2007

TR-2007006: Realizing Substitution Instances of Modal Theorems

Melvin Fitting

Realizing Substitution Instances of Modal Theorems

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 4, 2007

Abstract

Suppose X is a theorem of S4, and a realization for X has been constructed. If X' is a substitution instance of X, it is also a theorem of S4, and so is realizable, but the only available algorithm for producing a realization of X', so far, has been to apply a general realization algorithm to a cut-free proof of X'. In effect we start over and the realization of X plays no role. It is the purpose of this report to present an algorithm for realizing substitution instances of a realizable formula that is, we believe, more efficient than a simple appeal to a general realization algorithm itself.

1 Introduction

This report is about *justification logics*, the specific logic LP, and *realizations*. These topics are not defined here, nor is there any explanation of why we might be interested in them. For justification logics in general, see [2]. For LP, the primary reference is [1]. And for realizations, the machinery used here is taken from [3]. Knowledge of this material is essential for what follows.

The *Realization Theorem* of Artemov, for LP, says that if X is a theorem of S4, there is some way of replacing the modal operator $\Box$ with explicit proof polynomials, justifications, to produce a theorem of LP. Further, this can be done so that negative occurrences of $\Box$ are replaced with variables, with different occurrences being replaced by distinct variables. There are, by now, several proofs of this, but all known algorithmic ones make use of a cut-free sequent proof of X in S4. It is the subject of current research to produce a proof of the Realization Theorem building on axiomatic, rather than sequent, proofs in S4. The present report can be seen as a small contribution to this research.

Realizations, as defined in [1], are not required to meet the condition that negative occurrences of $\Box$ are replaced with variables, with distinct negative occurrences receiving

distinct variables. Realizations that meet this condition (and another that is not significant here) are called *normal*. All realizations considered in this paper will be normal. Consequently, in the interests of simple terminology the word "normal" will not be used. Its implied presence should be understood, however, since the condition is fundamental.

Suppose X is a theorem of S4, and a realization for X has been constructed. A *substitution instance* of X is the result of replacing propositional letters of X by other modal formulas. If X' is a substitution instance of X, it is also a theorem of S4, and so is realizable. But the only available technique for producing a realization of X', so far, has been to apply the basic realization algorithm to a cut-free proof of X'. In effect we start over and the realization of X plays no role. It is the purpose of this report to present an algorithm for realizing substitution instances of a realizable formula. We believe this algorithm is more efficient than a simple appeal to a general realization algorithm itself.

To illustrate the difficulties involved, here is a very simple example, Peirce's law, $((P \supset Q) \supset P) \supset P$, which, of course, is a tautology. Since it has no modal operators, a realization in LP is trivial—it is realized by itself. But now, suppose we substitute $\Box R$ for P, getting $((\Box R \supset Q) \supset \Box R) \supset \Box R$. The two left-most occurrences of $\Box$ are in negative positions, and so must be realized by distinct variables. Here is a realization for this formula—we leave it to the reader to check that it is, indeed, a theorem of LP. The realization is $((x{:}R \supset Q) \supset y{:}R) \supset (x + y){:}R$. Things are even more complicated if we substitute $\Box(\Box R \supset \Box S)$ for P, and obviously much more complex examples are easy to come by.

The present report does not avoid cut-free proofs entirely. But, to continue the Peirce law example, if we have a single cut-free proof of $((P \supset Q) \supset P) \supset P$, this is enough. We can then realize all instances of this formula, such as $((\Box R \supset Q) \supset \Box R) \supset \Box R$, by making use of an algorithm from [3] that depends directly on formula complexity, and not on proof complexity.

One goal of this paper is to show that the results of [3] have useful applications. The proofs (but not the algorithms) in that paper are somewhat formidable, so to balance that there is an emphasis here on exposition. As part of this, the work is first presented for cases where substitution is in non-modal tautologies. The work is clearest and cleanest in this setting. Only afterword are the modifications to get a general result given.

2 Annotated Formulas

Realizations of modal formulas pay attention to *occurrences* of modal operators, with positive and negative occurrences treated differently. In [3] some simple machinery was introduced for this purpose, and it will be useful here as well. First, let $L_\Box$ be the usual language of propositional modal logic, built up from propositional letters using, say, $\bot$, $\supset$, and $\Box$, with other connectives and $\Diamond$ taken as defined, in the usual way.

Next, an *annotated* version of $L_\Box$ is introduced, in which occurrences of $\Box$ become syn-

tactically distinct, and negative and positive occurrences are distinguished. The language $L^a_\Box$ is like $L_\Box$ except that instead of the single operator $\Box$ there is an infinite family, $\Box_1$, $\Box_2$, ... These will be called *indexed* modal operators and formulas of $L^a_\Box$ will be referred to as *annotated formulas*. 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. A *properly* annotated formula is an annotated formula in which no indexed modal operator occurs twice, if $\Box_n$ occurs in a negative position n is even, and if $\Box_n$ occurs in a positive position n is odd.

