P)$, $\langle r, \sigma\rangle$ replaces $r_k(A)$ with $r_k(B)$ at P in $\psi(P)$ within $X(P)$, for each $k = 1, \ldots, N$.

We will show that for each subformula $\varphi(P)$, the pair $\langle r_\varphi, \sigma_\varphi\rangle$ meets the Hereditary Condition on $\varphi(P)$. Then taking $\varphi(P)$ to be $X(P)$ will finish the argument. We show this by going through each of the cases in the proof of Theorem 4.3. When we refer to *C–1* we mean this conclusion of Theorem 4.3.

The base case is where $\varphi(P)$ is atomic, possibly P, possibly not. Either way there are no proper subformulas to consider. Since $\langle r_\varphi, \sigma_\varphi\rangle$ has property *C–1* on $\varphi(P)$ itself, $\langle r_\varphi, \sigma_\varphi\rangle$ meets the Hereditary Condition on $\varphi(P)$ in this case.

Suppose $\varphi(P)$ is $\Box_i\theta(P)$, a positive subformula of $X(P)$, and $\langle r_\theta, \sigma_\theta\rangle$ meets the Hereditary Condition on $\theta(P)$. In this case, r_φ is the same as r_θ except on i and i does not occur in $\theta(B)$ since $X(B)$ is properly annotated, so r_φ and r_θ agree on $\theta(B)$ and its subformulas. Also $\sigma_\varphi = \sigma_\theta$. It follows that $\langle r_\varphi, \sigma_\varphi\rangle$ also meets the Hereditary Condition on $\theta(P)$. By construction, $\langle r_\varphi, \sigma_\varphi\rangle$ has property *C–1* on $\varphi(P)$ itself, so it meets the Hereditary Condition on $\varphi(P)$.

Similarly suppose $\varphi(P)$ is $\Box_{2j}\theta(P)$, a negative subformula, and $\langle r_\theta, \sigma_\theta\rangle$ meets the Hereditary Condition on $\theta(P)$. In this case $r_\varphi = r_\theta$ and σ_φ is the same as σ_θ except on x_j. Since each r_k $(k = 1, \ldots, N)$ is non self-referential on variables over $X(A)$, x_j cannot occur in any $r_k(\theta(A))$, and hence σ_φ and σ_θ agree on the variables of each $r_k(\theta(A))$, and their subformulas. It follows that $\langle r_\varphi, \sigma_\varphi\rangle$ meets the Hereditary Condition on $\theta(P)$. Since $\langle r_\varphi, \sigma_\varphi\rangle$ has property *C–1* on $\varphi(P)$ itself by construction, it meets the Hereditary Condition on $\varphi(P)$.

Finally, suppose $\varphi(P)$ is $\theta(P) \supset \eta(P)$ and $\langle r_\theta, \sigma_\theta \rangle$ meets the Hereditary Condition on $\theta(P)$ and $\langle r_\eta, \sigma_\eta \rangle$ meets the Hereditary Condition on $\eta(P)$. Let $W(P)$ be any subformula of $\varphi(P)$. I'll discuss the case where $W(P)$ is a positive subformula, the argument is similar if $W(P)$ is negative. In this positive case we must show $r_k(W(A))\sigma_\varphi \supset r_\varphi(W(B))$ is injectively provable for each $k = 1, \ldots, N$. If $W(P)$ is $\varphi(P)$ itself we have the result by Theorem 4.3, so now suppose $W(P)$ is a proper subformula. Then it must be a subformula of $\theta(P)$ or of $\eta(P)$, say the former—the argument is similar if it is the later. Then $r_\varphi(W(B)) = r_\theta(W(B))\sigma_\eta$, from the definition of r_φ. Also $\sigma_\varphi = \sigma_\theta\sigma_\eta$. We have injective provability of $r_k(W(A))\sigma_\theta \supset r_\theta(W(B))$ since $\langle r_\theta, \sigma_\theta \rangle$ meets the Hereditary Condition on $\theta(P)$. Then by Theorem 2.3 we have injective provability of $r_k(W(A))\sigma_\theta\sigma_\eta \supset r_\theta(W(B))\sigma_\eta$, which is what was wanted.

End Correctness Proof

6 The Replacement Theorem

In S4 (and in normal modal logics generally) one can show a replacement result: If $A \equiv B$ is provable, and $X(B)$ is like $X(A)$ except that some subformula occurrences of A have been replaced with B, then $X(A) \equiv X(B)$ is also provable. Equivalence usually plays a central role here, and this has its problems for LP. If $A \equiv B$ is expanded into a formula in conjunctive normal form one sees that A occurs both positively and negatively, as does B. Since positive and negative occurrences of modal operators play different roles when realized in LP, any LP analog of the replacement result in a form that uses equivalence should not be expected. There is, however, a version of Replacement that is less problematic for present purposes. If $X(B)$ is like $X(A)$ except that some *positive* occurrences of A have been replaced with B, then if $A \supset B$ is provable so is $X(A) \supset X(B)$. Here is a formal statement of it, for S4, in a version that uses notation from Section 4.

Proposition 6.1 *Let $\varphi(P)$ be a formula of $L_\Box$ in which the propositional letter P has only positive occurrences. Let $\varphi(Z)$ be the result of replacing occurrences of P with occurrences of the $L_\Box$ formula Z. Then, if $A \supset B$ is provable in S4, so is $\varphi(A) \supset \varphi(B)$.*

In this form, Replacement respects polarity of subformula occurrence. There is one more minor problem before we get to the serious ones for an LP analog. The Proposition allows for the replacement of several occurrences of A with occurrences of B. We will be interested in using properly annotated formulas. But if $\varphi(P)$ and A are both properly annotated, and A actually contains indexed modal operators, $\varphi(A)$ can never be properly annotated if P occurs more than once in $\varphi(P)$. Very simply, the requirement on proper annotations that no indexed modal operator occurs more than once would be violated in $\varphi(A)$. So we must restrict ourselves to the replacement of single occurrences of subformulas. Of course multiple replacements can be done sequentially.

Now we get to the serious matters. Proof polynomials represent justifications. If A is replaced with B inside a more complex LP formula, justifications for A must be adjusted to incorporate a justification for the passage from A to B, justifications for subformulas containing justifications for A need adjustment, and so on up. A version of Replacement for LP is not simple to formulate. The following originated in [5] (with a somewhat stronger conclusion), but here it is treated as an immediate consequence of the more general Uniform Realization Modification Theorem 5.1. Once again we have a result proved under a non self-referentiality condition on variables.

Theorem 6.2 (Replacement For LP) *Assume the following.*

1. *$X(P)$ is a properly annotated formula in which the propositional letter P has one positive occurrence, A and B are properly annotated formulas, A and $X(P)$ share no annotations, and B and $X(P)$ share no annotations;*

2. *r_1 is a realization function that is non self-referential on variables over $X(A)$;*

3. *$r_1(A) \supset r_1(B)$ has an injective LP proof.*

Then there is some realization/substitution pair $\langle r, \sigma\rangle$ that hereditarily replaces $r_1(A)$ with $r_1(B)$ at P in $X(P)$. Also, σ lives on the input positions in $\varphi(P)$ and meets the no new variable condition. Finally, if r_1 is non self-referential on variables over $X(B)$ then r will also be non self-referential on variables over $X(B)$.

Proof This is the special case of Theorem 5.1 in which only a single realization function is given. ■

Example 6.3 This continues Example 3.4, and an explicit connection will be made shortly.

