Realization for Justification Logics via Nested Sequents: Modularity through Embedding

Remo Goetschi*, Roman Kuznets*

Institut für Informatik und angewandte Mathematik, Universität Bern
Neubrückstrasse 10, 3012 Bern, Switzerland

Abstract

Justification logics are refinements of modal logics, where justification terms replace modalities. Modal and justification logics are connected via so-called realization theorems. We develop a general constructive method of proving the realization of a modal logic in an appropriate justification logic by means of cut-free modal nested sequent systems. We prove a constructive realization theorem that uniformly connects every normal modal logic formed from the axioms d, t, b, 4, and 5 with one of its justification counterparts. We then generalize the notion of embedding introduced by Fitting for justification logics, which enables us to extend our realization theorem to all natural justification counterparts. As a result, we obtain a modular realization theorem that provides several justification counterparts based on various axiomatizations of a modal logic. We also prove that these justification counterparts realize the same modal logic if and only if they belong to the same equivalence class induced by our embedding relation, thereby demonstrating that the embedding provides the right level of granularity among justification logics.

Keywords: justification logic, modal logic, nested sequents, realization theorem, embedding

1. Introduction

Justification logic. The language of justification logic is a refinement of the language of modal logic. It replaces the single modality $\Box$ by a family of so-called *justification terms*. While a modal formula $\Box A$ can be read as *A is provable* or *A is known*, a justification counterpart $t : A$ of this formula is read as *t is a proof of A* or *A is known for reason t*. By introducing operations on terms, justification logic studies the operational content of modality in various modal logics. In this paper, we develop a method for testing whether a given set of operations on justifications is sufficient to represent a given modal logic defined via a nested sequent system. We also apply the method to study comparative strengths of several such sets of operations.

The first justification logic, called the *Logic of Proofs* or LP, was introduced by Artemov [1, 2] as a stepping stone for giving an arithmetical semantics for the modal logic S4. Epistemic logic is another promising area of application for justification logics. For example, as shown in [5], justification logics avoid the well-known logical omniscience problem because justification terms have a structure and thus provide a measure of how hard it is to obtain knowledge of something.

The formal correspondence between S4 and LP, called a *realization theorem*, has two directions. First, it says that each provable formula of S4 can be turned into a provable formula of LP by realizing, i.e., replacing, instances of modalities with justification terms. The converse direction says that replacing all terms in a provable formula of LP with modalities results in a modal formula provable in S4. Similar correspondences

*Corresponding author
Email addresses: goetschi@iam.unibe.ch (Remo Goetschi), kuznets@iam.unibe.ch (Roman Kuznets)

have been established for several other modal logics by means of various proof methods (see an overview in [3]).

Methods for proving realization. There are essentially two methods of establishing realization theorems: the syntactic method due to Artemov [1, 2] and the semantic method due to Fitting [13]. The syntactic method makes use of cut-free sequent systems for modal logics, while the semantic method makes use of a Kripke-style semantics for justification logics. In contrast to the semantic method, the syntactic method is constructive: it provides an algorithm for computing justification terms that realize all the occurrences of modalities in a given modal theorem.

The semantic method has been used to prove several realization theorems: for S4, S5, K45, and KD45 [3, 13, 23]. Prior to the publication of [10], constructive realizations, via the syntactic method, were available only for K, D, T, K4, D4, S4, and S5 [1, 2, 4, 7, 16, 17]. In the case of S5, for which no cut-free sequent system is available, two approaches have been used: via a translation from S5 to K45 [17] in conjunction with the *realization merging* technique developed in [16] and via a cut-free hypersequent system [4]. However, neither approach can be applied to other modal logics that lack cut-free sequent systems, such as K5 and KB.

General realization. In this paper, we develop a general method for proving realization theorems, which applies to a wide class of modal logics that can be captured by cut-free nested sequent systems consisting of so-called context-sharing rules. Nested sequents, which can be viewed as trees of sequents, naturally generalize both sequents, which are nested sequents of depth zero, and hypersequents, which are essentially nested sequents of depth one. A crucial feature of these proof systems is *deep inference* [8, 19], which in this case means applying inference rules to formulas arbitrarily deep inside a nested sequent. We show that in order to realize the modal logic of a nested sequent system, it is enough to realize the non-nested, or shallow, version of each rule. We apply our method to the nested sequent systems by Brünnler [9] that capture all the 15 normal modal logics formed by the axioms d, t, b, 4, and 5, which gives us a uniform constructive realization theorem for these logics. In particular, this proves Pacuit's conjecture implicit in [22] that J5[1] is a justification counterpart of K5. Our method also helps provide justification counterparts for the modal logics D5, KB, DB, TB, and KB5, which, to our knowledge, did not have justification counterparts prior to the publication of [10].

Embedding and modular realization. Based on our realization method, we discuss the question of modularity of realizations: each modal axiom has a natural corresponding justification axiom. However, a modal logic may have several axiomatizations and thus, a priori, may have several justification counterparts, supposedly one for each axiomatization. These counterparts mainly differ in the set of operations on justifications they employ. We classify these various justification counterparts by introducing an embedding relation on them that extends that of Fitting [15]. This embedding gives rise to an equivalence relation, which is natural in the sense that justification logics are equivalent iff they realize the same modal logic. The machinery of embeddings enables us to study minimal sets of operations on justifications that are sufficient to realize a given modal logic. For instance, we have shown that the operation of positive introspection is not necessary to realize the modal logic S5, although it enjoys positive introspection.

Outline. In Section 2, we introduce justification logics and modal logics. In Section 3, we introduce notation and prove auxiliary lemmas to be used in the following sections, as well as recall Fitting's merging technique. In Section 4, we introduce nested sequent systems and describe our general method for proving realization theorems. We use this method in Section 5 to prove our central result: the uniform realization theorem. In Section 6, we classify the justification logics using our notion of embedding and prove a modular realization theorem.

Relationship to previous work. In [10], which was a joint work with Kai Brünnler, we proved a uniform realization theorem for all the 15 normal modal logics formed by the axioms d, t, b, 4, and 5. The proof of the realization theorem presented there is a special case of the general method described in this paper. Here, we also prove a modular realization theorem that provides axiomatization-based justification counterparts for those modal logics among the above-mentioned 15 that have more than one axiomatization. Note that the definition of justification logics we use here slightly differs from the one used in [10]. To

[1]Pacuit used the name LP(K5).

minimize the number of operations on justifications, the negative introspection operation ? was used in [10] to realize both the modal axioms 5 and b. However, because of the new definition of embedding for justification logics, introduced in Section 6 of this paper, it makes more sense to use a new operation $\bar{?}$ to realize b and to establish the exact relationship between the operation ?, typically used to realize 5, and this new $\bar{?}$ by exploring the conditions under which one can be replaced by the other. Another difference from [10] is that justification constants are assigned levels to make the formulation of the results on embedding more elegant (see Remark 2.2 for details).

2. Justification Logics and Their Modal Counterparts

In this section, we define the modal and justification languages, give axiom systems we work with, both modal and justification, and introduce forgetful projection and realization theorems, which provide a formal connection between these languages and between these logics. We also explain in detail the naming conventions for axiom systems and logics to be employed throughout the paper. A reader already familiar with these basics is still encouraged to skim through the section because the justification language we use is not entirely standard (e.g., constants are divided into levels, and there is a new operation $\bar{?}$).

We start by recalling the languages of modal and justification logics. For modal formulas, we adopt the negation normal form, with conjunction and disjunction as primary propositional connectives. The negation normal form makes possible the use of *one-sided* nested sequent calculi for modal logics, which is more common and also minimizes the number of propositional sequent rules, thereby shortening our proofs. At the same time, justification formulas are given in a more traditional format, with falsum and implication as primary propositional connectives. As a result, the process of realization also encompasses a Boolean translation between two complete systems of propositional connectives. Not distinguishing between primary and defined connectives in either language enables us to perform these translations implicitly, except for cases where a Boolean transformation affects justification terms.

Modal language. *Modal formulas* are given by the grammar

$$A ::= P_i \mid \neg P_i \mid (A \vee A) \mid (A \wedge A) \mid \Box A \mid \Diamond A \ ,$$

where i ranges over positive natural numbers, P_i denotes a *proposition*, and $\neg P_i$ denotes its *negation*. The negation operation is extended from propositions to all formulas by means of the usual De Morgan laws, with $\neg\neg P_i := P_i$. Using this negation operation, we define $(A \to B) := (\neg A \vee B)$. Equivalence is defined as usual, and $\bot := (P_j \wedge \neg P_j)$ for some fixed proposition P_j.

Justification language. Apart from formulas, the language of justification logic has another type of syntactic objects called *justification terms*, or simply *terms*, that are given by the grammar

$$t ::= c_i^j \mid x_i \mid (t \cdot t) \mid (t + t) \mid\, !\,t \mid ?\,t \mid \bar{?}\,t \ ,$$

where i and j range over positive natural numbers, c_i^j denotes a (*justification*) *constant of level* j, and x_i denotes a (*justification*) *variable*. The binary operations $\cdot$ and $+$, which are left-associative, are called *application* and *sum* respectively. The unary operations !, ?, and $\bar{?}$ are called *positive introspection* (or *proof checker*), *negative introspection*, and *weak negative introspection* respectively. Terms that do not contain variables are called *ground* and are denoted by p, with or without a sub- and/or a superscript, whereas arbitrary terms are denoted by t and s, with or without a sub- and/or a superscript. We use the notation $t(x_{i_1}, \dots, x_{i_n})$ for terms that do not contain variables other than $x_{i_1}, \dots, x_{i_n}$.

Justification formulas are given by the grammar

$$A ::= P_i \mid \bot \mid (A \to A) \mid t : A \ ,$$

where P_i denotes a proposition, as in the modal language, and t is a justification term. The remaining Boolean connectives are defined as usual. While writing formulas, we assume that implication is right-associative and that both conjunction and disjunction bind stronger than implication.

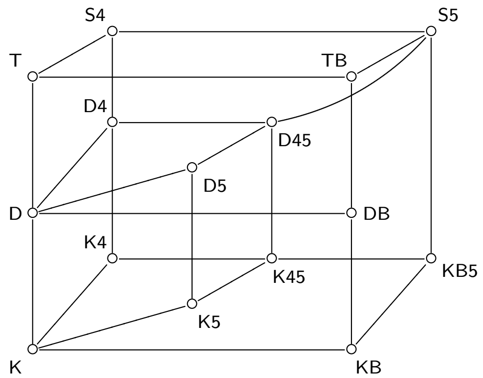

Figure 1: The *modal cube*

taut: A fixed complete set of propositional axioms

distr: $\Box(A \to B) \to (\Box A \to \Box B)$

$$\mathsf{MP}\ \frac{A \quad A \to B}{B} \qquad\qquad \mathsf{N}\ \frac{A}{\Box A}$$

Figure 2: The axiom system for the basic normal modal logic K

Modal logics and their axiom systems. One of our goals is to prove a uniform realization theorem for all modal logics in the so-called *modal cube* from [18] (see Figure 1). All these logics are extensions of the basic normal modal logic K that are obtained by taking its axiom system from Figure 2 and adding to it the modal axioms d, t, b, 4, and 5 from Figure 3 in various combinations. Figure 1 contains only 15 logics for $2^5 = 32$ such axiom systems because several axiom systems may yield one modal logic. For the modal logics with variant axiomatizations, we distinguish these axiomatizations because we realize them individually, thereby providing alternative realizations for such modal logics. To this end, we adopt the following naming conventions. Axiom systems are denoted by listing the (always present) axiom k, followed by the names of the axioms added to the axiom system for K from Figure 2, with all letters capitalized. For example, KD45 is the axiom system with additional axioms d, 4, and 5. If a logic from the cube has only one such axiom system, we use the same notation for both the logic and the axiom system, except that some logics traditionally have the initial letter 'K' omitted from their names: for instance, the logic of the axiom system KD45 is often called D45.