Semantically and proof theoretically, annotations are ignored. For instance, in an $\mathsf{S4}$ model $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \Vdash \rangle$ we assume:

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

Then in a model, an annotated formula X and its unannotated version X' behave alike at each world. Similarly, in giving an axiomatic proof using annotated formulas, annotations are simply ignored.

3 Realizations

By making use of annotated formulas, realizations can be treated as functions. Specifically, a *realization function* is a mapping from positive integers to proof polynomials that maps even integers to LP variables. 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.

If X is an annotated formula and r is a realization function, $r(X)$ is the result of replacing each modal operator $\Box_i$ in X with the proof polynomial $r(i)$. The result, $r(X)$, is a formula of LP.

If X is a modal formula (of $L_\Box$) a *realization* of X is a 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 says the following, using this terminology. If Z is a theorem of $\mathsf{S4}$, there is a realization of Z that is an injectively provable theorem of LP. In fact, if Z is a theorem of $\mathsf{S4}$, then for any properly annotated version X of Z there is a realization function r such that $r(X)$ is injectively provable in LP. (Injectively provable means provable using an injective constant specification.)

To continue an earlier example, consider $((\Box R \supset Q) \supset \Box R) \supset \Box R$ again. A properly annotated version of this is $((\Box_2 R \supset Q) \supset \Box_4 R) \supset \Box_1 R$. Let r be a realization function such that $r(2) = x_1$, $r(4) = x_2$ (these are required of all realization functions), and $r(1) = x_1 + x_2$. Then $r(((\Box_2 R \supset Q) \supset \Box_4 R) \supset \Box_1 R) = ((x_1{:}R \supset Q) \supset x_2{:}R) \supset (x_1 + x_2){:}R$ and this is provable in LP.

4 The Realization Merging Theorem

In [3] several results concerning realization functions were proved. Only one of these will be needed in this paper, though it will be needed in a somewhat extended form. The proof is not supplied here, but it should be noted that it is algorithmic—the algorithm makes use of the structure of subformulas, and not that of cut-free proofs. Besides realizations, *substitutions* come into play. Here a substitution is typically denoted σ and it maps LP proof variables to proof terms. Applying the substitution σ throughout an LP formula X produces another formula which is denoted here by $X\sigma$.

Theorem 4.1 (Realization Merging) *Let X be a properly annotated formula, and r_1, r_2, ..., r_n be realization functions. Then there is a realization/substitution pair $\langle r, \sigma\rangle$ such that (hereditary merging)*

1. *if φ is any positive subformula of X then $r_i(\varphi)\sigma \supset r(\varphi)$ is an injective theorem of LP, for $i = 1, 2, \ldots, n$;*

2. *if φ is any negative subformula of X then $r(\varphi) \supset r_i(\varphi)\sigma$ is an injective theorem of LP, for $i = 1, 2, \ldots, n$.*

Further, σ will be the identity substitution on all variables except for those x_k where $\Box_{2k}$ occurs in X (σ lives on the input positions in X), and for each variable x, the term $x\sigma$ contains no variables except x (σ meets the no new variable condition).

This is almost, but not quite, Theorem 7.2 of [3]. The actual result in the earlier paper is for two realization functions, r_1 and r_2, but the same proof works for any quantity bigger than 2 as well. It is the extended version, stated above, that will be needed here.

5 Classical Implication Bases

This is the one place in our treatment of classical tautologies that a cut-free proof is needed.

Definition 5.1 For each propositional letter, say P, let P_a, P_b, P_c, ..., be a list of new, distinct propositional letters; we say these are *associates* of P. Let X be a (classical or modal) formula, and let X' be the result of replacing each propositional letter occurrence in X by an associate, so that different occurrences are replaced by distinct associates. We say X' is a *discriminant* of X.

For example, $((P_a \supset Q) \supset P_b) \supset P_c$ is a discriminant of $((P \supset Q) \supset P) \supset P$, Peirce's Law.

We write $S \models_{\mathsf{PC}} X$ to symbolize that formula X is a classical propositional consequence of formula set S. This will only be used when S is finite, and can be taken to mean $\bigwedge S \supset X$ is a tautology.

Definition 5.2 Let X be a classical formula, and let X' be a discriminant of it. We say a set S is a *classical atomic implication basis* for X' if the members of S are all of the form $P_a \supset P_b$, where P_a and P_b are associates of the same propositional letter P, with P_a occurring negatively and P_b occurring positively in X, and $S \models_{\mathsf{PC}} X$.

For example, it is easy to check that

$$\{(P_a \supset P_c), (P_b \supset P_c)\} \models_{\mathsf{PC}} ((P_a \supset Q) \supset P_b) \supset P_c$$

and $\{(P_a \supset P_c), (P_b \supset P_c)\}$ is an atomic implication basis for $((P_a \supset Q) \supset P_b) \supset P_c$.