First consider the $L_\Box$ formula $\Box(\Box U \supset \Box(\Box\Box R \supset \Box V)) \supset \Box W$. In this, $\Box R$ has a positive occurrence. Since $\Box R \supset \Box\Box R$ is a theorem of $\mathsf{S4}$, by an application of Proposition 6.1 the following must also be an $\mathsf{S4}$ theorem—$\Box R$ is replaced with $\Box\Box R$.

$$[\Box(\Box U \supset \Box(\Box\Box R \supset \Box V)) \supset \Box W] \supset [\Box(\Box U \supset \Box(\Box\Box\Box R \supset \Box V)) \supset \Box W] \tag{4}$$

Next, consider the LP formula $x_1{:}[g(x_2, x_3, x_5){:}U \supset x_2{:}(f(x_3){:}k(x_3){:}R \supset x_3{:}V)] \supset h(x_1, x_2, x_3){:}W$, where $f(x_3)$, $g(x_2, x_3, x_5)$, $h(x_1, x_2, x_3)$, and $k(x_3)$ are proof polynomials whose details need not concern us for this example. Notice that the forgetful projection of this LP formula is the $L^\Box$ formula $\Box(\Box U \supset \Box(\Box\Box R \supset \Box V)) \supset \Box W$ looked at above. Now, $k(x_3){:}R \supset !k(x_3){:}k(x_3){:}R$ is an LP theorem, an axiom in fact, with forgetful projection $\Box R \supset \Box\Box R$. If a simple analog of Proposition 6.1 held for LP, we would expect the following to be an LP theorem.

$$\begin{array}{l}\{x_1{:}[g(x_2, x_3, x_5){:}U \supset x_2{:}(f(x_3){:}k(x_3){:}R \supset x_3{:}V)] \supset h(x_1, x_2, x_3){:}W\}\\ \supset\\ \{x_1{:}[g(x_2, x_3, x_5){:}U \supset x_2{:}(f(x_3){:}!k(x_3){:}k(x_3){:}R \supset x_3{:}V)] \supset h(x_1, x_2, x_3){:}W\}\end{array}$$

We have not verified that it is not a theorem, but almost certainly this is the case. Instead we apply Theorem 6.2 and conclude that formula (5) below really is an LP theorem for particular proof constants c, d, and e—how this comes about will be discussed in detail. Notice that the forgetful projection of LP theorem (5) is the $\mathsf{S4}$ theorem (4).

$$\begin{array}{l}\{(e \cdot x_1){:}[g(d \cdot x_2, x_3, x_5){:}U \supset (d \cdot x_2){:}(f(x_3){:}k(x_3){:}R \supset x_3{:}V)] \supset h(e \cdot x_1, d \cdot x_2, x_3){:}W\}\\ \supset\\ \{x_1{:}[g(d \cdot x_2, x_3, x_5){:}U \supset x_2{:}((c \cdot f(x_3)){:}!k(x_3){:}k(x_3){:}R \supset x_3{:}V)] \supset h(e \cdot x_1, d \cdot x_2, x_3){:}W\}\end{array} \tag{5}$$

In order to apply Theorem 6.2 to produce (5) annotated formulas must be introduced. It is here that we continue Example 3.4. Suppose we take A to be $\Box_7 R$ and B to be $\Box_9\Box_{11} R$, so that we have the following.

$$A \supset B \quad \text{is} \quad \Box_7 R \supset \Box_9\Box_{11} R$$

A and B are properly annotated, as was $X(P)$ from (1), and there is no annotation overlap. Then we have the following.

$$\begin{array}{lll} X(A) & \text{is} & \Box_2(\Box_1 U \supset \Box_4(\Box_3\Box_7 R \supset \Box_6 V)) \supset \Box_5 W \\ X(B) & \text{is} & \Box_2(\Box_1 U \supset \Box_4(\Box_3\Box_9\Box_{11} R \supset \Box_6 V)) \supset \Box_5 W \end{array}$$

Let us specify more of the realization function r that was partly given in (2).

$$\begin{aligned} r(7) &= k(x_3) \\ r(9) &= !k(x_3) \\ r(11) &= k(x_3) \end{aligned}$$

Then r is non self-referential on variables over $X(A)$, and we have the following.

$$r(X(A)) = x_1{:}[g(x_2, x_3, x_5){:}U \supset x_2{:}(f(x_3){:}k(x_3){:}R \supset x_3{:}V)] \supset h(x_1, x_2, x_3){:}W$$

Also $r(A) \supset r(B)$ is $k(x_3){:}R \supset !k(x_3){:}k(x_3){:}R$, which is an LP axiom.

Let c, d, and e be ground proof polynomials such that the following have injective proofs.

$$\begin{aligned} &c{:}[k(x_3){:}R \supset !k(x_3){:}k(x_3){:}R] \\ &d{:}\{[(c \cdot f(x_3)){:}!k(x_3){:}k(x_3){:}R \supset x_3{:}V] \supset [f(x_3)){:}k(x_3){:}R \supset x_3{:}V]\} \\ &e{:}\{[g(d \cdot x_2, x_3, x_5){:}U \supset x_2{:}((c \cdot f(x_3)){:}!k(x_3){:}k(x_3){:}R \supset x_3{:}V)] \supset \\ &\quad [g(d \cdot x_2, x_3, x_5){:}U \supset (d \cdot x_2){:}(f(x_3){:}k(x_3){:}R \supset x_3{:}R)]\} \end{aligned}$$

Using the algorithm of Theorem 6.2, a realization/substitution pair that hereditarily replaces $r(A)$ with $r(B)$ at P in $X(P)$ is $\langle r^*, \sigma^* \rangle$, specified as follows.

$$\begin{array}{lcllcl} r^*(1) & = & g(d \cdot x_2, x_3, x_5) & r^*(6) & = & x_3 \\ r^*(2) & = & x_1 & r^*(7) & = & k(x_3) \\ r^*(3) & = & c \cdot f(x_3) & r^*(9) & = & !k(x_3) \\ r^*(4) & = & x_2 & r^*(11) & = & k(x_3) \\ r^*(5) & = & h(e \cdot x_1, d \cdot x_2, x_3) & & & \end{array}$$

And σ^* is the identity substitution except that $\sigma^*(x_1) = e \cdot x_1$ and $\sigma^*(x_2) = d \cdot x_2$. The formula $r(X(A))\sigma^*$ is

$$(e \cdot x_1){:}[g(d \cdot x_2, x_3, x_5){:}U \supset (d \cdot x_2){:}(f(x_3){:}k(x_3){:}R \supset x_3{:}V)] \supset h(e \cdot x_1, d \cdot x_2, x_3){:}W$$

and $r^*(X(B))$ is

$$x_1{:}[g(d \cdot x_2, x_3, x_5){:}U \supset x_2{:}((c \cdot f(x_3)){:}!k(x_3){:}k(x_3){:}R \supset x_3{:}V)] \supset h(e \cdot x_1, d \cdot x_2, x_3){:}W.$$

Finally $r(X(A))\sigma^* \supset r^*(X(B))$ is formula (5) as given earlier. This example is worked out in greater detail in [5].

7 Realization Weakening

The theorems proved so far all have a non self-referentiality condition in their hypotheses. Can it be dropped? In the general case we don't know, but it can be dropped in certain special cases of interest. In fact, to give our proof of the Realization Theorem in Section 10 we only need a very special case of Theorem 5.1, and that is proved here—for it, non self-referentiality conditions can be avoided. The special case is where we replace $u_1{:}F$, $u_2{:}F$, ..., $u_N{:}F$ with the uniformly weaker $(u_1 + u_2 + \ldots + u_N){:}F$. For our proof of the Realization Theorem, we can take $N = 2$.