Two of the logics predate this modular axiomatization and, hence, bear traditional names S4 and S5. The former is the logic of the axiom systems KT4 and KDT4, while the latter is the logic of the following 13 axiom systems: KT5, KDT5, KDB4, KTB4, KDTB4, KDB5, KTB5, KDTB5, KT45, KDT45, KDB45, KTB45, and KDTB45. Further, the three axiom systems KB4, KB5, and KB45 produce the same modal logic, which, following [18], we call KB5. Thus, there is a small ambiguity between the logic KB5 and the axiom system KB5, which will be resolved by explicit typification, as in this sentence. Finally, the

d:	$\Box\bot \to \bot$	t:	$\Box A \to A$	b:	$A \to \Box\neg\Box\neg A$
jd:	$t : \bot \to \bot$	jt:	$t : A \to A$	jb:	$A \to \bar{?}\, t : (\neg t : \neg A)$
4:	$\Box A \to \Box\Box A$	5:	$\neg\Box A \to \Box\neg\Box A$		
j4:	$t : A \to\, !t : t : A$	j5:	$\neg t : A \to\, ?t : (\neg t : A)$		

Figure 3: Modal axioms and their corresponding justification axioms

taut:	A fixed finite complete set of propositional axioms
app:	$s : (A \to B) \to (t : A \to (s \cdot t) : B)$
sum:	$s : A \to (s + t) : A$ and $s : A \to (t + s) : A$

$$\mathsf{MP}\ \frac{A \quad A \to B}{B} \qquad\qquad \mathsf{iAN}\ \frac{A \text{ is an axiom instance}}{c^n_{i_n} : c^{n-1}_{i_{n-1}} : \ldots : c^1_{i_1} : A}$$

Figure 4: The axiom system for the basic justification logic J

axiom systems KT and KDT produce the same modal logic, as do the axiom systems KTB and KDTB. The traditional names for these logics are M and B respectively. To avoid confusing the latter with the logic KB, where the initial letter is omitted, we use TB instead of B. By analogy, T is used instead of M.

Justification logics and their axiom systems. The 15 modal logics of the modal cube are realized into 24 justification logics that we similarly define as extensions of the basic justification logic J. Its axiom system, also denoted J, consists of the axioms and rules given in Figure 4; the iAN-rule is called *iterated axiom necessitation.* We define the zero-premise iAN-rule as a rule and not as an axiom to prevent it from referring to itself. The finiteness of the set of propositional axioms in taut is required for the results on embedding in Section 6 (it is also a standard requirement for proving decidability and estimating complexity of justification logics). To define *extensions of the system* J, we add to its axiom system the justification axioms jd, jt, jb, j4, and j5 from Figure 3 in various combinations.

The axioms j4 and jt occur already in Artemov [1]; jd and j5 were introduced by Brezhnev [7] and by Pacuit [22] respectively. The axiom jb, as presented here, is new but has been independently proposed by Meghdad Ghari in an unpublished manuscript. The idea to use a new operation $\bar{?}$ rather than rebrand ? to mimic the modal axiom b is consistent with the general policy that incomparable axioms should be realized via different operations (cf. Remark 6.19).

Remark 2.1 (Alternative axiomatizations). Axiomatizations of justification logics that contain the axiom j4 often use a simpler version of the iAN-rule, called *axiom necessitation*:

$$\frac{A \text{ is an axiom instance}}{c^1_i : A} \ .$$

Since we are interested in the relationships among justification logics, it is more natural to use the form of axiom necessitation suitable for all justification logics rather than switch between different versions of the rule (cf. also [3, 15]).

Remark 2.2 (Levels of constants). The assignment of levels to constants is useful for proving the results on embedding in Section 6. A similar concept of levels was introduced in [21] (see also the definition of constant specification in [3]). Levels would not be needed for justification logics that contain the axiom j4 if we had chosen the rule from Remark 2.1 instead of iAN.

Remark 2.3 (Common language). We have decided to use a common language with all five operations on justifications for all justification logics to avoid cluttered formulations of lemmas and theorems that apply to all justification logics. For instance, the operation !, present in the common language, does not occur in Figure 4 and, hence, has no special meaning for the logic J. As a side effect, in this language, it is not possible to formulate conservativity results for justification logics. Instead of conservativity results, we introduce a more elaborate relationship among logics that is based on translation of operations rather than on their presence/absence in the language.

Naming conventions. The naming conventions for justification logics and their axiom systems are similar to those for modal logics. For example, the axiom system JB5 is J extended by the axioms jb and j5, and its logic is also denoted JB5. The only exceptions from the one-axiom-system-per-justification-logic rule are due to the fact that all instances of the axiom jd are also instances of jt. Hence, adding the axiom jd to an axiom system that already contains jt does not change the logic, thereby creating for it a second axiomatization. Accordingly, we omit the letter 'D' from the names of all the 8 logics with two axiom systems each. For instance, the logic JT5 is the logic of the axiom systems JT5 and JDT5. Note that in all the other cases, every two axiom systems yield different logics simply because their sets of axioms are different and so are their sets of provable formulas of the form $c_i^1 : A$, where A is an axiom instance and c_i^1 is a constant of level 1.

Unless stated otherwise, from this point on, by a *justification logic* we mean the logic of either the axiom system J or one of its extensions. Likewise, by a *modal axiom system* we mean either the axiom system K or one of its extensions, and by a *modal logic* we mean the logic of a modal axiom system. We denote an arbitrary modal axiom system, modal logic, and justification logic by AS, ML, and JL respectively.

We have named the axiom systems in such a way that each modal axiom system has a natural *corresponding* justification axiom system, and vice versa. The names of corresponding systems differ only in the first letter: K for a modal axiom system and J for a justification one. For example, KT45 corresponds to JT45.

Realization theorems. A deeper *correspondence* between modal and justification logics is established by realization theorems. The first realization theorem was proved by Artemov [1, 2] for the modal logic S4. It connects S4 with a justification logic that he called LP, or the *Logic of Proofs*, and that we mostly refer to as JT4 (note that JT4 is indeed the justification axiom system that corresponds to KT4, one of the axiom systems of S4).

Realization theorems are formulated using a natural translation function from justification to modal formulas:

Definition 2.4 (Forgetful projection and realization). Given a justification formula A, its *forgetful projection* A° is defined by induction on the structure of A:

$$P_i^\circ := P_i, \quad \bot^\circ := \bot, \quad (A \to B)^\circ := A^\circ \to B^\circ, \quad \text{and} \quad (t : A)^\circ := \Box A^\circ.$$

The *forgetful projection of a set* X of justification formulas is the set of their forgetful projections: $X^\circ := \{A^\circ \mid A \in X\}$. A justification logic JL *realizes* a modal logic ML if $\mathsf{JL}^\circ = \mathsf{ML}$: i.e., if the forgetful projection of the set of theorems of JL is exactly the set of theorems of ML.

In the next section, we impose an additional standard restriction on realizations: namely, diamonds (i.e., negative boxes) should be realized by distinct variables.

To date, no systematic study exists of the effects of variant axiomatizations of a modal logic on its realizations. In this paper, we present such a study and provide realizations that are based on alternative modal axiomatizations and are *modular* in the following sense: given an axiom system AS for a modal logic ML, the justification axiom system that corresponds to AS yields a justification logic that realizes ML. To this end, we say that every modal logic ML has one or several *justification counterparts*, i.e., the justification logics of justification axiom systems that correspond to one of the modal axiom systems of ML. In particular, the justification counterparts of KB5 are JB4, JB5, and JB45. The ones for S5 are JT5, JTB5, JDB5, JT45, JTB45, JDB45, JTB4, and JDB4. Every other modal logic has exactly one justification counterpart, e.g., JD45 for D45.

3. Preparation for Realization

Proving realization theorems involves turning provable formulas of a given modal logic into provable formulas of a corresponding justification logic by replacing occurrences of $\Box$ with terms and of $\Diamond$ with variables. We employ an induction on a given sequent-style derivation. In order to describe this constructive procedure, we introduce *realization functions* that assign terms to modalities. To distinguish between different occurrences of modalities in a formula, we *annotate* them with distinct natural numbers, using parity to distinguish between $\Box$'s and $\Diamond$'s. These annotations, which we adopt and adapt from [16], are

$(P_i)^r$	$:=$	P_i	$(A \vee B)^r$	$:=$	$A^r \vee B^r$	$(\Diamond_{2l} A)^r$	$:=$	$\neg r(2l) : \neg A^r$
$(\neg P_i)^r$	$:=$	$\neg P_i$	$(A \wedge B)^r$	$:=$	$A^r \wedge B^r$	$(\Box_{2k-1} A)^r$	$:=$	$r(2k-1) : A^r$

Figure 5: Realization of annotated formulas

purely syntactic devices and have no semantic meaning. In this section, we also describe technical machinery to be used for operating with realization functions, including their interaction with substitutions. In addition, we formulate the Internalization Property (Lemma 3.4 and Corollary 3.5) enjoyed by all the justification logics, which is necessary for proving realization theorems, and state the Merging Theorem by Fitting (Theorem 3.11), which plays a major role in our method of realization.

Definition 3.1 (Annotations). *Annotated modal formulas*, or simply *annotated formulas*, are given by the grammar

$$A ::= P_i \mid \neg P_i \mid (A \vee A) \mid (A \wedge A) \mid \Box_{2k-1} A \mid \Diamond_{2l} A \ ,$$

where i, k, and l range over positive natural numbers, P_i and $\neg P_i$ denote a proposition and its negation, as in the unannotated modal language. If A' is a modal formula obtained from an annotated formula A by dropping all indices on its modalities, then we call A an *annotated version* of A'. An annotated formula is called *properly annotated* if no index occurs twice in it.

We mostly work with properly annotated formulas, for which the use of negation normal form has a positive effect of every subformula of a properly annotated formula being itself properly annotated, in contrast to [16].

Remark 3.2 (Negation and substitution of annotated formulas). Note that it is not clear how to define the negation operation for annotated formulas. The obvious definition of $\neg\Box_k A$ as $\Diamond_k \neg A$ does not work because it does not produce an annotated formula. In particular, the substitution of annotated formulas for propositions is only possible for positive, i.e., non-negated, propositions.