Theorem 5.3 (Classical Atomic Implication Basis) *Let X be a classical propositional formula, and let X' be any discriminant of X. Then X is a tautology if and only if there is some set S that is a classical atomic implication basis for X'.*

One direction of the proof is rather simple. It is easy to see that consequence is closed under substitution. That is, if $S \models_{\mathsf{PC}} Z$, and if Z' is a substitution instance of Z and S' is the set of substitution instances of S, all using the same substitution, then $S' \models_{\mathsf{PC}} Z'$. Now suppose X' is a discriminant of X, and S is an atomic implication basis for X', so $S \models_{\mathsf{PC}} X'$. Carry out the substitution that replaces each of the associates of P by P itself, for each propositional letter P. This turns X' back into X, and it turns each member $P_a \supset P_b$ of S into the tautology $P \supset P$. So X is a consequence of tautologies, and hence is itself a tautology.

The other direction makes use of cut-free proofs. This can be given either using the machinery of Gentzen sequents, or of tableaus. Here tableaus will be used. It is assumed that the reader is generally familiar with the tableau idea, and what follows is just a sketch.

A (classical propositional) tableau is a special kind of tree with nodes labeled with *signed* formulas, $T\,X$ or $F\,X$, where X is a formula. A tableau proof of X begins by placing $F\,X$ at the root, then 'growing' the tree using branch extension rules, which are as follows, assuming $\supset$ is the only connective.

$$\frac{F\,X \supset Y}{\begin{array}{c} T\,X \\ F\,Y \end{array}} \qquad \frac{T\,X \supset Y}{F\,X \mid T\,Y}$$

In the first of these, if the signed formula above the line is on a tableau branch, the formulas below the line can be added to the end of the branch. In the second rule, if the signed formula above the line is on a tableau branch, the end of the branch can be split and extended with each of the formulas displayed below the line on the new left and right branches respectively.

A tableau branch is *closed* if it contains $T\,Z$ and $F\,Z$, or if it contains $T\,\bot$. A tableau is closed if every branch is closed. A closed tableau for $F\,X$ constitutes a tableau proof of

X. If X is a tautology, not only will it have a tableau proof, but one in which each branch closes at the *atomic* level.

Now, suppose X is a tautology and X' is a discriminant of X. There is a closed tableau $\mathcal{T}$ for $F\,X$, in which each branch closes atomically. Use $\mathcal{T}$ to create a new tableau $\mathcal{T}'$, having the same shape as $\mathcal{T}$. Begin by placing $F\,X'$ at the root instead of $F\,X$. If $T\,A$ and $F\,B$ occur on a branch of $\mathcal{T}$, where $F\,A \supset B$ occurs earlier on the branch, and if the construction of $\mathcal{T}'$ has been carried through to the point where a discriminant, $F\,A' \supset B'$ of $F\,A \supset B$, has been added to $\mathcal{T}'$ at the position corresponding to that of $F\,A \supset B$ in $\mathcal{T}$, then add $T\,A'$ and $F\,B'$ to $\mathcal{T}'$ at the positions corresponding to $T\,A$ and $F\,B$ in $\mathcal{T}$. And similarly for the other tableau branch extension rule. Essentially, if an occurrence of P in X has been replaced with the associate P_a, one follows that occurrence of P down through the tableau $\mathcal{T}$, replacing each instance with P_a, and so on. Thus one creates a tableau $\mathcal{T}'$ in which each signed formula is a discriminant of the signed formula at the corresponding position in $\mathcal{T}$.

In this way we get a tableau $\mathcal{T}'$ for $F\,X'$. Of course $\mathcal{T}'$ will not generally be closed. If a branch of $\mathcal{T}$ closed because it contained $T\,P$ and $F\,P$, the corresponding branch in $\mathcal{T}'$ will contain $T\,P_a$ and $F\,P_b$, where P_a and P_b are different associates of P, with P_a having a negative occurrence in X' and P_b a positive occurrence. Let us say that branch of $\mathcal{T}'$ *requires the formula* $P_a \supset P_b$ to close. Let S be the set of all formulas required by the various branches of $\mathcal{T}'$. It is not hard to see that $S \models_{\mathsf{PC}} X'$.

For example, here is a closed tableau for Peirce's Law, $((P \supset Q) \supset P) \supset P$.

$$
\begin{array}{l}
F\,((P \supset Q) \supset P) \supset P \\
T\,(P \supset Q) \supset P \\
F\,P \\
\qquad\diagup\quad\diagdown \\
F\,P \supset Q \qquad T\,P \\
T\,P \\
F\,Q
\end{array}
$$

And here is a closed tableau for the associate $((P_a \supset Q) \supset P_b) \supset P_c$.

$$
\begin{array}{l}
F\,((P_a \supset Q) \supset P_b) \supset P_c \\
T\,(P_a \supset Q) \supset P_b \\
F\,P_c \\
\qquad\diagup\quad\diagdown \\
F\,P_a \supset Q \qquad T\,P_b \\
T\,P_a \\
F\,Q
\end{array}
$$

From it we read off the atomic implication basis $\{(P_a \supset P_c), (P_b \supset P_c)\}$.