Theorem 7.1 (Realization Weakening) *Assume the following.*

S–1. $X(P)$ *is a properly annotated formula in which the propositional letter* P *has at most one positive occurrence;*

S–2. $\Box_p K$ *and* $\Box_q K$ *are both are properly annotated formulas, there is no annotation overlap between* $X(P)$ *and* $\Box_p K$*, and* $X(P)$ *and* $\Box_q K$*, and* p *and* q *are different;*

S–3. $r_1, r_2, \ldots, r_N$ *are realization functions that agree on* K*;*

S–4. $r_1, r_2, \ldots, r_N$ *all agree on* q*, mapping it to* $r_1(p) + r_2(p) + \ldots + r_N(p)$ *(any parenthesization will do).*

Then there is a realization/substitution pair $\langle r, \sigma \rangle$ *that hereditarily replaces each* $r_k(\Box_p K)$ *with* $r_k(\Box_q K)$ *at* P *in* $X(P)$*, for* $k = 1, 2, \ldots, N$*. Also,* σ *will live on the input positions in* $X(P)$ *and will meet the no new variable condition.*

If we set $r_1(K) = r_2(K) = \ldots = r_N(K)$ to be F, $r_1(p) = u_1$, $r_2(p) = u_2$, $\ldots$, $r_N(p) = u_n$, and $r_1(q) = r_2(q) = \ldots = r_N(q) = u_1 + u_2 + \ldots + u_N$, the theorem above provides a replacement of $u_1{:}F$, $u_2{:}F$, $\ldots$, $u_N{:}F$ with $(u_1 + u_2 + \ldots + u_N){:}F$, as promised at the beginning of this section.

The idea of the algorithm, loosely, is as follows. First determine the 'problem points,' those places where self-referentiality on variables occurs. Eliminate these problems by introducing new variables as place-holders, to replace terms that cause trouble. With this done, Theorems 4.3 and 5.1 can be applied. Then we remove the place-holding variables by substituting back the original terms for them.

Definition 7.2 Call an index n in a properly annotated formula Z *problematic* with respect to a realization function r if x_k occurs in $r(n)$ while $\Box_n$ occurs in Z within the scope of $\Box_{2k}$. Schematically, n is problematic with respect to r if somewhere in Z we have the following situation.

$$r(\overbrace{\cdots \Box_{2k}(\cdots \Box_n W \cdots)\cdots}^{Z}) = \overbrace{\cdots x_k{:}(\cdots \underbrace{r(n)}_{\text{contains } x_k}{:}\, r(W)\cdots)\cdots}^{r(Z)}$$

Problematic indexes occur when there is self-referentiality on variables. A problematic index can be made unproblematic by re-defining r on that index to be a new variable. This is the first step of the algorithm that follows.

Begin Algorithm We refer to a variable as *new* to mean it does not occur in any of $r_k(X(P))$ or $r_k(K)$ or $r_k(p)$ or $r_k(q)$, for $k = 1, \ldots, N$, and has not been introduced previously in the process of carrying out Step 1 below.

Step 1 Define new realization functions r'_1, r'_2, $\ldots r'_N$ as follows. On indexes that are not problematic, and are different than q, let r'_1 and r_1 agree, r'_2 and r_2 agree, $\ldots$, r'_N and r_N agree. The problematic indexes in $X(\Box_p K)$ are the significant ones.

First consider problematic indexes outside K. For each index n in $X(\Box_p K)$ that is problematic with respect to r_1 and that occurs outside K, introduce a new variable and set $r'_1(n)$ to be that variable; $\ldots$; for each index n that is problematic in $X(\Box_N K)$ with respect to r_N and that occurs outside K, introduce a new variable and set $r'_N(n)$ to be that variable.

Next consider problematic indexes that occur in K. It will be shown that if an index is in K and is problematic in $X(\Box_p K)$ with respect to any r_k, it is problematic with respect to all r_k. For each index n in K that is problematic in $X(\Box_p K)$ with respect to r_1, r_2, ..., r_N, introduce a new variable and set $r'_1(n) = r'_2(n) = \ldots = r'_N(n)$ to be that variable.

Call the new variables that have been introduced *place-holding* variables.

Finally, the index q cannot occur in $X(\Box_p K)$; set $r'_1(q) = r'_2(q) = \ldots = r'_N(q) = r'_1(p) + r'_2(p) + \ldots + r'_N(p)$ (parenthesized somehow).

Step 2 r'_1, r'_2, ..., r'_N are realization functions that are non self-referential on variables over $X(\Box_p K)$. It will be shown that all the hypotheses of Theorem 5.1 are met. Now apply the algorithm of that Theorem, getting a realization/substitution pair $\langle r^*, \sigma^* \rangle$ that hereditarily replaces $r'_k(\Box_p K)$ with $r'_k(\Box_q K)$ at P in $X(P)$, for $k = 1, \ldots, N$.

Step 3 Define an 'undoing' substitution σ' as follows. Let k be one of $1, \ldots, N$. If n is an index in $X(\Box_p K)$ outside K that is problematic with respect to r_k, a place-holding variable $r'_k(n)$ was introduced. On the variable $r'_k(n)$ set

$$r'_k(n)\sigma' = r_k(n)\sigma^*$$

If n is an index in K that is problematic in $X(\Box_p K)$ with respect to any (all) r_k a place-holding variable $r'_k(n)$ was introduced (the same for all k). On this variable $r'_k(n)$ set

$$r'_k(n)\sigma' = r_k(n)\sigma^*$$

(where only the variable and not the choice of k will be shown to matter).

Finally, if x is not one of the place-holding variables, set

$$x\sigma' = x$$

Step 4 Define a realization function by setting $r(n) = r^*(n)\sigma'$ for indexes in $X(P)$, $\Box_p K$, and $\Box_q K$, and arbitrarily otherwise (except that even indexes must map to proof variables). The realization/substitution pair we want is $\langle r, \sigma^* \rangle$.

End Algorithm

Begin Correctness Proof First we show that if n is a problematic index in some properly annotated formula Z with respect to a realization function r then n must be odd. Here is the simple argument. If n is a problematic index, $\Box_n$ occurs in the scope of $\Box_{2k}$ and $r(n)$ contains x_k. If n were even, say $2m$, then $r(n) = x_m$ since r is a realization function. Then if $r(n)$ contains x_k it must be that $x_m = x_k$, hence $m = k$, and so $n = 2m = 2k$. But then the same index would occur twice in Z, which is impossible since Z is properly annotated.

Now, assume *S–1* to *S–4*. Suppose n is an index in K that is problematic in $X(\Box_p K)$ with respect to r_i. Since $r_i(n) = r_j(n)$ by *S–3*, and all realization functions agree on even indexes, it follows immediately that n is problematic with respect to r_j as well.

We thus have the background needed for Step 1 of the algorithm. Carry that step out.

Since problematic indexes must be odd, if we change a realization function by redefining it on problematic indexes, the result will be another realization function, since it will still map even

indexes to variables. This ensures that the functions r'_k defined in Step 1 are all realization functions. And since we have removed problematic indexes by introducing new variables, it is easy to see that $r'_1, r'_2, \ldots, r'_N$ are non self-referential on variables over $X(\Box_p K)$. We check that the other conditions of Theorems 4.3 and 5.1 are met.

$r'_1, r'_2, \ldots, r'_N$ agree on K by construction and by *S-3*. Also they agree on q. Consequently $r'_1(\Box_q K) = r'_2(\Box_q K) = \ldots = r'_N(\Box_q K)$.

For each $k = 1, 2, \ldots, N$, $r'_k(\Box_p K) \supset r'_k(\Box_q K)$ is the formula

$$r'_k(p){:}r'_k(K) \supset (r'_1(p) + r'_2(p) + \ldots + r'_N(p)){:}r'_k(K)$$

This is an LP theorem using the Sum Axiom.

We have shown enough to verify that conditions *H–1* to *H–4* of Theorems 4.3 and 5.1 are met, using $r'_1, r'_2, \ldots, r'_N$, with $A = \Box_p K$ and $B = \Box_q K$.