We now define realizations as functions from positive natural numbers to terms, with a restriction that the set of even numbers is in one-to-one correspondence with the set of variables. This restriction, which is called the *normality condition*, is standard and corresponds to the intuition that $\Diamond$'s (or negatively occurring boxes if $\neg$ is a primary connective instead of $\Diamond$) represent assumptions on what should be provable and that they become Skolem variables if $\Box$'s, existentially read as '$\exists$ a proof,' are skolemized.

Definition 3.3 (Realization function). A *prerealization function* r is a partial mapping from positive natural numbers to terms. A prerealization function r is called a *realization function* if $r(2l) = x_l$ whenever $r(2l)$ is defined. A *(pre)realization function on a given annotated formula* is one that is defined on all indices of that formula.

If A is an annotated formula and r is a prerealization function on A, then the justification formula A^r is inductively defined as in Figure 5. Note that if r is a realization function on $\Diamond_{2l} A$, then $(\Diamond_{2l} A)^r = \neg x_l : \neg A^r$. Further, note that every justification formula B can be written as $B = A^r$ for some properly annotated formula A and some prerealization function r.

A basic feature of justification logics used extensively in this paper is the Internalization Property, which enables one to *internalize* as a term any proof of a formula B, with or without hypotheses. This is formally stated in the lemma below, originally proved for LP [2].

Lemma 3.4 (Internalization). For any justification logic JL, if

$$A_1, \ldots, A_n \vdash_{\mathsf{JL}} B \ , \tag{1}$$

then there exists a term $t(x_1, \ldots, x_n)$ such that

$$s_1 : A_1, \ldots, s_n : A_n \vdash_{\mathsf{JL}} t(s_1, \ldots, s_n) : B$$

for all terms $s_1, \ldots, s_n$. Note that the term t is ground if $n = 0$.

Proof sketch. This can be easily proved by induction on JL-proof (1). For an axiom, the term t is taken to be a constant of level 1. For an instance of iAN with the outermost constant of level n, the term t is taken to be a constant of level $n+1$. For a hypothesis A_i, the term $t := x_i$. For a conclusion D of the MP-rule with premises $C \to D$ and C, there must exist terms t_1 for $C \to D$ and t_2 for C. The term for D is taken to be $t := t_1 \cdot t_2$. □

In our realization proof, we mostly use the following form of Internalization, obtained by using the rule MP and the *Deduction Theorem*. The proof of the latter for justification logics can be almost literally adopted from that for classical propositional logic since MP remains the only rule with premises.

Corollary 3.5 (Internalization). For any justification logic JL, if

$$\mathsf{JL} \vdash A_1 \to \ldots \to A_n \to B \ ,$$

then there exists a term $t(x_1, \ldots, x_n)$ such that

$$\mathsf{JL} \vdash s_1 : A_1 \to \ldots \to s_n : A_n \to t(s_1, \ldots, s_n) : B$$

for all terms $s_1, \ldots, s_n$. The term t is ground if $n = 0$.

Our general method for proving realization theorems is by induction on the depth of a proof in a nested sequent system (to be introduced later) for a modal logic. Since realizations of side formulas need not be the same in different premises of branching rules, these realizations need to be reconciled, which will be done using Fitting's merging technique [16]. In order to formulate it, we need additional notation and definitions, especially the notion of substitution, which also plays an important role in the realization procedure itself.

Definition 3.6 (Additional notation). Let A be an annotated formula and r be a prerealization function. We define

$$\mathrm{vars}_{\Diamond}(A) := \{x_k \mid \Diamond_{2k} \text{ occurs in } A\} \ ,$$
$$r \restriction A := r \restriction \{i \mid i \text{ occurs in } A\} \ ,$$

where $f \restriction S$ is the restriction of the partial function f to the set $S \cap \mathrm{dom}(f)$.

The following definition is mostly standard (see, e.g., [6]).

Definition 3.7 (Substitution). A *substitution*, denoted by σ, is a total mapping from variables to terms. For any term t, the term $t\sigma$ is inductively defined as follows: $c\sigma := c$ for any constant c, $x\sigma := \sigma(x)$ for any variable x, $(*t)\sigma := *(t\sigma)$ for any unary operation $*$, and $(t_1 * t_2)\sigma := (t_1\sigma) * (t_2\sigma)$ for any binary operation $*$. We write $A\sigma$ for the formula that is obtained from A by simultaneously replacing every term t in A with $t\sigma$.

The definition of domain for substitutions differs from the standard one for ordinary functions, such as prerealization functions. The *domain of* σ is $\mathrm{dom}(\sigma) := \{x \mid \sigma(x) \neq x\}$. The *variable range of* σ, denoted by $\mathrm{vrange}(\sigma)$, is the set of variables that occur in terms from the set $\{\sigma(x) \mid x \in \mathrm{dom}(\sigma)\}$.

Composition of substitutions is defined as $(\sigma_2 \circ \sigma_1)(x) := \sigma_1(x)\sigma_2$ for any variable x. Composition of a substitution with a prerealization function is defined as $(\sigma \circ r)(n) := r(n)\sigma$; in particular, $(\sigma \circ r)(n)$ is undefined whenever $r(n)$ is. Finally, for substitutions σ_1 and σ_2 with disjoint domains, i.e., with $\mathrm{dom}(\sigma_1) \cap \mathrm{dom}(\sigma_2) = \varnothing$, their union is a substitution defined as follows:

$$(\sigma_1 \cup \sigma_2)(x) := \begin{cases} \sigma_1(x) & \text{if } x \in \mathrm{dom}(\sigma_1), \\ \sigma_2(x) & \text{if } x \in \mathrm{dom}(\sigma_2), \\ x & \text{otherwise.} \end{cases}$$

A substitution σ *lives on an annotated formula* A if $\mathrm{dom}(\sigma) \subseteq \mathrm{vars}_{\Diamond}(A)$. A substitution σ *lives away from an annotated formula* A if $\mathrm{dom}(\sigma) \cap \mathrm{vars}_{\Diamond}(A) = \varnothing$.

The following lemma is easily proved by induction on a proof of A (see, e.g., [20]).

Lemma 3.8 (Substitution). If $\mathsf{JL} \vdash A$ for a justification logic JL, then

(1) $\mathsf{JL} \vdash A\sigma$ for any substitution σ and

(2) $\mathsf{JL} \vdash A[P_{i_1} \mapsto B_1, \dots, P_{i_n} \mapsto B_n]$, where $A[P_{i_1} \mapsto B_1, \dots, P_{i_n} \mapsto B_n]$ denotes the result of simultaneously replacing all occurrences of the propositions $P_{i_1}, \dots, P_{i_n}$ in A with the formulas $B_1, \dots, B_n$ respectively.

Remark 3.9 (Simultaneous substitution). In Lemma 3.8 (2), we formulate simultaneous substitution of several formulas for propositions. Naturally, it would have been sufficient to allow only a single such substitution at a time, but this would have resulted in more cumbersome proofs later on when this lemma is actually used, e.g., in Lemma 5.11. In addition, the proof for the simultaneous version is exactly the same as for the single-proposition version, and the given formulation is more in line with Lemma 3.8 (1).

Since the process of realizing a modal formula starts with annotating it, a priori the realizability of the formula might depend on the annotation chosen. The following lemma shows that this is not the case.

Lemma 3.10 (Renaming Annotations). Let JL be a justification logic, A_1 and A_2 be properly annotated versions of the same modal formula A', and r_1 be a realization function on A_1 with $\mathsf{JL} \vdash (A_1)^{r_1}$. Then there exists a realization function r_2 on A_2 such that $\mathsf{JL} \vdash (A_2)^{r_2}$.

Proof. For every index n of A_1, let n' denote the corresponding index of A_2. Since both A_1 and A_2 are properly annotated, n' has the same parity as n. Let the substitution σ be defined as follows:

$$\sigma(x_m) := \begin{cases} x_n & \text{if } 2m \text{ is an index of } A_1 \text{ and } (2m)' = 2n, \\ x_m & \text{otherwise.} \end{cases}$$

For every $n > 0$, let the realization function r_2 be defined as follows:

$$r_2(n) := \begin{cases} x_m & \text{if } n = 2m \text{ is an index of } A_2, \\ r_1(m)\sigma & \text{if } n \text{ is an odd index of } A_2 \text{ and } m' = n, \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Clearly, r_2 is a realization function on A_2.

We show by induction on the structure of A' that $(A_1)^{r_1}\sigma = (A_2)^{r_2}$. It then follows by Substitution Lemma 3.8 that $(A_2)^{r_2}$ is provable in JL. The base case and the propositional cases are trivial.

Let $A' = \Box B'$. Then $A_1 = \Box_m B_1$ and $A_2 = \Box_n B_2$ for some odd indices m and n with $m' = n$ and for some properly annotated formulas B_1 and B_2, both annotated versions of B'. Then $r_2(n) = r_1(m)\sigma$ by definition of r_2. By induction hypothesis, $(B_1)^{r_1}\sigma = (B_2)^{r_2}$. Therefore,

$$(\Box_m B_1)^{r_1}\sigma = r_1(m)\sigma : (B_1)^{r_1}\sigma = r_2(n) : (B_2)^{r_2} = (\Box_n B_2)^{r_2} \ .$$

Let $A' = \Diamond B'$. Then $A_1 = \Diamond_{2m} B_1$ and $A_2 = \Diamond_{2n} B_2$ for some indices $2m$ and $2n$ with $(2m)' = 2n$ and for some properly annotated formulas B_1 and B_2, both annotated versions of B'. Then $x_m\sigma = x_n$ by definition of σ. By induction hypothesis, $(B_1)^{r_1}\sigma = (B_2)^{r_2}$. Therefore,

$$(\Diamond_{2m} B_1)^{r_1}\sigma = \neg x_m\sigma : \neg(B_1)^{r_1}\sigma = \neg x_n : \neg(B_2)^{r_2} = (\Diamond_{2n} B_2)^{r_2} \ .$$ □

We now formulate the merging theorem, which is an instance of Theorem 8.2 from [16]. There it is formulated for LP, but the proof makes use only of the operations $+$ and $\cdot$ and of the Internalization Property. Hence, the theorem also holds for all justification logics we consider.

Theorem 3.11 (Realization Merging). Let JL be a justification logic, A be a properly annotated formula, and $r_1, \dots, r_n$ be realization functions on A. Then there exists a realization function r on A and a substitution σ that lives on A such that $\mathsf{JL} \vdash A^{r_i}\sigma \to A^r$ for $i = 1, \dots, n$. (Note that it is not assumed that the A^{r_i}'s are provable.)

The following properties are used, often implicitly, in many of the proofs in this paper.

Fact 3.12 (Combinations of Substitutions and Realization Functions). Let A be an annotated formula, σ, σ_1, and σ_2 be substitutions, and r be a prerealization function.