6 The Substitution Realization Algorithm—Simple Version

In this section we confine things to realizing substitution instances of pure tautologies. In Section 8 we look at substituting in theorems of S4 generally. Suppose $\varphi(P, Q, \ldots)$ is a classical tautology with no modal operators, where P, Q, ... are all the propositional letters in the formula. Of course $\varphi(P, Q, \ldots)$ is its own realization in LP. Let Z, W, ... be modal formulas (of $L_\Box$), and let $\varphi(Z, W, \ldots)$ be the modal formula that results on substituting Z for occurrences of P, W for occurrences of Q, and so on. Then $\varphi(Z, W, \ldots)$ will be an S4 validity. We show how to realize this formula in LP.

To begin, let $\varphi'(P_a, P_b, \ldots, Q_a, Q_b, \ldots)$ be a discriminant of $\varphi(P, Q, \ldots)$ where $P_a, P_b, \ldots$ are associates of P, and $Q_a, Q_b, \ldots$ are associates of Q, and so on. By Theorem 5.3 there is an atomic implication basis for $\varphi'(P_a, P_b, \ldots, Q_a, Q_b, \ldots)$, say it is $\{(P_i \supset P_j), \ldots, (Q_m \supset Q_n), \ldots\}$.

We need a properly annotated version of $\varphi(Z, W, \ldots)$ and, in this, each different occurrence of Z must be annotated using distinct indexes, similarly for W, and so on. So, a properly annotated version of $\varphi(Z, W, \ldots)$ can be represented as $\varphi'(Z_a, Z_b, \ldots, W_a, W_b, \ldots)$ where each Z_i is an annotated version of Z, each W_i is an annotated version of W, and so on, and there is no annotation overlap. Further, if P_i occurs positively, Z_i will be properly annotated, and if P_i occurs negatively, $\neg Z_i$ will be properly annotated, and so on.

We got $\varphi'(Z_a, Z_b, \ldots, W_a, W_b, \ldots)$ by substituting in $\varphi'(P_a, P_b, \ldots, Q_a, Q_b, \ldots)$. In a similar way we can substitute in the atomic implication basis $\{(P_i \supset P_j), \ldots, (Q_m \supset Q_n), \ldots\}$ to get $\{(Z_i \supset Z_j), \ldots, (W_m \supset W_n), \ldots\}$. Though we will not need it, since $\{(P_i \supset P_j), \ldots, (Q_m \supset Q_n), \ldots\} \models_{\mathsf{PC}} \varphi'(P_a, P_b, \ldots, Q_a, Q_b, \ldots)$, it is easy to see that $\{(Z_i \supset Z_j), \ldots, (W_m \supset W_n), \ldots\} \models_{\mathsf{PC}} \varphi'(Z_a, Z_b, \ldots, W_a, W_b, \ldots)$ also holds.

For example, an instance of Peirce's Law is $\varphi(P, Q) = ((P \supset Q) \supset P) \supset P$, where P and Q are propositional letters. This has no modal operators. Suppose X is some complicated modal formula and we substitute X for P, getting the following instance of Peirce's Law: $\varphi(X, Q) = ((X \supset Q) \supset X) \supset X$. $\varphi'(P_a, P_b, P_c, Q) = ((P_a \supset Q) \supset P_b) \supset P_c$ is a discriminant of $\varphi(P, Q)$. A properly annotated version of $\varphi(X, Q)$ will be $\varphi'(X_a, X_b, X_c, Q) = ((X_a \supset Q) \supset X_b) \supset X_c$ where X_a, X_b, and X_c are annotated versions of X that share no indexes, with even indexes in negative positions in X_c, but in positive positions in X_a and X_b, while the opposite holds for odd indexes.

Definition 6.1 Annotated formulas U and V are *mirror twins*, with U the negative twin and V the positive twin, if

1. Both U and V are annotated versions of the same modal formula.
2. U and V do not share an index and no index occurs more than once in either annotated formula.
3. if $\Box_n$ occurs positively in U then n is even; if $\Box_n$ occurs negatively in V then n is odd.

4. if $\Box_n$ occurs negatively in V then n is even; if $\Box_n$ occurs positively in U then n is odd.

The definition above can be stated in the following equivalent form. U and V are mirror twins with U the negative twin and V the positive one, if U and V are annotated versions of the same modal formula and $U \supset V$ is *properly* annotated.

It is not hard to see that, in the set $\{(Z_i \supset Z_j), \ldots, (W_m \supset W_n), \ldots\}$, Z_i and Z_j are mirror twins with Z_i negative and Z_j positive, and similarly for W_m and W_n, and so on.