Now apply Step 2 of the algorithm, producing a realization/substitution pair $\langle r^*, \sigma^* \rangle$ such that σ^* lives on input positions in $X(P)$ and meets the no new variable condition, and $\langle r^*, \sigma^* \rangle$ hereditarily replaces $r'_k(\Box_p K)$ with $r'_k(\Box_q K)$ at P in $X(P)$, for each $k = 1, 2, \ldots, N$.

Apply Step 3 of the algorithm, re-introducing the proof polynomials that were eliminated earlier. We now have a substitution, σ'. There is a minor point involving the case where n is an index in K that is problematic—we must verify that this case is well-defined. The variable $r'_k(n)$ is the same for all k, by the construction in Step 1. But also, $r_k(n)\sigma^*$ is the same for all k, since $r_1, \ldots, r_N$ all agree on indexes in K by condition *S–3*. This ensures well-definedness.

Before going on to Step 4 it will be convenient to introduce yet another substitution, σ'', closely related to σ'. It does not play a role in the algorithm itself, but does play a significant role in verifying its correctness.

On variables that are not place-holding, set σ'' to be the identity. And on place-holding variables we proceed as follows.

If n is an index in $X(\Box_p K)$ outside K that is problematic with respect to r_k, a new place-holding variable $r'_k(n)$ was introduced in Step 1 of the algorithm. On this variable set $r'_k(n)\sigma'' = r_k(n)$.

If n is an index in K that is problematic in $X(\Box_p K)$ with respect to any (all) of $r_1, \ldots, r_N$, a new place-holding variable $r'_k(n)$ was introduced in Step 1 (the same for all k). On this variable set $r'_k(n)\sigma'' = r_k(n)$. (The choice of k doesn't matter, by *S–3*.)

An important fact concerning σ'' is this, for each $k = 1, \ldots, N$.

$$r'_k(Z)\sigma'' = r_k(Z) \text{ for } Z \text{ a subformula of } X(\Box_p K) \tag{6}$$

Here is the argument for (6): we show that for every index n in $X(\Box_p K)$, $r'_k(n)\sigma'' = r_k(n)$. Suppose first that n is a problematic index in $X(\Box_p K)$ (whether in K or not). Then $r'_k(n)$ is a place-holding variable, so by definition of σ'' we have $r'_k(n)\sigma'' = r_k(n)$. If n is not problematic, $r'_k(n) = r_k(n)$. Since all place-holding variables were new, none can occur in $r_k(n)$, and on non place-holding variables σ'' is the identity. So for n not problematic, $r'_k(n)\sigma'' = r_k(n)\sigma'' = r_k(n)$.

Another key fact concerning σ'' is the following 'semi-commutativity' result.

$$\sigma^*\sigma' = \sigma''\sigma^* \tag{7}$$

In the verification of (7) there are three cases to consider.

First, suppose x is a variable that is not place-holding. Then $x\sigma''\sigma^* = x\sigma^*$ because σ'' is the identity on such variables. Also $x\sigma^*\sigma' = x\sigma^*$ because σ^* meets the no new variable condition,

hence $x\sigma^*$ contains only the variable x, on which σ' is the identity. Hence $x\sigma^*\sigma' = x\sigma''\sigma^*$ in this case.

Next consider a place-holding variable, say $r'_k(n)$ where n is a problematic index in $X(\Box_p K)$ occurring outside K. For this, $r'_k(n)\sigma^*\sigma' = r'_k(n)\sigma'$ since σ^* lives on input positions in $X(\Box_p K)$ and $r'_k(n)$ was a new variable, on which σ^* must be the identity. Further, $r'_k(n)\sigma' = r_k(n)\sigma^*$ by definition of σ'. But also $r'_k(n)\sigma''\sigma^* = r_k(n)\sigma^*$ by definition of σ''. Hence $r'_k(n)\sigma^*\sigma' = r'_k(n)\sigma''\sigma^*$ in this case.

The case of a place-holding variable arising from a problematic index occurring in K is similar. Since this covers all the cases, we have established that $\sigma^*\sigma' = \sigma''\sigma^*$.

Finally we come to Step 4 of the algorithm. Define the function r as specified in the algorithm. This is actually a realization function since if $2n$ is an even index in $X(P)$, $\Box_p K$, or $\Box_q K$, $r(2n) = r^*(2n)\sigma' = x_n\sigma' = x_n$, since σ' is the identity on variables that are not place-holders, and place-holding variables were all new. Of course r may be self-referential on variables, but that won't matter now.

It is immediate from the definition of r that we have the following, for each $k = 1, \ldots, N$.

$$r^*(Z)\sigma' = r(Z) \text{ for } Z \text{ a subformula of } X(\Box_q K) \tag{8}$$

We must show that $\langle r, \sigma^* \rangle$ hereditarily replaces each $r_k(\Box_p K)$ with $r_k(\Box_q K)$ at P in $X(P)$, for $k = 1, \ldots, N$. To see this, let $\varphi(P)$ be a subformula of $X(P)$, say it is positive; the argument is similar if it is negative. What must be shown is the injective provability of the following.

$$r_k(\varphi(\Box_p K))\sigma^* \supset r(\varphi(\Box_q K)) \tag{9}$$

Since $\langle r^*, \sigma^* \rangle$ hereditarily replaces $r'_k(\Box_p K)$ with $r'_k(\Box_q K)$ at P in $X(P)$, we have the injective provability of the following.

$$r'_k(\varphi(\Box_p K))\sigma^* \supset r^*(\varphi(\Box_q K))$$

Then by Theorem 2.3 we also have the following.

$$r'_k(\varphi(\Box_p K))\sigma^*\sigma' \supset r^*(\varphi(\Box_q K))\sigma'$$

By (7) this gives us the following.

$$r'_k(\varphi(\Box_p K))\sigma''\sigma^* \supset r^*(\varphi(\Box_q K))\sigma'$$

But $r'_k(\varphi(\Box_p K))\sigma'' = r_k(\varphi(\Box_p K))$ by (6), and $r^*(\varphi(\Box_q K))\sigma' = r(\varphi(\Box_q K))$ by (8), hence we have the following,

$$r_k(\varphi(\Box_p K))\sigma \supset r(\varphi(\Box_q K))$$

which is formula (9).

End Correctness Proof

The problem in stating a more general version of the result above lies in the definition of r'_1, $\ldots$, r'_N. They were chosen to remove self-referentiality on variables. But then, it is not enough that $r_1(A) \supset r_1(B)$, $\ldots$, $r_2(A) \supset r_2(B)$ be injectively provable, we also need that $r'_1(A) \supset r'_1(B)$, $\ldots$, $r'_2(A) \supset r'_2(B)$ be. In the special case considered above, this was carefully arranged to produce axioms. In other cases we might not be so fortunate. There is no statement of a general case attempted here, but it is not needed for our present purposes.

8 The Realization Merging Theorem

It may happen that several realization functions arise that are applied to the same annotated formula. The results of this section show that they may be merged into one. An important thing to note about the theorem is that it does not have a non self-referentiality condition on variables.

Definition 8.1 Let X be an annotated formula and r_1, r_2, ..., r_N be realization functions.

1. For a subformula φ of X, we say a realization/substitution pair $\langle r, \sigma\rangle$ *merges* r_1, ..., r_N *on* φ *in* X provided, for each $k = 1, \ldots, N$:
   (a) if φ is a positive subformula of X then $r_k(\varphi)\sigma \supset r(\varphi)$ is an injective theorem of LP;
   (b) if φ is a negative subformula of X then $r(\varphi) \supset r_k(\varphi)\sigma$ is an injective theorem of LP.
2. We say $\langle r, \sigma\rangle$ *hereditarily merges* r_1, ..., r_N on X provided, for each subformula φ of X, $\langle r, \sigma\rangle$ merges r_1, ..., r_N on φ in X.