(1) $\sigma_2 \circ \sigma_1$ is a substitution with $\mathrm{dom}(\sigma_2 \circ \sigma_1) \subseteq \mathrm{dom}(\sigma_1) \cup \mathrm{dom}(\sigma_2)$ and $\mathrm{vrange}(\sigma_2 \circ \sigma_1) \subseteq \mathrm{vrange}(\sigma_1) \cup \mathrm{vrange}(\sigma_2)$. Moreover, $A(\sigma_2 \circ \sigma_1) = (A\sigma_1)\sigma_2$;

(2) if $\mathrm{dom}(\sigma_1) \cap \mathrm{dom}(\sigma_2) = \varnothing$, then $\mathrm{dom}(\sigma_1 \cup \sigma_2) = \mathrm{dom}(\sigma_1) \cup \mathrm{dom}(\sigma_2)$;

(3) if $\mathrm{dom}(\sigma_1) \cap \mathrm{dom}(\sigma_2) = \varnothing$ and $\mathrm{vrange}(\sigma_1) \cap \mathrm{dom}(\sigma_2) = \varnothing$, then $\sigma_1 \cup \sigma_2 = \sigma_2 \circ \sigma_1$;

(4) $\sigma \circ r$ is a prerealization function with $\mathrm{dom}(\sigma \circ r) = \mathrm{dom}(r)$;

(5) if r is a prerealization function on A, then so is $\sigma \circ r$ and $A^{\sigma \circ r} = A^r\sigma$;

(6) if r is a (pre)realization function on A, then so is $r \restriction A$.

Whenever r, r_1, and r_2 are realization functions,

(7) if $\mathrm{dom}(r_1) \cap \mathrm{dom}(r_2) \subseteq \{n \mid n \text{ is even}\}$, then $r_1 \cup r_2$ is a realization function;

(8) if $r_1 \cup r_2$ is a realization function, then $\mathrm{dom}(r_1 \cup r_2) = \mathrm{dom}(r_1) \cup \mathrm{dom}(r_2)$;

(9) $\sigma \circ r$ is a realization function iff $x_n \notin \mathrm{dom}(\sigma)$ whenever $r(2n)$ is defined.

Corollary 3.13. If r is a realization function on an annotated formula A and if a substitution σ lives away from A, then $\sigma \circ (r \restriction A)$ is a realization function on A.

4. A General Realization Method for Nested Sequent Systems

In this section, we introduce the formalism of nested sequent calculi and describe a general framework for proving realization theorems based on such calculi. The essence of the method is that realizing arbitrary nested sequent rules can be reduced to realizing their non-nested (or *shallow*) versions (Lemma 4.11), which is even simpler than realizing rules of an ordinary sequent calculus. As a consequence, in order to prove a realization theorem for a modal logic presented via a nested sequent system, it is sufficient to realize the shallow versions of all the rules of the system (Theorem 4.12). The realization of various (shallow versions of) nested sequent rules and proofs of actual realization theorems are postponed until Section 5.

Nested sequents. *Nested sequents*, or simply *sequents*, are inductively defined as follows: the empty sequence $\varnothing$ is a nested sequent; if Σ and Δ are nested sequents and A is a modal formula, then Σ, A and $\Sigma, [\Delta]$ are nested sequents, where the comma denotes concatenation. The brackets of the expression $[\Delta]$ are called *structural box*. The *corresponding formula* of a sequent Γ, denoted $\underline{\Gamma}$, is inductively defined as follows:

$$\underline{\varnothing} := \bot; \qquad \underline{\Sigma, A} := \begin{cases} \underline{\Sigma} \vee A & \text{if } \Sigma \neq \varnothing, \\ A & \text{otherwise;} \end{cases} \qquad \underline{\Sigma, [\Delta]} := \begin{cases} \underline{\Sigma} \vee \Box\underline{\Delta} & \text{if } \Sigma \neq \varnothing, \\ \Box\underline{\Delta} & \text{otherwise.} \end{cases} \tag{2}$$

We use the letters Γ, Δ, Λ, Π, and Σ with or without a sub- and/or a superscript to denote sequents.

Sequent contexts. A *sequent context*, or simply *context*, is a sequent with exactly one occurrence of the symbol $\{\,\}$, called a *hole*, which does not occur inside formulas. Contexts are denoted by $\Gamma\{\,\}$. An inductive definition can be given as follows: $\{\,\}$ is a context and if $\Sigma\{\,\}$ is a context, then so are $[\Sigma\{\,\}]$ and $\Delta, \Sigma\{\,\}, \Pi$, where Δ and Π are sequents. For a context $\Gamma\{\,\}$ and a sequent Δ, the sequent $\Gamma\{\Delta\}$ is obtained by replacing the hole in $\Gamma\{\,\}$ with Δ. For example, if $\Gamma\{\,\} = A, [[B], \{\,\}]$ and $\Delta = C, [D]$, then $\Gamma\{\Delta\} = A, [[B], C, [D]]$.

Sequent contexts are used to formulate nested rules. As an example, the nested version of the exchange rule can be formulated as follows:

$$\mathsf{exch}\,\frac{\Gamma\{\Delta, \Sigma\}}{\Gamma\{\Sigma, \Delta\}} \quad . \tag{3}$$

One of the instances of (3) is $\mathsf{exch}\,\dfrac{[P_2 \wedge \Diamond P_3, [P_1], P_1], [P_1, \neg P_1]}{[[P_1], P_1, P_2 \wedge \Diamond P_3], [P_1, \neg P_1]}$, where context $\Gamma\{\ \} = [\{\ \}], [P_1, \neg P_1]$ and sequents $\Delta = P_2 \wedge \Diamond P_3$ and $\Sigma = [P_1], P_1$.

In the next section, we provide systems of such rules for all the logics in the modal cube and use these systems to prove realization theorems for these logics. In this section, however, we treat arbitrary *context-preserving* nested rules, i.e., rules of the form

$$\frac{\Gamma\{S_1\} \quad \ldots \quad \Gamma\{S_n\}}{\Gamma\{S\}}\ ,$$

where n is a non-negative integer, $\Gamma\{\ \}$ denotes an arbitrary context, common for all the premises and the conclusion of the rule, and $S, S_1, \ldots, S_n$ are sequent schemas. Each context-preserving nested rule ρ has a *shallow version* $\mathsf{sh}\text{-}\rho$ that corresponds to the common context being empty, $\Gamma\{\ \} = \{\ \}$:

$$\frac{S_1 \quad \ldots \quad S_n}{S}\ .$$

For instance, the shallow version of the nested exchange rule (3) is $\mathsf{sh\text{-}exch}\,\dfrac{\Delta, \Sigma}{\Sigma, \Delta}$. From now on, by a *nested rule* we mean a context-preserving nested rule.

Contexts provide for an especially simple definition of subsequents:

Definition 4.1 (Subsequent). A *subsequent* of a given sequent Γ is any sequent Δ such that $\Gamma = \Sigma\{\Delta\}$ for some context $\Sigma\{\ \}$.

Definition 4.2 (Annotated sequent). An *annotated sequent* (*context*) is a sequent (context) in which only annotated formulas occur and all structural boxes are annotated by odd indices. The *corresponding formula* of an annotated sequent is an annotated formula defined as in (2), except that the third case is replaced with

$$\underline{\Sigma, [\Delta]_k} := \begin{cases} \underline{\Sigma} \vee \Box_k \underline{\Delta} & \text{if } \Sigma \neq \varnothing, \\ \Box_k \underline{\Delta} & \text{otherwise.} \end{cases}$$

Remark 4.3 (Notions extended from formulas to sequents). Many notions, such as an *annotated version* and *proper annotation*, naturally apply to sequents as well. Other notions are extended from (annotated) formulas to (annotated) sequents by being applied to the corresponding formula of the (annotated) sequent. For instance, a *realization function on an annotated sequent* Γ is a realization function on $\underline{\Gamma}$, $\Gamma^r := (\underline{\Gamma})^r$, $\mathrm{vars}_\Diamond(\Gamma) := \mathrm{vars}_\Diamond(\underline{\Gamma})$, etc.

Whenever safe, we do not explicitly distinguish between an annotated formula A and the annotated sequent that consists of this formula A: e.g., r is a realization function on a formula A iff it is a realization function on the sequent A, which enables us to call it simply a *realization function on* A.

We often use the following trivial fact without mentioning it explicitly:

Fact 4.4 (Preservation of Structure in Annotated Versions). If an annotated sequent Δ is an annotated version of $\Gamma'\{\Lambda'\}$ for some context $\Gamma'\{\ \}$ and some sequent Λ', there exists a unique annotated version $\Gamma\{\ \}$ of the context $\Gamma'\{\ \}$ and a unique annotated version Λ of the sequent Λ' such that $\Delta = \Gamma\{\Lambda\}$. Moreover, if Δ is properly annotated, so is Λ.

If an annotated sequent Γ is an annotated version of a sequent Γ', then its corresponding formula $\underline{\Gamma}$ is an annotated version of $\underline{\Gamma'}$.

A realization function on a formula A is trivially a realization function on any subformula of A; the same is true for sequents and their subsequents. Note, however, that realization functions are defined on corresponding formulas rather than on sequents themselves and that $\underline{\Delta}$ is not in general a subformula of $\underline{\Gamma\{\Delta\}}$. The following fact will be used as a matter of course without explicit mention.

Fact 4.5 (Realization Function on a Subsequent). If r is a realization function on an annotated sequent $\Gamma\{\Delta\}$, then r is also a realization function on its subsequent Δ.

The following lemma can be easily obtained from the associativity of Boolean disjunction by induction on the structure of Γ. The lemma is needed because, in general, the formula $\underline{\Gamma, \Sigma}$ does not coincide with the formula $\underline{\Gamma} \vee \underline{\Sigma}$.

Lemma 4.6 (Associativity of Disjunction). For any annotated sequents Γ and Σ, for any realization function r on Σ, Γ, and for any substitution σ, we have $\mathsf{J} \vdash (\Sigma, \Gamma)^r \sigma \;\leftrightarrow\; \Sigma^r \sigma \vee \Gamma^r \sigma$.

Definition 4.7 (Annotated rule instance). Given an instance of a nested rule

$$\frac{\Gamma'\{\Lambda'_1\} \quad \dots \quad \Gamma'\{\Lambda'_n\}}{\Gamma'\{\Lambda'\}} \ ,$$

with common context $\Gamma'\{\ \}$, an *annotated version* of this instance is of the form

$$\frac{\Gamma\{\Lambda_1\} \quad \dots \quad \Gamma\{\Lambda_n\}}{\Gamma\{\Lambda\}} \ ,$$

where $\Gamma\{\ \}$, $\Lambda_1, \dots, \Lambda_n$, and Λ are annotated versions of $\Gamma'\{\ \}$, $\Lambda'_1, \dots, \Lambda'_n$, and Λ' respectively, sequents $\Gamma\{\Lambda_1\}, \dots, \Gamma\{\Lambda_n\}$, and $\Gamma\{\Lambda\}$ are properly annotated, and no index occurs in both Λ_i and Λ_j for any $1 \leq i < j \leq n$. Note that the annotated context $\Gamma\{\ \}$ is the same for every premise and the conclusion.