If U and V are mirror twins, and $\Box_i$ occurs in one and $\Box_j$ occurs in the other, there is an obvious notion of the two occurring in *corresponding positions*. If indexes are dropped, thus converting annotated formulas to standard modal formulas, $\Box_i$ and $\Box_j$ convert to the same occurrence of $\Box$ if they are in corresponding positions in the annotated formulas. Note that if $\Box_i$ and $\Box_j$ are in corresponding positions in U and V then exactly one of i and j will be even, exactly one will be odd.

Returning to the example of Peirce's Law again, X_c and X_a are mirror twins, and so are X_c and X_b, with X_c being the positive twin in each case.

Definition 6.2 Let U and V be mirror twins with U the negative twin and V the positive twin. Let r be a realization function. The behavior of r on even indexes is then completely determined; $r(2n) = x_n$. We say r is a *matching function* for U and V provided the following condition is met: if $\Box_{2n}$ occurs in one formula and $\Box_{2k+1}$ occurs in the other formula at the corresponding position, then $r(2k+1) = r(2n) = x_n$.

For example, the two formulas $\Box_4 P \supset \Box_3 P$ and $\Box_1 P \supset \Box_2 P$ are mirror twins with the first formula being the positive twin and the second formula the negative one. Indexed modal operators $\Box_1$ and $\Box_4$ occur in corresponding positions, as do $\Box_2$ and $\Box_3$. Any realization function r must map 2 to x_1 and 4 to x_2. Then r is a matching function for these formulas if $r(1) = r(4) = x_2$ and $r(3) = r(2) = x_1$. Note that $r(\Box_1 P \supset \Box_2 P)$ and $r(\Box_4 P \supset \Box_3 P)$ are the same.

Given any mirror twins, there is always a matching function for them, and any two matching functions for them will agree on the indexes that occur in them.

We return to the statement of the algorithm. Recall, we began with the classical tautology $\varphi(P, Q, \ldots)$, a modal substitution instance $\varphi(Z, W, \ldots)$ of this, a discriminant, $\varphi'(P_a, P_b, \ldots, Q_a, Q_b, \ldots)$, and an annotated version of the modal substitution instance, $\varphi'(Z_a, Z_b, \ldots, W_a, W_b, \ldots)$. We also had an atomic implication basis $\{(P_i \supset P_j), \ldots, (Q_m \supset Q_n), \ldots\}$ for $\varphi'(P_a, P_b, \ldots, Q_a, Q_b, \ldots)$, and we know that $\{(Z_i \supset Z_j), \ldots, (W_m \supset W_n), \ldots\} \models_{\mathsf{PC}} \varphi'(Z_a, Z_b, \ldots, W_a, W_b, \ldots)$. Finally, Z_i and Z_j are mirror twins, $\ldots$, W_m and W_n are mirror twins, $\ldots$.

Let $r_{Z,i,j}$ be a matching function for Z_i and Z_j, $\ldots$ let $r_{W,m,n}$ be a matching function for W_m and W_n, $\ldots$. Let $\langle r, \sigma \rangle$ be the realization function that results from merging $r_{Z,i,j}$, $\ldots$, $r_{W,m,n}$, $\ldots$ on the formula $(Z_i \supset Z_j) \wedge \ldots \wedge (W_m \supset W_n) \wedge \ldots$, using Theorem 4.1.

Claim r realizes $\varphi'(Z_a, Z_b, \ldots, W_a, W_b, \ldots)$.

Verifying this claim is now rather easy, given all that has gone before. Since

$$\{(P_i \supset P_j), \ldots, (Q_m \supset Q_n), \ldots\} \models_{\mathsf{PC}} \varphi'(P_a, P_b, \ldots, Q_a, Q_b, \ldots) \tag{1}$$

then the following consequence is correct in LP,

$$\{(P_i \supset P_j), \ldots, (Q_m \supset Q_n), \ldots\} \models_{\mathsf{PC}} \varphi'(P_a, P_b, \ldots, Q_a, Q_b, \ldots) \tag{2}$$

and since theorems of LP are closed under substitution, it follows that

$$\{(r(Z_i) \supset r(Z_j)), \ldots, (r(W_m) \supset r(W_n)), \ldots\} \models \varphi'(r(Z_a), r(Z_b), \ldots, r(W_a), r(W_b), \ldots) \tag{3}$$

is a theorem of LP. Since

$$r(\varphi(Z_a, Z_b, \ldots, W_a, W_b, \ldots)) = \varphi(r(Z_a), r(Z_b), \ldots, r(W_a), r(W_b), \ldots)$$

we are done if we show each formula in the set of premises is an LP theorem. Consider one of them, $r(Z_i) \supset r(Z_j)$. Since $\langle r, \sigma \rangle$ merges $r_{Z,i,j}$ with other realization functions on a formula that includes $Z_i \supset Z_j$ as a conjunct, and Z_i occurs negatively in this formula and Z_j positively, then by Theorem 4.1 both $r_{Z,i,j}(Z_j)\sigma \supset r(Z_j)$ and $r(Z_i) \supset r_{Z,i,j}(Z_i)\sigma$ are LP theorems. But, $r_{Z,i,j}$ is a matching function for Z_i and Z_j, so $r_{Z,i,j}(Z_i) = r_{Z,i,j}(Z_j)$. It follows that $r(Z_i) \supset r(Z_j)$ is an LP theorem.