Theorem 8.2 (Realization Merging) *Let X be a properly annotated formula, and r_1, ..., r_N be realization functions. Then there is a realization/substitution pair $\langle r, \sigma\rangle$ that hereditarily merges r_1, ..., r_N on X. Further, σ will live on the input positions in X, and will meet the no new variable condition.*

Begin Algorithm Let P be a propositional letter not in X, let p and q be distinct odd indexes not in X, and let K be P. Now apply the algorithm of Theorem 7.1.
End Algorithm

Begin Correctness Proof Assume the hypothesis. Let P, p, q, and K be as in the algorithm. Without loss of generality we can assume $r_1(q) = \ldots, = r_N(q) = r_1(p) + \ldots + r_N(p)$, since neither p nor q occur in X. Then conditions *S–1* to *S–4* of Theorem 7.1 are met (some of them rather vacuously). It follows that there is a realization/substitution pair $\langle r, \sigma\rangle$ that hereditarily replaces each $r_k(\Box_p K)$ with $r_k(\Box_q K)$ at P in $X(P)$. Since P does not occur in X, this simply amounts to saying $\langle r, \sigma\rangle$ hereditarily merges r_1, ..., r_N on X. We also have that σ lives on input positions in X and meets the no new variable condition.
End Correctness Proof

Example 8.3 Here is a simple example illustrating Theorem 8.2. Let X be the annotated formula $\Box_2 A \supset (\Box_4 B \supset \Box_1 C)$, where A, B, and C are atomic. This is properly annotated, indeed trivially so since there are no nested modal operators. And let r_1 and r_2 be realization functions such that

$$\begin{array}{lcllcl} r_1(1) & = & f(x_1, x_2) & r_2(1) & = & g(x_1, x_2) \\ r_1(2) & = & x_1 & r_2(2) & = & x_1 \\ r_1(4) & = & x_2 & r_2(4) & = & x_2 \end{array}$$

As it happens, both r_1 and r_2 are non self-referential on variables over X. Here is a realization/substitution pair $\langle r, \sigma\rangle$ that hereditarily merges r_1 and r_2 on X.

Let a, b, c be ground proof polynomials such that $a{:}(A \supset A)$, $b{:}(B \supset B)$, and $c{:}(C \supset C)$ have injective proofs. For the substitution σ we have the following.

$$\begin{array}{l} x_1\sigma = a \cdot x_1 \\ x_2\sigma = b \cdot x_2 \\ x_n\sigma = x_n \text{ otherwise} \end{array}$$

And for the realization function we have the following.

$$\begin{aligned} r(1) &= [c \cdot f(a \cdot x_1, b \cdot x_2) + c \cdot g(a \cdot x_1, b \cdot x_2)] \\ r(2) &= x_1 \\ r(4) &= x_2 \end{aligned}$$

The final substitution and realization functions meet the conditions of Theorem 8.2.

In the Introduction we discussed a problem that can now be resolved. For starters, with annotated formulas available the problem can be restated more conveniently. Suppose $(A \vee B) \supset C$ is a properly annotated formula (with $\vee$ taken as an abbreviation). And suppose we have realization functions r_1 and r_2 such that $r_1(A \supset C)$ and $r_2(B \supset C)$ are both theorems of LP. Thus $r_1(A \supset C)$ and $r_2(B \supset C)$ both embody correct instances of our reasoning, using explicit justifications. Is there a realization function r such that $r((A \vee B) \supset C)$ is a theorem of LP?

One can give an immediate answer of 'yes' to the question just asked. The argument goes as follows. Let A_0, B_0 and C_0 be unannotated versions of A, B, and C. Since $r_1(A \supset C)$ and $r_2(B \supset C)$ are theorems of LP, then $A_0 \supset C_0$ and $B_0 \supset C_0$ are theorems of S4. But then so is $(A_0 \vee B_0) \supset C_0$, so by the Realization Theorem 3.6, there is a realization function r such that $r((A \vee B) \supset C)$ is a theorem of LP.

The problem with the answer just given is that the realization function r need not have any relationship with r_1 and r_2. By passing through the forgetful projection and the Realization Theorem we have, in effect, started fresh. So a better question would be: is there a realization function r such that $r((A \vee B) \supset C)$ is a theorem of LP where r is, in some natural way, constructed from r_1 and r_2, thus making use of the explicit justifications we had. The answer is 'yes.' We can use Theorem 8.2. Here is the argument.

For convenience, let X be the annotated formula $(A \vee B) \supset C$. By Theorem 8.2, there is a realization/substitution pair $\langle r, \sigma \rangle$ that hereditarily merges r_1 and r_2 on X. We claim $r((A \vee B) \supset C)$ is a theorem of LP. To see this, first note that since A and B are negative subformulas of X, and C is a positive subformula, the following are theorems of LP.

$$\begin{aligned} r(A) &\supset r_1(A)\sigma \\ r(B) &\supset r_2(B)\sigma \\ r_1(C)\sigma &\supset r(C) \\ r_2(C)\sigma &\supset r(C) \end{aligned}$$

Also, by assumption, we have as LP theorems the following.

$$\begin{aligned} r_1(A) &\supset r_1(C) \\ r_2(B) &\supset r_2(C) \end{aligned}$$

and hence by Theorem 2.3, we also have the following.

$$\begin{aligned} r_1(A)\sigma &\supset r_1(C)\sigma \\ r_2(B)\sigma &\supset r_2(C)\sigma \end{aligned}$$

Putting all this together, we have the following derivation.

$$\begin{aligned} r(A \vee B) &= r(A) \vee r(B) \\ &\supset r_1(A)\sigma \vee r_2(B)\sigma \\ &\supset r_1(C)\sigma \vee r_2(C)\sigma \\ &\supset r(C) \vee r(C) \\ &\supset r(C) \end{aligned}$$

The point is that r is directly calculated from r_1 and r_2. If LP and its relatives are to become viable logics of justifications, the ability to make use of already acquired justifications in subsequent arguments is fundamental. The example just given shows a way that this can sometimes be done.

9 Gentzen Sequent Calculi

Algorithmic proofs of the Realization Theorem have always been based on a cut free proof procedure for S4. Either a sequent calculus or a tableau system can be used—essentially one is the other run backwards. In the history of LP it is sequent formulations that have been the general standard, so that is what will be used here. We need versions for propositional S4, and for an annotated version of it.

For S4 the following calculus will be used. A *sequent* for S4 is a pair of finite multisets of formulas in the language $L_\Box$, where the pair is written $\Gamma \longrightarrow \Delta$, with Γ and Δ being multisets. Using multisets avoids the need for explicit permutation rules. Axioms are the following sequents, where P is any propositional letter.

$$P \longrightarrow P \qquad\qquad \bot \longrightarrow$$

Then the rules of derivation are as follows. In stating them, Γ and Δ are multisets, X and Y are formulas, and if $\Gamma = \{Y_1, \ldots, Y_k\}$ then $\Box\Gamma = \{\Box Y_1, \ldots, \Box Y_k\}$.

$$LW \quad \frac{\Gamma \longrightarrow \Delta}{\Gamma, X \longrightarrow \Delta} \qquad RW \quad \frac{\Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta, X}$$

$$LC \quad \frac{\Gamma, X, X \longrightarrow \Delta}{\Gamma, X \longrightarrow \Delta} \qquad RC \quad \frac{\Gamma \longrightarrow \Delta, X, X}{\Gamma \longrightarrow \Delta, X}$$

$$L\supset \quad \frac{\Gamma, Y \longrightarrow \Delta \quad \Gamma \longrightarrow \Delta, X}{\Gamma, X \supset Y \longrightarrow \Delta} \qquad R\supset \quad \frac{\Gamma, X \longrightarrow \Delta, Y}{\Gamma \longrightarrow \Delta, X \supset Y}$$

$$L\Box \quad \frac{\Gamma, X \longrightarrow \Delta}{\Gamma, \Box X \longrightarrow \Delta} \qquad R\Box \quad \frac{\Box\Gamma \longrightarrow X}{\Box\Gamma \longrightarrow \Box X}$$

As usual, a proof of a formula X in this calculus is a proof of the sequent $\longrightarrow X$. This is a standard sequent calculus for S4, and soundness and completeness arguments are well-known in the literature.