Definition 4.8 (Realizable rule). An instance $\dfrac{}{\Gamma'\{\Lambda'\}}$ of a 0-premise nested rule is called *realizable* in a justification logic JL if there exists an annotated version $\dfrac{}{\Gamma\{\Lambda\}}$ of it and a realization function r on $\Gamma\{\Lambda\}$ such that $\mathsf{JL} \vdash \Gamma\{\Lambda\}^r$. An instance $\dfrac{\Gamma'\{\Lambda'_1\} \quad \dots \quad \Gamma'\{\Lambda'_n\}}{\Gamma'\{\Lambda'\}}$ of an n-premise nested rule with $n > 0$ and with common context $\Gamma'\{\}$ is called *realizable* in JL if there exists an annotated version $\dfrac{\Gamma\{\Lambda_1\} \quad \dots \quad \Gamma\{\Lambda_n\}}{\Gamma\{\Lambda\}}$ of it such that for any realization functions $r_1, \dots, r_n$ on $\Gamma\{\Lambda_1\}, \dots, \Gamma\{\Lambda_n\}$ respectively, there exists a realization function r on $\Gamma\{\Lambda\}$ and a substitution σ that lives on each of $\Gamma\{\Lambda_i\}$, $i = 1, \dots, n$, such that

$$\mathsf{JL} \vdash \Gamma\{\Lambda_1\}^{r_1}\sigma \;\rightarrow\; \dots \;\rightarrow\; \Gamma\{\Lambda_n\}^{r_n}\sigma \;\rightarrow\; \Gamma\{\Lambda\}^r \ .$$

A rule is called *realizable* in JL if all its instances are realizable in JL.

The following fact trivially follows from the definition.

Fact 4.9 (Realizability in Extensions). If a nested rule is realizable in a justification logic JL, then it is also realizable in every extension of JL.

Remark 4.10 (Realizability of cut). Currently it is not known whether the cut rule is realizable in J or in some of its extensions. A more sophisticated definition of realizability may be necessary. Fortunately, all sequent systems we use are cut-free.

Lemma 4.11 (From Shallow to Nested). For any nested rule ρ, if its shallow version $\mathsf{sh}\text{-}\rho$ is realizable in a justification logic JL, then ρ itself is also realizable in JL.

Proof. We prove the lemma for the harder case where ρ has $n > 0$ premises. The proof for the case when $n = 0$ is similar and, hence, omitted. We consider an arbitrary instance

$$\frac{\Delta'\{\Lambda'_1\} \quad \ldots \quad \Delta'\{\Lambda'_n\}}{\Delta'\{\Lambda'\}} \tag{4}$$

of ρ and show that it is realizable in JL. By assumption, its shallow version $\frac{\Lambda'_1 \quad \ldots \quad \Lambda'_n}{\Lambda'}$, which is an instance of $\mathsf{sh}\text{-}\rho$, has an annotated version $\frac{\Lambda_1 \quad \ldots \quad \Lambda_n}{\Lambda}$ such that for arbitrary realization functions $r_1, \ldots, r_n$ on $\Lambda_1, \ldots, \Lambda_n$ respectively, there exists a realization function r_0 on Λ and a substitution σ_0 that lives on each of Λ_i, $i = 1, \ldots, n$, such that

$$\mathsf{JL} \vdash (\Lambda_1)^{r_1}\sigma_0 \rightarrow \ldots \rightarrow (\Lambda_n)^{r_n}\sigma_0 \rightarrow \Lambda^{r_0} \ .$$

We prove a stronger statement, namely that for any annotated context $\Gamma\{\ \}$ such that $\Gamma\{\Lambda_1\}, \ldots, \Gamma\{\Lambda_n\}$, and $\Gamma\{\Lambda\}$ are properly annotated and for arbitrary realization functions $r_1, \ldots, r_n$ on $\Gamma\{\Lambda_1\}, \ldots, \Gamma\{\Lambda_n\}$ respectively, there exists a realization function r on $\Gamma\{\Lambda\}$ and a substitution σ that lives on each of $\Gamma\{\Lambda_i\}$, $i = 1, \ldots, n$, such that

$$\mathsf{JL} \vdash \Gamma\{\Lambda_1\}^{r_1}\sigma \rightarrow \ldots \rightarrow \Gamma\{\Lambda_n\}^{r_n}\sigma \rightarrow \Gamma\{\Lambda\}^{r} \ .$$

It then follows that the above also holds for some particular annotated context $\Gamma\{\ \} = \Delta\{\ \}$ such that $\frac{\Delta\{\Lambda_1\} \quad \ldots \quad \Delta\{\Lambda_n\}}{\Delta\{\Lambda\}}$ is an annotated version of our arbitrary ρ-instance (4). The proof is by induction on the structure of $\Gamma\{\ \}$.

Base case $\Gamma\{\ \} = \{\ \}$. Given realization functions $r_1, \ldots, r_n$ on $\Lambda_1, \ldots, \Lambda_n$ respectively, take $r := r_0$ and $\sigma := \sigma_0$.

Case $\Gamma\{\ \} = [\Sigma\{\ \}]_k$. Let $r_1, \ldots, r_n$ be realization functions on $[\Sigma\{\Lambda_1\}]_k, \ldots, [\Sigma\{\Lambda_n\}]_k$ respectively. Since $\Sigma\{\Lambda_1\}, \ldots, \Sigma\{\Lambda_n\}$, and $\Sigma\{\Lambda\}$ are properly annotated as subsequents of properly annotated sequents $[\Sigma\{\Lambda_1\}]_k, \ldots, [\Sigma\{\Lambda_n\}]_k$, and $[\Sigma\{\Lambda\}]_k$ respectively and since $r_1, \ldots, r_n$ are also realization functions on $\Sigma\{\Lambda_1\}, \ldots, \Sigma\{\Lambda_n\}$ respectively, by induction hypothesis, there exists a realization function r' on $\Sigma\{\Lambda\}$ and a substitution σ' that lives on each of $\Sigma\{\Lambda_i\}$ such that

$$\mathsf{JL} \vdash \Sigma\{\Lambda_1\}^{r_1}\sigma' \rightarrow \ldots \rightarrow \Sigma\{\Lambda_n\}^{r_n}\sigma' \rightarrow \Sigma\{\Lambda\}^{r'} \ . \tag{5}$$

By Internalization Property 3.5, there exists a term $t(x_1, \ldots, x_n)$ such that

$$\mathsf{JL} \vdash r_1(k)\sigma' : \big(\Sigma\{\Lambda_1\}^{r_1}\sigma'\big) \rightarrow \ldots \rightarrow r_n(k)\sigma' : \big(\Sigma\{\Lambda_n\}^{r_n}\sigma'\big) \rightarrow t\big(r_1(k)\sigma', \ldots, r_n(k)\sigma'\big) : \Sigma\{\Lambda\}^{r'} \ . \tag{6}$$

Let $\sigma := \sigma'$ and let

$$r := \big(r' \restriction \Sigma\{\Lambda\}\big) \cup \big\{k \mapsto t\big(r_1(k)\sigma', \ldots, r_n(k)\sigma'\big)\big\} \ .$$

Since $[\Sigma\{\Lambda\}]_k$ is properly annotated, index k does not occur in $\Sigma\{\Lambda\}$. Hence, $k \notin \mathrm{dom}\big(r' \restriction \Sigma\{\Lambda\}\big)$ and r is a realization function on $[\Sigma\{\Lambda\}]_k$ by Fact 3.12. Now (6) can be rewritten as

$$\mathsf{JL} \vdash \big([\Sigma\{\Lambda_1\}]_k\big)^{r_1}\sigma \rightarrow \ldots \rightarrow \big([\Sigma\{\Lambda_n\}]_k\big)^{r_n}\sigma \rightarrow \big([\Sigma\{\Lambda\}]_k\big)^{r} \ .$$

For each $i = 1, \ldots, n$, since σ' lives on $\Sigma\{\Lambda_i\}$, it is obvious that $\sigma = \sigma'$ lives on $[\Sigma\{\Lambda_i\}]_k$.

Case $\Gamma\{\ \} = \Delta, \Sigma\{\ \}, \Pi$. Let $r_1, \ldots, r_n$ be realization functions on

$$\Delta, \Sigma\{\Lambda_1\}, \Pi, \qquad \ldots, \qquad \Delta, \Sigma\{\Lambda_n\}, \Pi$$

respectively. As in the previous case, by induction hypothesis, there exists a realization function r' on $\Sigma\{\Lambda\}$ and a substitution σ' that lives on each of $\Sigma\{\Lambda_i\}$ such that (5) holds. It follows from Fact 4.5 that each r_i,

$i = 1, \dots, n$, is a realization function on Δ, Π. Since each $\Delta, \Sigma\{\Lambda_i\}, \Pi$ is properly annotated and σ' lives on each $\Sigma\{\Lambda_i\}$, it lives away from Δ, Π. Thus, by Corollary 3.13, $\sigma' \circ (r_i \restriction \Delta, \Pi)$ is a realization function on Δ, Π, for each $i = 1, \dots, n$. By Theorem 3.11 (Realization Merging), there exists a realization function r_M on Δ, Π and a substitution σ_M that lives on Δ, Π such that for each $i = 1, \dots, n$

$$\mathsf{JL} \vdash (\Delta, \Pi)^{\sigma' \circ (r_i \restriction \Delta, \Pi)} \sigma_M \;\to\; (\Delta, \Pi)^{r_M} \;. \tag{7}$$

By Fact 3.12 (5), $(\Delta, \Pi)^{\sigma' \circ (r_i \restriction \Delta, \Pi)} \sigma_M = (\Delta, \Pi)^{r_i} \sigma' \sigma_M$. Therefore, (7) can be rewritten as

$$\mathsf{JL} \vdash (\Delta, \Pi)^{r_i} \sigma' \sigma_M \;\to\; (\Delta, \Pi)^{r_M} \;. \tag{8}$$

From the induction hypothesis (5), it follows by the Substitution Lemma that

$$\mathsf{JL} \vdash \Sigma\{\Lambda_1\}^{r_1} \sigma' \sigma_M \;\to\; \dots \;\to\; \Sigma\{\Lambda_n\}^{r_n} \sigma' \sigma_M \;\to\; \Sigma\{\Lambda\}^{r'} \sigma_M \;.$$

From this and (8), it follows by propositional reasoning that

$$\mathsf{JL} \vdash \Sigma\{\Lambda_1\}^{r_1} \sigma' \sigma_M \vee (\Delta, \Pi)^{r_1} \sigma' \sigma_M \;\to\; \dots \;\to\; \Sigma\{\Lambda_n\}^{r_n} \sigma' \sigma_M \vee (\Delta, \Pi)^{r_n} \sigma' \sigma_M \;\to\; \Sigma\{\Lambda\}^{r'} \sigma_M \vee (\Delta, \Pi)^{r_M} . \tag{9}$$

Since $\Delta, \Sigma\{\Lambda\}, \Pi$ is properly annotated and σ_M lives on Δ, Π, it lives away from $\Sigma\{\Lambda\}$; so $\sigma_M \circ \big(r' \restriction \Sigma\{\Lambda\}\big)$ is a realization function on $\Sigma\{\Lambda\}$ by Corollary 3.13. By Facts 3.12 (4), 3.12 (7), and 3.12 (8), we conclude that

$$r := \Big(\sigma_M \circ \big(r' \restriction \Sigma\{\Lambda\}\big)\Big) \;\cup\; (r_M \restriction \Delta, \Pi)$$

is a realization function on $\Delta, \Sigma\{\Lambda\}, \Pi$. Let $\sigma := \sigma_M \circ \sigma'$. This σ lives on $\Delta, \Sigma\{\Lambda_i\}, \Pi$ for each $i = 1, \dots, n$ by Fact 3.12 (1). By Fact 3.12 (5), we have

$$\Sigma\{\Lambda\}^{r'} \sigma_M = \Sigma\{\Lambda\}^{r' \restriction \Sigma\{\Lambda\}} \sigma_M = \Sigma\{\Lambda\}^{\sigma_M \circ (r' \restriction \Sigma\{\Lambda\})} \;.$$