7 S4 Implication Bases

We now turn to extending the work of the previous section, to allow for realizing substitution instances of *any* theorem of $\mathsf{S4}$, and not just instances of tautologies. The algorithm is more complicated. In this section the notion of Classical Implication Basis, from Section 5, is generalized. Definition 5.1, of associate and discriminant, applies in the modal setting as well as in the classical one. Theorem 5.3 extends to the following.

We write $S \models_{\mathsf{S4}} X$ to symbolize that formula X is a *local* consequence of formula set S in the logic $\mathsf{S4}$. As before, this will only be used when S is finite, and can be taken to mean $\bigwedge S \supset X$ is a theorem of $\mathsf{S4}$, as in the classical case.

Definition 7.1 Let X be a modal formula, and let X' be a discriminant of it. We say a set S is an *S4 atomic implication basis* for X' if the members of S are all of the form $P_a \supset P_b$ or $\Box(P_a \supset P_b)$, where P_a and P_b are associates of the same propositional letter P, with P_a occurring negatively and P_b occurring positively in X, and $S \models_{\mathsf{S4}} X$.

Theorem 7.2 (S4 Atomic Implication Basis) *Let X be a modal formula, and let X' be any discriminant of X. Then X is a theorem of S4 if and only if there is some set S that is an S4 atomic implication basis for X'.*

The proof that the existence of an atomic implication basis for the discriminant X' of X implies theoremhood of X is essentially the same as in the classical setting of Section 5. The other, more interesting, direction is also similar except that a tableau system for S4 must be used. The rules include the classical ones given earlier, though formulas are now in a modal language. And there are two additional specifically modal rules. The first is simply this.

$$\frac{T\,\Box X}{T\,X}$$

The second is a more complicated branch modification rule, which replaces one branch by another. To state this, if S is a set of signed formulas, let $S^\sharp$ be $\{T\,\Box X \mid T\,\Box X \in S\}$. Then the rule is as follows.

$$\frac{S, F\,\Box X}{S^\sharp, F\,X}$$

In this, if the set of signed formulas above the line matches what is on a branch, that branch can be replaced by one labeled with the signed formulas below the line.

Now, if there is a closed tableau $\mathcal{T}$ for $F\,X$, we construct a tableau (unclosed) $\mathcal{T}'$ for $F\,X'$ as described in Section 5 except that now modal rules are also allowed. The set S is created from $\mathcal{T}'$. If a branch of $\mathcal{T}'$ requires the formula $P_a \supset P_b$ to close, and the branch modification rule above *has not* been applied on that branch, put $P_a \supset P_b$ in S, and if the branch modification rule *has* been applied on that branch, put $\Box(P_a \supset P_b)$ in S.

For example, $[\Box(P \supset Q) \wedge R] \supset [(\Box P \supset \Box Q) \wedge (R \vee S)]$ is a theorem of S4. One discriminant for it is $[\Box(P_a \supset Q_a) \wedge R_a] \supset [(\Box P_b \supset \Box Q_b) \wedge (R_b \vee S)]$. Following the algorithm just sketched produces an S4 atomic implication basis for this formula: $\{R_a \supset R_b, \Box(P_b \supset P_a), \Box(Q_a \supset Q_b)\}$.

8 The Substitution Realization Algorithm—Full Version

The algorithm given in Section 6 is now extended to the general case. This time, suppose $\varphi(P, Q, \ldots)$ is a theorem of S4, not just a tautology, and let Z, W, ... be modal formulas. We show how to realize $\varphi(Z, W, \ldots)$.

Let $\varphi'(P_a, P_b, \ldots, Q_a, Q_b, \ldots)$ be a discriminant for $\varphi(P, Q, \ldots)$. Then, using Theorem 7.2 and the algorithm in its proof, let $S = \{\Box(P_i \supset P_j), (P_k \supset P_l), \ldots, \Box(Q_m \supset Q_n), (Q_o \supset Q_p) \ldots\}$ be an S4 atomic implication basis for $\varphi'(P_a, P_b, \ldots, Q_a, Q_b, \ldots)$. Then $S \models_{\mathsf{S4}} \varphi'(P_a, P_b, \ldots, Q_a, Q_b, \ldots)$. Since

$$[\Box(P_i \supset P_j) \wedge (P_k \supset P_l) \wedge \ldots \wedge \Box(Q_m \supset Q_n) \wedge (Q_o \supset Q_p) \ldots] \supset \varphi'(P_a, P_b, \ldots, Q_a, Q_b, \ldots) \tag{4}$$