We also need a version of this sequent calculus for *annotated* formulas. Except for the two modal rules, all the axioms and rules have exactly the same form *but formulas are from* $L^a_\Box$. Thus

annotations must be preserved in moving from sequents above the line to sequents below the line. The modal rules become the following.

$$L\Box^a \quad \frac{\Gamma, X \longrightarrow \Delta}{\Gamma, \Box_{2n} X \longrightarrow \Delta} \qquad\qquad R\Box^a \quad \frac{\Box_{2n_1} Y_1, \ldots, \Box_{2n_k} Y_k \longrightarrow X}{\Box_{2n_1} Y_1, \ldots, \Box_{2n_k} Y_k \longrightarrow \Box_{2p+1} X}$$

Both sequent calculi have the *subformula property*. If X has a proof, it has one in which every formula that appears is a subformula of X. Indeed, something stronger will be needed: if X has a proof, it has one in which every formula on the left of an arrow is a negative subformula of X and every formula on the right of an arrow is a positive subformula of X. This is well-known for the S4 calculus, and can easily be established for the annotated version. It plays a central role in the proof of the Realization Theorem given in Section 10.

There is one more fundamental result that we will need, concerning the annotated calculus. Since it is not standard, it is set forth more formally.

Proposition 9.1 *If Z is a formula of $L_\Box$ that is a theorem of S4, and if X is any properly annotated version of Z, then X has a proof in the annotated Gentzen calculus.*

Here is a sketch of the verification of this Proposition. If one takes a proof of Z in the (unannotated) sequent calculus, one can use this to construct an annotated proof of X. Replace the final $\longrightarrow Z$ with $\longrightarrow X$, then step-by-step propagate the annotations upward, from conclusions of rules to premises, until the entire sequent construction has been annotated. A formal version of this amounts to an induction on the number of sequents in the unannotated proof, and is omitted.

10 The Realization Theorem

The goal of this section is to give a proof, using the machinery developed in earlier sections, of Artemov's Realization Theorem. For convenience we repeat the earlier statement of it.

Theorem 3.6 If Z_0 is a theorem of S4, there is a realization of Z_0 that is an injectively provable theorem of LP. In fact, if Z_0 is a theorem of S4, then for any properly annotated version Z of Z_0 there is a realization function r such that $r(Z)$ is injectively provable in LP.

Begin Algorithm If Z is a properly annotated version of Z_0, and Z_0 is a theorem of S4, Z will have a sequent calculus proof in the system with annotations. We construct a realization function for each sequent in a proof $\mathcal{P}$ of Z—roughly, a realization of a sequent is a realization of the formula that says the conjunction of the left of the sequent implies the disjunction of the right of the sequent. The realization function for the final sequent in $\mathcal{P}$ is the desired realization function.

For sequents that are axioms, any realization function will do.

For all rules except $L\supset$ and $R\Box^a$, if r realizes the premise of a sequent rule, r will also realize the conclusion.

For the $L\supset$ rule, if r_1 realizes $\Gamma, Y \rightarrow \Delta$ and r_2 realizes $\Gamma \rightarrow \Delta, X$, then r will realize $\Gamma, X \supset Y \rightarrow \Delta$, where r is the merging of r_1 and r_2 using the algorithm of Theorem 8.2.

For the $R\Box^a$ rule, suppose r_0 realizes $\Box_{2n_1} Y_1, \ldots, \Box_{2n_k} Y_k \longrightarrow X$. We must realize $\Box_{2n_1} Y_1, \ldots, \Box_{2n_k} Y_k \longrightarrow \Box_m X$.

Since r_0 realizes $\Box_{2n_1}Y_1, \ldots, \Box_{2n_k}Y_k \longrightarrow X$, there is an LP proof of

$$(x_{n_1}{:}r_0(Y_1) \wedge \ldots \wedge x_{n_k}{:}r_0(Y_k)) \supset r_0(X).$$

Then by the Lifting Lemma there is a proof polynomial $t(x_{n_1}, \ldots, x_{n_k})$ such that

$$(x_{n_1}{:}r_0(Y_1) \wedge \ldots \wedge x_{n_k}{:}r_0(Y_k)) \supset t(x_{n_1}, \ldots, x_{n_k}){:}r_0(X)$$

is provable.

The realization r that we want is like the original r_0 except that $r(m)$ is the result of replacing $r_0(m)$ with $r_0(m) + t(x_{n_1}, \ldots, x_{n_k})$. The following makes this precise.

Let p and q be distinct new odd indexes. Let r_1 and r_2 be the same as r_0 except that $r_1(p) = r_0(m)$, $r_2(p) = t(x_{n_1}, \ldots, x_{n_k})$, and $r_1(q) = r_2(q) = r_1(p) + r_2(p)$. Let P be a new propositional letter, and let $Z(P)$ be Z with $\Box_m X$ replaced with P. Use the algorithm of Theorem 7.1 to hereditarily replace $r_1(\Box_p X)$ with $r_1(\Box_q X)$ and $r_2(\Box_p X)$ with $r_2(\Box_q X)$ at P in $Z(P)$, getting a realization/substitution pair $\langle r^*, \sigma^* \rangle$. Finally, let r be like r^* except that $r(m) = r^*(q)$. Then r realizes $\Box_{2n_1}Y_1, \ldots, \Box_{2n_k}Y_k \longrightarrow \Box_m X$.

End Algorithm

Begin Correctness Proof Assume Z_0 is a theorem of S4. Let Z be any properly annotated version of Z_0. By Proposition 9.1 there is a proof of $\rightarrow Z$ in the annotated Gentzen calculus—call the proof $\mathcal{P}$. We need a connection between sequents and formulas. This was stated informally in the algorithm, now we make it precise.

For each sequent S of annotated formulas, a corresponding annotated formula $\|S\|$ is defined.

1. $\|X_1, \ldots, X_n \longrightarrow Y_1, \ldots, Y_m\|$ is the annotated formula $(X_1 \wedge \ldots \wedge X_n) \supset (Y_1 \vee \ldots \vee Y_m)$.
2. $\|X_1, \ldots, X_n \rightarrow \|$ is the annotated formula $(X_1 \wedge \ldots \wedge X_n) \supset \bot$.
3. $\| \longrightarrow Y_1, \ldots, Y_k\|$ is the annotated formula $(Y_1 \vee \ldots \vee Y_k)$.

Let us say an annotated sequent S is *realized* if there is a realization function r such that $r(\|S\|)$ is injectively provable in LP. The algorithm supplies a realization function for each sequent S in proof $\mathcal{P}$. Then this will be the case for the final sequent, $\longrightarrow Z$, but $\| \longrightarrow Z\|$ is simply Z, and so the existence of a realization function r such that $r(Z)$ is injectively provable in LP is established. To show each sequent S in $\mathcal{P}$ is realized, we show the correctness of the steps outlined in the algorithm.

The ground case, axioms, is trivial. Sequents that are axioms have no modal operators, so any realization function will do.

We next must show that each of the rules of inference preserves being realized. This is immediate for most of them. Take RC as a representative example. If r is a realization function such that $r(\|\Gamma \longrightarrow \Delta, X, X\|)$ has an injective LP proof, it is easy to see that $r(\|\Gamma \longrightarrow \Delta, X\|)$ will also have an injective LP proof. All rules except for $L\supset$ and $R\Box^a$ follow this pattern.