Therefore, we can rewrite (9) as

$$\mathsf{JL} \vdash \big(\Sigma\{\Lambda_1\} \vee (\Delta, \Pi)\big)^{r_1} \sigma \;\to\; \dots \;\to\; \big(\Sigma\{\Lambda_n\} \vee (\Delta, \Pi)\big)^{r_n} \sigma \;\to\; \big(\Sigma\{\Lambda\} \vee (\Delta, \Pi)\big)^{r} \;,$$

which, by Lemma 4.6, is propositionally equivalent to

$$\mathsf{JL} \vdash \big(\Delta, \Sigma\{\Lambda_1\}, \Pi\big)^{r_1} \sigma \;\to\; \dots \;\to\; \big(\Delta, \Sigma\{\Lambda_n\}, \Pi\big)^{r_n} \sigma \;\to\; \big(\Delta, \Sigma\{\Lambda\}, \Pi\big)^{r} \;. \qquad \square$$

Theorem 4.12 (Realization of Nested Systems). Let S be a system of nested rules whose shallow versions are realizable in a justification logic JL. Then for every sequent Γ' provable in S there exists a properly annotated version Γ of it and a realization function r on Γ such that $\mathsf{JL} \vdash \Gamma^r$.

Proof. By induction on the depth of a proof of the sequent Γ' in S. By Lemma 4.11, all rules used in this proof are realizable in JL. If Γ' is the conclusion of an instance of a 0-premise rule, the statement of the lemma follows from the fact that this rule is realizable in JL. Let $\Gamma' = \Delta'\{\Lambda'\}$ be the conclusion of an instance

$$\frac{\Delta'\{\Lambda'_1\} \quad \dots \quad \Delta'\{\Lambda'_n\}}{\Delta'\{\Lambda'\}} \tag{10}$$

of an n-premise rule ρ with common context $\Delta'\{\}$, where $n > 0$. Since ρ is realizable in JL, there exists an annotated version $\dfrac{\Delta\{\Lambda_1\} \quad \dots \quad \Delta\{\Lambda_n\}}{\Delta\{\Lambda\}}$ of the ρ-instance (10) such that for any realization functions $r_1, \dots, r_n$ on $\Delta\{\Lambda_1\}, \dots, \Delta\{\Lambda_n\}$ respectively, there exists a realization function r on $\Delta\{\Lambda\}$ and a substitution σ that lives on each of $\Delta\{\Lambda_i\}$, $i = 1, \dots, n$, such that

$$\mathsf{JL} \vdash \Delta\{\Lambda_1\}^{r_1} \sigma \;\to\; \dots \;\to\; \Delta\{\Lambda_n\}^{r_n} \sigma \;\to\; \Delta\{\Lambda\}^{r} \;. \tag{11}$$

$$
\mathsf{id}\,\frac{}{\Gamma\{P_i, \neg P_i\}} \qquad \vee\,\frac{\Gamma\{A, B\}}{\Gamma\{A \vee B\}} \qquad \wedge\,\frac{\Gamma\{A\} \quad \Gamma\{B\}}{\Gamma\{A \wedge B\}}
$$

$$
\mathsf{ctr}\,\frac{\Gamma\{A, A\}}{\Gamma\{A\}} \qquad \mathsf{exch}\,\frac{\Gamma\{\Delta, \Sigma\}}{\Gamma\{\Sigma, \Delta\}} \qquad \Box\,\frac{\Gamma\{[A]\}}{\Gamma\{\Box A\}} \qquad \mathsf{k}\,\frac{\Gamma\{[A, \Delta]\}}{\Gamma\{\Diamond A, [\Delta]\}}
$$

$$
\mathsf{d}\,\frac{\Gamma\{[A]\}}{\Gamma\{\Diamond A\}} \qquad \mathsf{t}\,\frac{\Gamma\{A\}}{\Gamma\{\Diamond A\}} \qquad \mathsf{b}\,\frac{\Gamma\{[\Delta], A\}}{\Gamma\{[\Delta, \Diamond A]\}} \qquad \mathsf{4}\,\frac{\Gamma\{[\Diamond A, \Delta]\}}{\Gamma\{\Diamond A, [\Delta]\}}
$$

$$
\mathsf{5a}\,\frac{\Gamma\{[\Delta], \Diamond A\}}{\Gamma\{[\Delta, \Diamond A]\}} \qquad \mathsf{5b}\,\frac{\Gamma\{[\Delta], [\Pi, \Diamond A]\}}{\Gamma\{[\Delta, \Diamond A], [\Pi]\}} \qquad \mathsf{5c}\,\frac{\Gamma\{[\Delta, [\Pi, \Diamond A]]\}}{\Gamma\{[\Delta, \Diamond A, [\Pi]]\}}
$$

Figure 6: Rules of nested sequent calculi

By induction hypothesis, for each $i = 1, \ldots, n$, there exists a properly annotated version $\Delta_i\{\overline{\Lambda}_i\}$ of the premise $\Delta'\{\Lambda'_i\}$ and a realization function $\overline{r}_i$ on $\Delta_i\{\overline{\Lambda}_i\}$ such that $\mathsf{JL} \vdash \Delta_i\{\overline{\Lambda}_i\}^{\overline{r}_i}$. Since $\Delta\{\Lambda_i\}$ is another properly annotated version of the same premise $\Delta'\{\Lambda'_i\}$, by Lemma 3.10, there exists a realization function r_i on $\Delta\{\Lambda_i\}$ such that $\mathsf{JL} \vdash \Delta\{\Lambda_i\}^{r_i}$.

Let r and σ be obtained from the realizability of ρ for these functions $r_1, \ldots, r_n$. By the Substitution Lemma, $\mathsf{JL} \vdash \Delta\{\Lambda_i\}^{r_i}\sigma$ for each $i = 1, \ldots, n$. It now follows from (11) by n applications of MP that $\mathsf{JL} \vdash \Delta\{\Lambda\}^r$. It remains to note that $\Delta\{\Lambda\}$ is a properly annotated version of the conclusion $\Delta'\{\Lambda'\} = \Gamma'$ of the rule instance (10). $\Box$

5. The Realization Theorem

In this section, we use Theorem 4.12 to prove a uniform realization theorem for all the modal logics: i.e., we prove that the shallow versions of the rules of various nested sequent systems for our modal logics are realizable. This leads to a series of lemmas—essentially one for each rule—of which contraction (Lemma 5.8) is the most interesting one. While there is no principal difference in the treatment of modal rules (Lemmas 5.9 and 5.15), some of the rules require extra work. In this respect, the rules that are used in logics with negative introspection have turned out to be the hardest. In order to make their presentation more readable, we separate parts of the argument into auxiliary lemmas (Lemmas 5.10–5.13 and Corollary 5.14).

Remark 5.1 (Merging and the contraction rule). It is interesting to note that while dealing with contraction (Lemma 5.8) is one of the main challenges of our method, it did not create any problems for Fitting in [16], where he applies a similar method to sequent calculi. For an advanced reader, it might be useful to ponder on the roots of such an inequality. Merging, which plays a crucial role both in Fitting's and in our method, prohibits repetitions in the annotation, forcing us to annotate the formulas being contracted in a nested sequent differently and prompting the explicit reconciliation of the annotations as detailed in Lemma 5.8. In contrast, Fitting merged things on a formula level and, thus, was able to use the same annotation for the formulas being contracted. The richer structure of nested sequents with its structural modalities, which also require merging, prevents us from using the same trick.

Remark 5.2 (Merging and the conjunction rule). Note that, whereas dealing with the shallow versions of all the logical propositional rules is equally trivial, the case of conjunction would be significantly more complicated in the actual implementation of our constructive procedure. This is due to the fact that conjunction is the only multi-premise rule, by virtue of which the use of merging in Lemma 4.11 is essential for its nested version.

D	T	KB	K4	K5	DB	D4	D5	TB	K45	S4	KB5	D45	S5
d	t	b	4	5	d, b	d, 4	d, 5	t, b	4, 5	t, 4	b, 4, 5	d, 4, 5	t, 4, 5

Figure 7: Additional rules in nested sequent systems for modal logics

Consider the inference rules in Figure 6. The *sequent system* SK consists of the rules id, ∨, ∧, ctr, exch, □, and k. It corresponds to the axiom system K. *Extensions of the system* SK are obtained by adding further rules from Figure 6 according to Figure 7, where adding 5 means that all the three rules 5a, 5b, and 5c are added. Note that a name in the first row of Figure 7 now simultaneously denotes 1) a logic, 2) an axiom system, and 3) a sequent system.

These sequent systems are essentially the same as the ones in [9], where their completeness is proved, so we have the following theorem.

Theorem 5.3 (Completeness). The system SK and its extensions are sound and complete with respect to their corresponding modal logics.

Lemma 5.4 (id-rule). The shallow version of the id-rule is realizable in J.

Proof. Since $\mathsf{J} \vdash P_i \vee \neg P_i$, the nowhere defined realization function $r := \varnothing$ suffices. □

Lemma 5.5 (∨- and exch-rules). The shallow versions of the rules ∨ and exch are realizable in J.

Proof. For an arbitrary instance $\dfrac{A', B'}{A' \vee B'}$ of sh-∨, let an annotated sequent A, B be a properly annotated version of its premise. Then $\dfrac{A, B}{A \vee B}$ is an annotated version of this instance. For any realization function r_1 on the annotated sequent A, B, let $r := r_1$ and σ be the identity substitution. Then $\underline{A, B} = A \vee B = \underline{A \vee B}$. Hence, $\left(\underline{A, B}\right)^{r_1} \sigma \to \left(\underline{A \vee B}\right)^{r}$ is a propositional tautology and, thus, is provable in J.

For an arbitrary instance $\dfrac{\Delta', \Sigma'}{\Sigma', \Delta'}$ of sh-exch, let annotated sequents Δ and Σ be annotated versions of Δ' and Σ' respectively such that the sequent Δ, Σ is a properly annotated version of the premise Δ', Σ'. Then $\dfrac{\Delta, \Sigma}{\Sigma, \Delta}$ is an annotated version of this instance. For any realization function r_1 on Δ, Σ, let $r := r_1$ and σ be the identity substitution. Then $\mathsf{J} \vdash (\Delta, \Sigma)^{r_1} \sigma \to (\Sigma, \Delta)^{r}$ follows from Lemma 4.6. □

The realizability for the □-rule is trivial:

Lemma 5.6 (□-rule). The shallow version of the □-rule is realizable in J.

Lemma 5.7 (∧-rule). The shallow version of the ∧-rule is realizable in J.