is a theorem of S4, it is realizable. This is the one place in this algorithm where we make use of a standard realization algorithm, and we use it on (4) and not on $\varphi(P, Q, \ldots)$ itself. To say this using the machinery of the present paper, we need a properly annotated version of (4). The modal operators displayed on the left of the implication are the only ones on the left of the implication, and they occur negatively. φ' itself can contain modal operators. Let $\varphi''(P_a, P_b, \ldots, Q_a, Q_b, \ldots)$ be a properly annotated version of $\varphi'(P_a, P_b, \ldots, Q_a, Q_b, \ldots)$ and let $\Box_{2f}$, $\Box_{2g}$, ... be evenly indexed modal operators with distinct indexes which do not appear in φ''. Then the following is a properly annotated version of (4).

$$[\Box_{2f}(P_i \supset P_j) \wedge (P_k \supset P_l) \wedge \ldots \wedge \Box_{2g}(Q_m \supset Q_n) \wedge (Q_o \supset Q_p) \ldots] \supset \varphi''(P_a, P_b, \ldots, Q_a, Q_b, \ldots) \tag{5}$$

Now, let s be a realization function that maps formula (5) to a theorem of LP, thus realizing formula (4). The only indexed modal operators on the left of the implication in (5) are the ones displayed and on these, $s(2f) = x_f$, $s(2g) = x_g$, and so on. So, the following formula is provable in LP.

$$[x_f{:}(P_i \supset P_j) \wedge (P_k \supset P_l) \wedge \ldots \wedge x_g{:}(Q_m \supset Q_n) \wedge (Q_o \supset Q_p) \ldots] \supset s(\varphi''(P_a, P_b, \ldots, Q_a, Q_b, \ldots)) \tag{6}$$

Since P_a, P_b, ... are all propositional variables, the realization function s on the right side of the implication in (6) will not affect them. To make this clear, we rewrite the formula suggestively as follows.

$$[x_f{:}(P_i \supset P_j) \wedge (P_k \supset P_l) \wedge \ldots \wedge x_g{:}(Q_m \supset Q_n) \wedge (Q_o \supset Q_p) \ldots] \supset s(\varphi'')(P_a, P_b, \ldots, Q_a, Q_b, \ldots) \tag{7}$$

Our task is to realize $\varphi(Z, W, \ldots)$, where Z, W, ... are modal formulas. We proceed roughly the way we did in Section 6. We need a properly annotated version of $\varphi(Z, W, \ldots)$ and this will have the form $\varphi''(Z_a, Z_b, \ldots, W_a, W_b, \ldots)$ where each Z_i is an annotated version of Z, each W_i is an annotated version of W, and so on, and there is no annotation overlap between them or between them and φ''. We can also arrange things so that none of $\Box_{2f}$, $\Box_{2g}$, ... occurs in any of $Z_a, Z_b, \ldots, W_a, W_b, \ldots$. Also, if P_i occurs positively, Z_i will be properly annotated, and if P_i occurs negatively, $\neg Z_i$ will be properly annotated, and so on.

$\varphi''(Z_a, Z_b, \ldots, W_a, W_b, \ldots)$ is a properly annotated version of $\varphi(Z, W, \ldots)$, substituting Z_a for P_a, and so on, in $\varphi''(P_a, P_b, \ldots, Q_a, Q_b, \ldots)$. Then, as in Section 6, in the formula (7) $x_f{:}(P_i \supset P_j)$ occurs on the left of the implication and so Z_i and Z_j will be mirror twins with Z_i the negative one. Similarly $(P_k \supset P_l)$ occurs and so Z_k and Z_l will be mirror twins with Z_k the negative one. And so on.

Let $r_{Z,i,j}$ be a matching function for Z_i and Z_j, ... let $r_{W,m,n}$ be a matching function for W_m and W_n, ... Let $\langle r, \sigma \rangle$ be the realization function that results from merging $r_{Z,i,j}$, ..., $r_{W,m,n}$, ... on the formula $(Z_i \supset Z_j) \wedge \ldots \wedge (W_m \supset W_n) \wedge \ldots$, using Theorem 4.1.