Of the two hard cases, we treat $L\supset$ first. Suppose both $\Gamma, Y \longrightarrow \Delta$ and $\Gamma \longrightarrow \Delta, X$ are realized. We must show that $\Gamma, X \supset Y \longrightarrow \Delta$ is also realized. Of course $\Gamma, Y \longrightarrow \Delta$ and $\Gamma \longrightarrow \Delta, X$ might be realized by different realization functions. Assume we have a realization function r_1 such that $r_1(\|\Gamma, Y \longrightarrow \Delta\|)$ has an injective LP proof, and a possibly different realization function r_2 such that $r_2(\|\Gamma \longrightarrow \Delta, X\|)$ has an injective LP proof. By the Realization Merging Theorem 8.2, there is a realization/substitution pair $\langle r, \sigma \rangle$ that hereditarily merges r_1 and r_2 on Z. We show that $r(\|\Gamma, X \supset Y \longrightarrow \Delta\|)$ is injectively provable.

Suppose φ is an annotated formula on the left in one of the two premise sequents of the $L\supset$ rule application. That is, φ is in Γ or is the formula Y. Then φ is a negative subformula of Z, so both $r(\varphi)\supset r_1(\varphi)\sigma$ and $r(\varphi)\supset r_2(\varphi)\sigma$ are injectively provable. Also if φ on the right, that is, if φ is a member of Δ or is X, it must be a positive subformula of Z and so both $r_1(\varphi)\sigma\supset r(\varphi)$ and $r_2(\varphi)\sigma\supset r(\varphi)$ are injectively provable. As a consequence of all this, $r_1(\|\Gamma, Y\to\Delta\|)\sigma\supset r(\|\Gamma, Y\to\Delta\|)$ and $r_2(\|\Gamma\to\Delta, X\|)\sigma\supset r(\|\Gamma\to\Delta, X\|)$ are easily seen to be injectively provable. By the Substitution Lemma 2.3, both $r_1(\|\Gamma, Y\longrightarrow\Delta\|)\sigma$ and $r_2(\|\Gamma\longrightarrow\Delta, X\|)\sigma$ are injectively provable. It follows that both $r(\|\Gamma, Y\to\Delta\|)$ and $r(\|\Gamma\to\Delta, X\|)$ are injectively provable. Now we have a single realization function, r, that works for both sequents, and it is an easy task to show that $r(\|\Gamma, X\supset Y\longrightarrow\Delta\|)$ is injectively provable.

Finally we consider the case $R\Box^a$. Suppose $\Box_{2n_1}Y_1,\ldots,\Box_{2n_k}Y_k\longrightarrow X$ is realized, say using the realization function r_0. We show $\Box_{2n_1}Y_1,\ldots,\Box_{2n_k}Y_k\longrightarrow\Box_m X$ is realized, where both these sequents occur in proof $\mathcal{P}$. Note that the indexes $2n_i$ are all even, while m must be odd. The hypothesis is that the following formula is injectively provable

$$(x_{n_1}{:}r_0(Y_1)\wedge\ldots\wedge x_{n_k}{:}r_0(Y_k))\supset r_0(X) \tag{10}$$

We must produce a realization function r such that $(x_{n_1}{:}r(Y_1)\wedge\ldots\wedge x_{n_k}{:}r(Y_k))\supset r(m){:}r(X)$ is injectively provable.

Using (10) and the Lifting Lemma, Theorem 2.4, there is a proof polynomial $t(x_{n_1},\ldots,x_{n_k})$ such that

$$(x_{n_1}{:}r_0(Y_1)\wedge\ldots\wedge x_{n_k}{:}r_0(Y_k))\supset t(x_{n_1},\ldots,x_{n_k}){:}r_0(X) \tag{11}$$

is injectively provable.

A natural attempt at creating an appropriate realization function would be to define r to be the same as r_0 except that $r(m)=t(x_{n_1},\ldots,x_{n_k})$. Now Z is properly annotated and $\Box_m X$ is a subformula, so m does not occur in X. Thus we would have $r(X)=r_0(X)$, and hence we would have the injective provability of the following.

$$(x_{n_1}{:}r_0(Y_1)\wedge\ldots\wedge x_{n_k}{:}r_0(Y_k))\supset r(m){:}r(X)$$

But this does not ensure we would have the injective provability of the thing we need,

$$(x_{n_1}{:}r(Y_1)\wedge\ldots\wedge x_{n_k}{:}r(Y_k))\supset r(m){:}r(X)$$

because $\Box_m$ may have occurrences in one or more of $Y_1,\ldots,Y_k$, and so r_0 and r may not be the same on them. The possible presence of $r_0(m)$ in $(x_{n_1}{:}r_0(Y_1)\wedge\ldots\wedge x_{n_k}{:}r_0(Y_k))$ must be dealt with, and we do this by realizing $\Box_m$ with $r_0(m)+t(x_{n_1},\ldots,x_{n_k})$ rather than just $t(x_{n_1},\ldots,x_{n_k})$. Demonstrating that this yields a provable sequent brings Theorem 7.1 into play, and this is the source of some complications. In order to apply Theorem 7.1 we need distinct indexes p and q. In our application of the Theorem here both are, in a sense, stand-ins for m. The formal argument below amounts to introducing new indexes p and q, applying Theorem 7.1, then eliminating the indexes at the end.

Both $\Box_{2n_1}Y_1,\ldots,\Box_{2n_k}Y_k\longrightarrow X$ and $\Box_{2n_1}Y_1,\ldots,\Box_{2n_k}Y_k\longrightarrow\Box_m X$ are sequents in a proof, $\mathcal{P}$, of the formula Z, so all formulas in these sequents are subformulas of Z. Then $\Box_m X$ occurs as a (positive) subformula of Z exactly once, since Z is properly annotated. Let P be a propositional letter that does not occur in Z, and let $Z(P)$ be like Z except that the subformula $\Box_m X$ has been replaced with P. Then P must have a single positive occurrence in $Z(P)$, and Z is the same as $Z(\Box_m X)$. Also, no index in $\Box_m X$ can occur in $Z(P)$, again since $Z=Z(\Box_m X)$ is properly annotated.

Let p and q be distinct odd indexes that do not occur in Z. Then $Z(P)$, $\Box_p X$, and $\Box_q X$ are properly annotated, $Z(P)$ and $\Box_p X$ have no annotation overlap, and $Z(P)$ and $\Box_q X$ have no annotation overlap.

Define two realization functions r_1 and r_2 to be the same as r_0, except that $r_1(p) = r_0(m)$, $r_2(p) = t(x_{n_1}, \ldots, x_{n_k})$ and $r_1(q) = r_2(q) = r_0(m) + t(x_{n_1}, \ldots, x_{n_k})$. Note that $r_1(X) = r_2(X) = r_0(X)$ since p and q do not occur in X.

Now we can apply Theorem 7.1, Realization Weakening. There is a realization/substitution pair $\langle r^*, \sigma^* \rangle$ that hereditarily replaces $r_1(\Box_p X)$ with $r_1(\Box_q X)$ at P in $Z(P)$ and hereditarily replaces $r_2(\Box_p X)$ with $r_2(\Box_q X)$ at P in $Z(P)$. The realization function r^* is almost the one we want. That is, we will show that

$$(x_{n_1}{:}r^*(Y_1(\Box_q X)) \wedge \ldots \wedge x_{n_k}{:}r^*(Y_k(\Box_q X))) \supset r^*(\Box_q X) \tag{12}$$

has a sequent calculus proof. The difficulty is the presence of the index q. The last step will be to eliminate it.