Proof. For an arbitrary instance $\dfrac{A' \quad B'}{A' \wedge B'}$ of sh-∧, let an annotated sequent $A \wedge B$ be a properly annotated version of its conclusion. Then $\dfrac{A \quad B}{A \wedge B}$ is an annotated version of this instance since A and B do not share indices. For arbitrary realization functions r_1 and r_2 on the annotated sequents A and B respectively, let $r := (r_1 \restriction A) \cup (r_2 \restriction B)$ and σ be the identity substitution. The former is a realization function on $A \wedge B$ by Facts 3.12 (6) and 3.12 (8). Finally, $A^{r_1}\sigma \to B^{r_2}\sigma \to (A \wedge B)^{r}$ is a propositional tautology and, thus, is provable in J since $(A \wedge B)^{r} = A^{r_1}\sigma \wedge B^{r_2}\sigma$. □

Lemma 5.8 (ctr-rule). The shallow version of the ctr-rule is realizable in J.

Proof. For an arbitrary instance $\dfrac{A', A'}{A'}$ of $\mathsf{sh\text{-}ctr}$, let annotated sequents A_1, A_2 and A_3 not share indices and be properly annotated versions of its premise and conclusion respectively. Then $\dfrac{A_1, A_2}{A_3}$ is an annotated version of this instance. Let r_1 be a realization function on A_1, A_2. Let B_3 be a subformula occurrence of A_3 and let B_1 and B_2 denote the subformula occurrences in A_1 and A_2 respectively that correspond to B_3 in A_3. By induction on the structure of B_3, we construct a realization function r on B_3 and a substitution σ with $\mathrm{vrange}(\sigma) \subseteq \mathrm{vars}_\Diamond(B_3)$ that lives on $B_1 \vee B_2$ such that

$$(B_1 \vee B_2)^{r_1}\sigma \ \rightarrow \ (B_3)^r \tag{12}$$

is provable in J. Recall that A_1, A_2, and A_3 are all annotated versions of A' and, hence, have the "same" structure. Note also that r_1 is clearly a realization function on $B_1 \vee B_2$ for any subformula occurrence B_3 of A_3.

Base case: $B_3 = P_i$ or $B_3 = \neg P_i$. In this case, $B_1 = B_2 = B_3$ and, independent of σ and r, (12) can be rewritten as $B_3 \vee B_3 \rightarrow B_3$, a propositional tautology provable in J. Hence, one can take σ to be the identity substitution and $r := \varnothing$.

To prove the **induction step**, the following cases have to be considered:

Case $B_3 = D_3 \vee C_3$. Then $B_1 = D_1 \vee C_1$ and $B_2 = D_2 \vee C_2$. By induction hypothesis, there exist realization functions r'_D and r'_C on D_3 and C_3 respectively, as well as substitutions σ'_D and σ'_C with $\mathrm{vrange}(\sigma'_D) \subseteq \mathrm{vars}_\Diamond(D_3)$ and $\mathrm{vrange}(\sigma'_C) \subseteq \mathrm{vars}_\Diamond(C_3)$ that live on $D_1 \vee D_2$ and $C_1 \vee C_2$ respectively, such that

$$\mathsf{J} \vdash (D_1 \vee D_2)^{r_1}\sigma'_D \ \rightarrow \ (D_3)^{r'_D} \qquad \text{and} \qquad \mathsf{J} \vdash (C_1 \vee C_2)^{r_1}\sigma'_C \ \rightarrow \ (C_3)^{r'_C} \ .$$

By the Substitution Lemma,

$$\mathsf{J} \vdash (D_1 \vee D_2)^{r_1}\sigma'_D\sigma'_C \ \rightarrow \ (D_3)^{r'_D}\sigma'_C \qquad \text{and} \qquad \mathsf{J} \vdash (C_1 \vee C_2)^{r_1}\sigma'_C\sigma'_D \ \rightarrow \ (C_3)^{r'_C}\sigma'_D \ . \tag{13}$$

Since C_1 and D_1, C_2 and D_2, and C_3 and D_3 are subformulas of A_1, A_2, and A_3 respectively, the latter three pairwise sharing no indices, it follows that $\mathrm{dom}(\sigma'_C) \subseteq \mathrm{vars}_\Diamond(C_1 \vee C_2)$ is disjoint from $\mathrm{vrange}(\sigma'_D) \subseteq \mathrm{vars}_\Diamond(D_3)$. Further, $\mathrm{dom}(\sigma'_C)$ is also disjoint from $\mathrm{dom}(\sigma'_D) \subseteq \mathrm{vars}_\Diamond(D_1 \vee D_2)$ because, in addition, $D_1 \vee C_1$ and $D_2 \vee C_2$ are properly annotated. It follows from Fact 3.12 (3) that $\sigma'_D \cup \sigma'_C = \sigma'_C \circ \sigma'_D$. Let $\sigma := \sigma'_D \cup \sigma'_C$. Then $(D_1 \vee D_2)^{r_1}\sigma'_D\sigma'_C = (D_1 \vee D_2)^{r_1}\sigma$ and σ lives on $B_1 \vee B_2$ by Fact 3.12 (2). It can be similarly shown that $\mathrm{vrange}(\sigma'_C) \subseteq \mathrm{vars}_\Diamond(C_3)$ is disjoint from $\mathrm{dom}(\sigma'_D)$ and, hence, $\sigma = \sigma'_D \circ \sigma'_C$, so that $(C_1 \vee C_2)^{r_1}\sigma'_C\sigma'_D = (C_1 \vee C_2)^{r_1}\sigma$. By Fact 3.12 (1), $\mathrm{vrange}(\sigma) \subseteq \mathrm{vars}_\Diamond(D_3) \cup \mathrm{vars}_\Diamond(C_3) = \mathrm{vars}_\Diamond(B_3)$. So σ is a suitable substitution and (13) can be rewritten as

$$\mathsf{J} \vdash (D_1 \vee D_2)^{r_1}\sigma \ \rightarrow \ (D_3)^{r'_D}\sigma'_C \qquad \text{and} \qquad \mathsf{J} \vdash (C_1 \vee C_2)^{r_1}\sigma \ \rightarrow \ (C_3)^{r'_C}\sigma'_D \ . \tag{14}$$

Since σ'_C and σ'_D live away from D_3 and C_3 respectively, by Corollary 3.13, both $r_D := \sigma'_C \circ (r'_D \restriction D_3)$ and $r_C := \sigma'_D \circ (r'_C \restriction C_3)$ are realization functions on D_3 and C_3 respectively. By Fact 3.12 (5), we have $(D_3)^{r'_D}\sigma'_C = (D_3)^{r_D}$ and $(C_3)^{r'_C}\sigma'_D = (C_3)^{r_C}$. Now (14) can be rewritten as

$$\mathsf{J} \vdash (D_1 \vee D_2)^{r_1}\sigma \ \rightarrow \ (D_3)^{r_D} \qquad \text{and} \qquad \mathsf{J} \vdash (C_1 \vee C_2)^{r_1}\sigma \ \rightarrow \ (C_3)^{r_C} \ .$$

Finally, by propositional reasoning, it is provable in J that

$$\big((D_1 \vee C_1) \vee (D_2 \vee C_2)\big)^{r_1}\sigma \ \rightarrow \ (D_3)^{r_D} \vee (C_3)^{r_C} \ ,$$

which is exactly (12) for $r := r_D \cup r_C$. It is easy to see, using Fact 3.12, that r is a realization function on the properly annotated formula $B_3 = D_3 \vee C_3$.

Case $B_3 = D_3 \wedge C_3$ is analogous to $B_3 = D_3 \vee C_3$.

Case $B_3 = \Diamond_{2n} C_3$. Then $B_1 = \Diamond_{2k} C_1$ and $B_2 = \Diamond_{2m} C_2$. By induction hypothesis, there exists a realization function r' on C_3 and a substitution σ' with $\mathrm{vrange}(\sigma') \subseteq \mathrm{vars}_\Diamond(C_3)$ that lives on $C_1 \vee C_2$ such that $\mathsf{J} \vdash (C_1 \vee C_2)^{r_1}\sigma' \to (C_3)^{r'}$. By propositional reasoning,

$$\mathsf{J} \vdash \neg(C_3)^{r'} \to \neg(C_1)^{r_1}\sigma' \qquad \text{and} \qquad \mathsf{J} \vdash \neg(C_3)^{r'} \to \neg(C_2)^{r_1}\sigma' \ .$$

By Internalization Property 3.5, there exist terms $t_1(x_1)$ and $t_2(x_1)$ such that

$$\mathsf{J} \vdash x_n : \neg(C_3)^{r'} \to t_1(x_n) : \big(\neg(C_1)^{r_1}\sigma'\big) \quad \text{and} \quad \mathsf{J} \vdash x_n : \neg(C_3)^{r'} \to t_2(x_n) : \big(\neg(C_2)^{r_1}\sigma'\big) \ .$$

It then follows by propositional reasoning that

$$\mathsf{J} \vdash \neg t_1(x_n) : \big(\neg(C_1)^{r_1}\sigma'\big) \vee \neg t_2(x_n) : \big(\neg(C_2)^{r_1}\sigma'\big) \to \neg x_n : \neg(C_3)^{r'} \ . \tag{15}$$

Since $\mathrm{dom}(\sigma') \subseteq \mathrm{vars}_\Diamond(C_1 \vee C_2) \not\ni x_n$ (indeed, $\Diamond_{2n}$ occurs in B_3, which shares indices with neither B_1 nor B_2), the substitution σ' affects neither $t_1(x_n)$ nor $t_2(x_n)$ because they contain no variables other than x_n. As a consequence, (15) can be rewritten as

$$\mathsf{J} \vdash \big(\neg t_1(x_n) : \neg(C_1)^{r_1} \vee \neg t_2(x_n) : \neg(C_2)^{r_1}\big)\sigma' \to \neg x_n : \neg(C_3)^{r'} \ .$$

Let $\sigma'' := \{x_k \mapsto t_1(x_n); x_m \mapsto t_2(x_n)\} \cup \{x_i \mapsto x_i \mid i \notin \{k, m\}\}$. By the Substitution Lemma and since $x_n \notin \{x_k, x_m\}$,

$$\mathsf{J} \vdash \big(\neg t_1(x_n) : \neg(C_1)^{r_1} \vee \neg t_2(x_n) : \neg(C_2)^{r_1}\big)\sigma'\sigma'' \to \neg x_n : \neg\big((C_3)^{r'}\sigma''\big) \ . \tag{16}$$

Since σ'' lives away from C_3 (indeed, $\Diamond_{2k}$ and $\Diamond_{2m}$ occur in B_1 and B_2 respectively, neither of which shares indices with B_3), we know by Corollary 3.13 that $\sigma'' \circ (r' \upharpoonright C_3)$ is a realization function on C_3. In addition, $(C_3)^{r'}\sigma'' = C_3^{\sigma'' \circ (r' \upharpoonright C_3)}$. Therefore, (16) can be rewritten as

$$\mathsf{J} \vdash \big(\neg t_1(x_n) : \neg(C_1)^{r_1} \vee \neg t_2(x_n) : \neg(C_2)^{r_1}\big)\sigma'\sigma'' \to \neg x_n : \neg(C_3)^{\sigma'' \circ (r' \upharpoonright C_3)} \ . \tag{17}$$

Let $\sigma := \sigma'' \circ \sigma'$ and $r := \big(\sigma'' \circ (r' \upharpoonright C_3)\big) \cup \{2n \mapsto x_n\}$. Clearly, r is a realization function on B_3. Since σ' affects none of x_k, x_m, $t_1(x_n)$, or $t_2(x_n)$, (17) can be rewritten to state the provability in J of $(\Diamond_{2k} C_1 \vee \Diamond_{2m} C_2)^{r_1}\sigma \to (\Diamond_{2n} C_3)^r$, which is exactly (12). It remains to note that, by Fact 3.12 (1),

$$\mathrm{dom}(\sigma) \subseteq \mathrm{dom}(\sigma') \cup \mathrm{dom}(\sigma'') \subseteq \mathrm{vars}_\Diamond(C_1 \vee C_2) \cup \{x_k, x_m\} = \mathrm{vars}_\Diamond(\Diamond_{2k} C_1 \vee \Diamond_{2m} C_2)$$

and also $\mathrm{vrange}(\sigma) \subseteq \mathrm{vrange}(\sigma') \cup \mathrm{vrange}(\sigma'') \subseteq \mathrm{vars}_\Diamond(C_3) \cup \{x_n\} = \mathrm{vars}_\Diamond(\Diamond_{2n} C_3)$.