Since (7) is a theorem of LP, so is the following substitution instance.

$$[x_f{:}(r(Z_i) \supset r(Z_j)) \wedge (r(Z_k) \supset r(Z_l)) \wedge \ldots \wedge x_g{:}(r(W_m) \supset r(W_n)) \wedge (r(W_o) \supset r(W_p)) \ldots] \supset s(\varphi'')(r(Z_a), r(Z_b), \ldots, r(W_a), r(W_b), \ldots) \tag{8}$$

Using exactly the same argument as in Section 6, each of the following is a theorem of LP: $(r(Z_i) \supset r(Z_j))$, $(r(Z_k) \supset r(Z_l))$, $\ldots$, $(r(W_m) \supset r(W_n))$, $(r(W_o) \supset r(W_p))$, $\ldots$. Then of course the following is also an LP theorem.

$$[x_f{:}(r(Z_i) \supset r(Z_j)) \wedge \ldots \wedge x_g{:}(r(W_m) \supset r(W_n)) \ldots] \supset s(\varphi'')(r(Z_a), r(Z_b), \ldots, r(W_a), r(W_b), \ldots) \tag{9}$$

We now need to use the Internalization Lemma. Since $(r(Z_i) \supset r(Z_j))$ is an LP theorem there is a closed proof polynomial t_f such that $t_f{:}(r(Z_i) \supset r(Z_j))$ is provable. Likewise since $(r(W_m) \supset r(W_n))$ is an LP theorem there is a closed proof polynomial t_g such that $t_g{:}(r(W_m) \supset r(W_n))$ is provable. And so on. Let τ be the substitution that replaces x_f with t_f, x_g with t_g, and so on. Then from (9) we also have provability of the following.

$$[x_f{:}(r(Z_i) \supset r(Z_j)) \wedge \ldots \wedge x_g{:}(r(W_m) \supset r(W_n)) \ldots]\tau \supset s(\varphi'')\tau(r(Z_a), r(Z_b), \ldots, r(W_a), r(W_b), \ldots)\tau \tag{10}$$

And by construction, the antecedent of (10) is provable, hence so is the following.

$$s(\varphi'')\tau(r(Z_a), r(Z_b), \ldots, r(W_a), r(W_b), \ldots)\tau \tag{11}$$

Finally, let r^* be the function given as follows. On indexes of φ'', $r^*(q) = s(q)\tau$. On an index q in one of Z_a, Z_b, $\ldots$, W_a, W_b, $\ldots$, $r^*(q) = r(q)\tau$. It is not hard to verify that this is a properly defined realization function. Then

$$r^*(\varphi''(Z_a, Z_b, \ldots, W_a, W_b, \ldots)) = s(\varphi'')\tau(r(Z_a), r(Z_b), \ldots, r(W_a), r(W_b), \ldots)\tau \tag{12}$$

and hence we have LP provability of

$$r^*(\varphi''(Z_a, Z_b, \ldots, W_a, W_b, \ldots)) \tag{13}$$

and so r^* is a realization function establishing realizability of $\varphi(Z, W, \ldots)$.

9 Conclusion

The work presented here is for LP, but it extends with little change to similar logics, LP without !, or LP with ? for negative introspection. The deeper issue is how to handle *modus ponens*. If we have a realization of X and a realization of $X \supset Y$ how do we get a

realization of Y? The methods given here do not work, because the role of positive and negative occurrence reverses between X and the X-part of $X \supset Y$. *Modus Ponens* remains a central open problem for efficient realization.

Here is a possible line of attack. Suppose $X \supset Y$ and X are theorems of $\mathsf{S4}$, and $X' \supset Y'$ is a provable realization of $X \supset Y$ in LP, and X'' is a provable realization of X, in LP. Since polarities are opposite in X and in the X-part of $X \supset Y$, variable occurrences in X' must match (possibly) non-variable occurrences in X'', and similarly the other way around. Let σ be the result of unifying X' with X''. Since the theorems of LP are closed under substitution, both $X'\sigma \supset Y'\sigma$ and $X''\sigma$ will be theorems of LP. But since σ was a unifier, by *modus ponens* in LP, $Y'\sigma$ is a theorem.

Unfortunately, here is an example of what can go wrong with what was just outlined. Suppose we have $\Box[(P \wedge Q) \supset P]$ and $\Box[(P \wedge Q) \supset P] \supset [\Box A \vee \neg\Box A]$, both of which are theorems of $\mathsf{S4}$, and from which $\Box A \vee \neg\Box A$ follows by *modus ponens*. Here are LP realizations of the two formulas.

$$[x{:}(P \wedge Q) \supset (c \cdot x){:}P]$$
$$[y{:}(P \wedge Q) \supset y{:}P] \supset [z{:}A \vee \neg z{:}A]$$

where c is a proof constant for $(P \wedge Q) \supset P$. An attempt to unify the first formula with the antecedent of the second violates the occurs check!

In the example above, of course $[z{:}A \vee \neg z{:}A]$ can be proved, just not *this* way. If it could be shown that any provable formula has a proof without the kind of circularity displayed above (as happens with the example), the unification method would work. But notice, this is a kind of 'global' property of axiomatic proofs. It is not clear if such an approach can succeed, but it does seem to hold some promise.

References

[1] S. Artemov. Explicit provability and constructive semantics. *The Bulletin for Symbolic Logic*, 7(1):1–36, 2001.
`http://www.cs.gc.cuny.edu/~sartemov/publications/BSL.ps`.

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

[3] M. C. Fitting. Realizations and LP. See also: Realizations and LP—Just the Algorithms, 2006.