Consider one of the formulas on the left of the sequent $\Box_{2n_1} Y_1, \ldots, \Box_{2n_k} Y_k \longrightarrow \Box_m X$, say $\Box_{2n_i} Y_i$. This is a negative subformula of Z. Let us write $\Box_{2n_i} Y_i(P)$ for the result of replacing the subformula $\Box_m X$ in $\Box_{2n_i} Y_i$ with P (the occurrence of $\Box_m X$ may be vacuous). Then $\Box_{2n_i} Y_i(P)$ is a subformula of $Z(P)$ amd $\Box_{2n_i} Y_i$ is the same thing as $\Box_{2n_i} Y_i(\Box_m X)$. (Note that $2n_i$ and m cannot be the same since one index is even and the other is odd.) Since we have hereditary replacement of $r_1(\Box_p X)$ with $r_1(\Box_q X)$ at P using $\langle r^*, \sigma^* \rangle$, we have the injective provability of the following.

$$r^*(\Box_{2n_i} Y_i(\Box_q X)) \supset r_1(\Box_{2n_i} Y_i(\Box_p X))\sigma^* \tag{13}$$

r_1 is the same as r_0 except on p and q. Now q does not occur in $\Box_{2n_i} Y_i(\Box_p X)$, p does not occur in $Y_i(P)$ or in X, and $r_1(p) = r_0(m)$, so $r_1(\Box_{2n_i} Y_i(\Box_p X)) = r_0(\Box_{2n_i} Y_i(\Box_m X)) = r_0(\Box_{2n_i} Y_i)$. Since r^* and r_0 are realization functions, $r^*(\Box_{2n_i}) = r_0(\Box_{2n_i}) = x_{n_i}$. Then from (13) we have the following, for each $i = 1, \ldots, k$.

$$x_{n_i}{:}r^*(Y_i(\Box_q X)) \supset x_{n_i}{:}r_0(Y_i)\sigma^* \tag{14}$$

The formula P itself is a positive subformula of $Z(P)$. Since $\langle r^*, \sigma^* \rangle$ hereditarily replaces $r_2(\Box_p X)$ with $r_2(\Box_q X)$ at P in $Z(P)$, we have injective provability of

$$r_2(\Box_p X)\sigma^* \supset r^*(\Box_q X) \tag{15}$$

Since $r_2(\Box_p X) = r_2(p){:}r_2(X) = t(x_{n_1}, \ldots, x_{n_k}){:}r_0(X)$, from (15) we have the following.

$$t(x_{n_1}, \ldots, x_{n_k}){:}r_0(X)\sigma^* \supset r^*(\Box_q X) \tag{16}$$

Now, from (11), using the Substitution Lemma, Theorem 2.3, we have the following.

$$(x_{n_1}{:}r_0(Y_1)\sigma^* \wedge \ldots \wedge x_{n_k}{:}r_0(Y_k)\sigma^*) \supset t(x_{n_1}, \ldots, x_{n_k}){:}r_0(X)\sigma^* \tag{17}$$

Combining (14), (16) and (17) we have injective provability of the following.

$$(x_{n_1}{:}r^*(Y_1(\Box_q X)) \wedge \ldots \wedge x_{n_k}{:}r^*(Y_k(\Box_q X))) \supset r^*(\Box_q X) \tag{18}$$

That is, we have succeeded in showing (12) as promised.

Now we are in the final phase, eliminating q in favor of m. The index q does not occur in X since it was chosen to be a new index, and m does not occur in X because $\Box_m X$ is a subformula of the properly annotated formula Z. Let r be like r^* except that $r(m) = r^*(q)$. Then

$r^*(\Box_q X) = r(\Box_m X)$. Likewise q does not occur in $Y_i(P)$, and neither does m, since $Y_i(P)$ is Y_i with the subformula $\Box_m X$ replaced with P. Then $r^*(Y_i(\Box_q X)) = r(Y_i(\Box_m X)) = r(Y_i)$. So, from (18) we have the following.

$$(x_{n_1}{:}r(Y_1) \wedge \ldots \wedge x_{n_k}{:}r(Y_k) \supset r(\Box_m X) \tag{19}$$

And this shows that the consequent of rule $R\Box^a$ is realized.
End Correctness Proof

11 Conclusion

The logic LP is now understood to be one of a family of related justification logics—it corresponds to S4. It can be weakened by dropping the ! operator and its axiom, yielding what is known as LP(T). In the other direction it can be strengthened by adding a *negative* proof checker, dual to !, yielding a logic called LP(S5), see [9] for instance. Likewise it can be extended to incorporate multiple agents, and even common knowledge. How much of the work in this paper extends to these logics? In this generality further investigation is needed, especially when multiple agents are involved. But here is an indication that the results very likely will extend.

The main theorems of this paper are Theorem 4.3 and 5.1; other results are corollaries of these. The essential features of LP that are used in proving these theorems are the Substitution Lemma and the Lifting Lemma. The logic LP(S5) satisfies both of these. Consequently the main theorems of this paper extend to LP(S5) as well. There is a (somewhat peculiar) sequent calculus for LP(S5) in [3] (actually, it is presented as a tableau system, but this is a minor difference). A Realization Theorem for LP(S5) can then be shown, along the same lines as the one for S4 was shown here. Recently the Realization Theorem for LP(S5) was shown in [9], by a semantic argument. Present methods provide a second, proof-theoretic, argument as well.

S4 is not only a modal logic, it is also well-known as a logic of knowledge in the Hintikka tradition—a logic with a single knower having positive introspection. Then LP can be seen as that logic of knowledge made explicit, with "known to be the case" replaced by "known because of such-and-such evidence". Seen this way, the Realization Theorem amounts to a way of extracting the explicit content of a knowledge statement in which justifications are present only implicitly. One can think of the whole LP/S4 package as a kind of labor-saving device. We can reason more easily about knowledge if we don't make things explicit, but when we need explicit justifications, they can be calculated.

This, however, is not the whole story. A theorem of S4 can have many realizations—explicit justifications for things are not unique. We may need to supply justifications, not according to a general paradigm, but based on justifications we already have. For this, a direct application of the Realization Theorem is not sufficient. This paper can be seen as a first attempt at providing tools for precisely this sort of reasoning about evidence.

Once LP is seen as a logic of knowledge, extending it to a multi-agent version is an obvious step. Work has been done on this by a number of people, and is still ongoing. It is likely that the theorems of this paper extend to multiple agents. Also, we have looked at what might be called 'pure' versions of LP and of S4. All knowledge is strictly logical knowledge. But there are facts of the world too. We might know it rained in some particular place at some particular time. We don't know this by a process of logical reasoning, but we may deduce facts from it using logic. To use terminology that is standard elsewhere in the modal logic literature, we might want to consider LP with *local* as well as with *global* assumptions. It must be left to future research to determine how LP will function with local assumptions. In particular, what happens to the Realization Theorem,

and what happens to the results of this paper. This would bring us closer to everyday reasoning, as contrasted with the reasoning of pure mathematics.

References

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

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

[3] M. C. Fitting. A simple propositional S5 tableau system. *Annals of Pure and Applied Logic*, 96:107–115, 1999. Originally in *The Parikh Project, Seven papers in honour of Rohit*, Uppsala Prints and Reprints in Philosophy, 1996 Number 18.

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

[5] 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/`.

[6] M. C. Fitting. Reasoning with justifications. submitted to *Studia Logica*, 2007.

[7] K. Gödel. An interpretation of the intuitionistic propositional calculus. In S. Feferman, editor, *Kurt Gödel, Collected Works, Volume One*, pages 300–303. Oxford, 1986. Originally published as 'Eine Interpretation des intuitionistischen Aussagenkalküls', in *Ergebnisse eines mathematischen Kolloquiums*, vol 4, pp 39–40 (1933).

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

[9] 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.