Case $B_3 = \Box_m C_3$. Then $B_1 = \Box_k C_1$ and $B_2 = \Box_l C_2$. By induction hypothesis, there exists a realization function r' on C_3 and a substitution σ' with $\mathrm{vrange}(\sigma') \subseteq \mathrm{vars}_\Diamond(C_3)$ that lives on $C_1 \vee C_2$ such that $\mathsf{J} \vdash (C_1 \vee C_2)^{r_1}\sigma' \to (C_3)^{r'}$. By propositional reasoning and Internalization Property 3.5, there exist terms $t_1(x_1)$ and $t_2(x_1)$ such that

$$\mathsf{J} \vdash r_1(k)\sigma' : \big((C_1)^{r_1}\sigma'\big) \to t_1\big(r_1(k)\sigma'\big) : (C_3)^{r'} \qquad \text{and} \qquad \mathsf{J} \vdash r_1(l)\sigma' : \big((C_2)^{r_1}\sigma'\big) \to t_2\big(r_1(l)\sigma'\big) : (C_3)^{r'} \ .$$

By the axiom sum, for $s := t_1\big(r_1(k)\sigma'\big) + t_2\big(r_1(l)\sigma'\big)$,

$$\mathsf{J} \vdash r_1(k)\sigma' : \big((C_1)^{r_1}\sigma'\big) \to s : (C_3)^{r'} \qquad \text{and} \qquad \mathsf{J} \vdash r_1(l)\sigma' : \big((C_2)^{r_1}\sigma'\big) \to s : (C_3)^{r'} \ .$$

Thus, by propositional reasoning,

$$\mathsf{J} \vdash \big(r_1(k) : (C_1)^{r_1} \vee r_1(l) : (C_2)^{r_1}\big)\sigma' \to s : (C_3)^{r'} \ . \tag{18}$$

Let $\sigma := \sigma'$ and $r := (r' \upharpoonright C_3) \cup \{m \mapsto s\}$. Clearly, r is a realization function on B_3, σ lives on $C_1 \vee C_2$, or equivalently on $B_1 \vee B_2$, and $\mathrm{vrange}(\sigma) \subseteq \mathrm{vars}_\Diamond(C_3) = \mathrm{vars}_\Diamond(B_3)$. Now (18) can be rewritten to state the provability in J of $(\Box_k C_1 \vee \Box_l C_2)^{r_1}\sigma \to (\Box_m C_3)^r$, which is exactly (12).

It remains to note that (12) for $B_3 = A_3$ and for thus constructed r and σ is $\big(\underline{A_1, A_2}\big)^{r_1}\sigma \to \big(\underline{A_3}\big)^r$. $\Box$

Lemma 5.9 (k-rule). The shallow version of the k-rule is realizable in J.

Proof. For an arbitrary instance $\dfrac{[A', \Delta']}{\Diamond A', [\Delta']}$ of sh-k, let $[A, \Delta]_k$ and $\Diamond_{2m} A, [\Delta]_i$ be properly annotated versions of its premise and conclusion respectively. Then $\dfrac{[A, \Delta]_k}{\Diamond_{2m} A, [\Delta]_i}$ is an annotated version of this instance. Let r_1 be an arbitrary realization function on $[A, \Delta]_k$. Consider the propositional tautology $(A, \Delta)^{r_1} \to \neg A^{r_1} \to \Delta^{r_1}$. By Internalization Property 3.5, there exists a term $t(x_1, x_2)$ such that

$$\mathsf{J} \vdash r_1(k) : (A, \Delta)^{r_1} \to\ x_m : \neg A^{r_1} \to\ t\big(r_1(k), x_m\big) : \Delta^{r_1}\ \ .$$

It follows by propositional reasoning that

$$\mathsf{J} \vdash r_1(k) : (A, \Delta)^{r_1} \to\ \neg x_m : \neg A^{r_1} \vee\ t\big(r_1(k), x_m\big) : \Delta^{r_1}\ \ . \tag{19}$$

The indices $2m$ and i cannot occur in either A or Δ because $\Diamond_{2m} A, [\Delta]_i$ is properly annotated. Hence,

$$r := (r_1 \restriction A, \Delta)\ \cup\ \big\{2m \mapsto x_m; i \mapsto t\big(r_1(k), x_m\big)\big\}$$

is a realization function on $\Diamond_{2m} A, [\Delta]_i$. For the identity substitution σ and this r, (19) can be rewritten as

$$\mathsf{J} \vdash \big([A, \Delta]_k\big)^{r_1}\sigma\ \to\ \big(\Diamond_{2m} A, [\Delta]_i\big)^{r}\ \ . \qquad \Box$$

Lemma 5.15 covers the remaining rules from Figure 6. The following auxiliary lemmas are used for the part of Lemma 5.15 that concerns the rules 5a, 5b, and 5c. The following lemma provides a uniform realization for the theorem $\Box(A \to B) \to \Box(B \to C) \to \Box(A \to C)$ of K.

Lemma 5.10 (Syllogism). There exists a term $\mathrm{syl}(x_1, x_2)$ such that for arbitrary terms t_1 and t_2 and for arbitrary justification formulas A, B, and C,

$$\mathsf{J} \vdash t_1 : (A \to B)\ \to\ t_2 : (B \to C)\ \to\ \mathrm{syl}(t_1, t_2) : (A \to C)\ \ .$$

Proof. From the propositional tautology $(P_1 \to P_2) \to (P_2 \to P_3) \to (P_1 \to P_3)$, by Internalization Property 3.5, there exists a term $\mathrm{syl}(x_1, x_2)$ such that for arbitrary terms t_1 and t_2,

$$\mathsf{J} \vdash t_1 : (P_1 \to P_2)\ \to\ t_2 : (P_2 \to P_3)\ \to\ \mathrm{syl}(t_1, t_2) : (P_1 \to P_3)\ \ .$$

The desired result now follows from the Substitution Lemma. Note that $\mathrm{syl}(x_1, x_2)$ does not depend on t_1, t_2, A, B, or C. $\Box$

Lemma 5.11 (Internalized Factivity). There exists a term $\mathrm{fact}(x_1)$ such that for any term s and any justification formula A,

$$\mathsf{J5} \vdash \mathrm{fact}(s) : (s : A \to A)\ \ .$$

Proof. From the propositional tautology $P_1 \to P_2 \to P_1$, by Internalization Property 3.5, there exists a term $t_1(x_1)$ such that $\mathsf{J5} \vdash s : P_1 \to t_1(s) : (P_2 \to P_1)$ for any term s. Hence, by the Substitution Lemma, for any formula A,

$$\mathsf{J5} \vdash s : A\ \to\ t_1(s) : (s : A \to A)\ \ . \tag{20}$$

Similarly, for $\neg P_2 \to P_2 \to P_1$, there exists $t_2(x_1)$ such that $\mathsf{J5} \vdash ?\,s : \neg P_2 \to t_2(?\,s) : (P_2 \to P_1)$ for any term s. By the Substitution Lemma,

$$\mathsf{J5} \vdash ?\,s : \neg s : A\ \to\ t_2(?\,s) : (s : A \to A)$$

for any formula A. Since $\neg s : A \to ?\,s : \neg s : A$ is a j5-instance, it follows that

$$\mathsf{J5} \vdash \neg s : A\ \to\ t_2(?\,s) : (s : A \to A)\ \ .$$

From this, (20), and sum, we have $\mathsf{J5} \vdash \mathrm{fact}(s) : (s : A \to A)$ for $\mathrm{fact}(x_1) := t_1(x_1) + t_2(?\,x_1)$. Note that $\mathrm{fact}(x_1)$ depends on neither s nor A. $\Box$

The following auxiliary lemma is used in the proofs of Lemmas 5.13 and 6.17.

Lemma 5.12 (Inverse to Negative Introspection, Internalized). There exists a term $\mathrm{invnegint}(x_1)$ such that for arbitrary terms t and s and for any justification formula A,

$$\mathsf{J5} \vdash s : \neg ?t : \neg t : A \;\rightarrow\; \mathrm{invnegint}(s) : t : A \;\;.$$

Proof. It follows from propositional reasoning and Internalization Property 3.5 that there exists a ground term p such that

$$\mathsf{J5} \vdash p : \big((\neg x_2 : P_1 \rightarrow ?x_2 : \neg x_2 : P_1) \;\rightarrow\; \neg ?x_2 : \neg x_2 : P_1 \;\rightarrow\; x_2 : P_1\big) \;\;. \tag{21}$$

For a fixed arbitrary constant c^1_j of level 1, $\mathsf{J5} \vdash c^1_j : \big(\neg x_2 : P_1 \rightarrow ?x_2 : \neg x_2 : P_1\big)$ by $\mathsf{j5}$ and iAN. From this and (21), by app and MP,

$$\mathsf{J5} \vdash (p \cdot c^1_j) : (\neg ?x_2 : \neg x_2 : P_1 \;\rightarrow\; x_2 : P_1) \;\;.$$

Also by app and MP,

$$\mathsf{J5} \vdash x_1 : \neg ?x_2 : \neg x_2 : P_1 \;\rightarrow\; (p \cdot c^1_j \cdot x_1) : x_2 : P_1 \;\;.$$

The statement of the lemma for $\mathrm{invnegint}(x_1) := p \cdot c^1_j \cdot x_1$ now follows from the Substitution Lemma. Note that $\mathrm{invnegint}(x_1)$ does not depend on t, s, or A. □

Lemma 5.13 (Internalized Positive Introspection). There exist terms $\mathrm{posint}(x_1)$ and $\mathrm{t}_!(x_1)$ such that for any term s and any justification formula A,

$$\mathsf{J5} \vdash \mathrm{posint}(s) : \big(s : A \;\rightarrow\; \mathrm{t}_!(s) : s : A\big) \;\;.$$

Proof. We first show that there exists a term $s(x_1)$ such that for any t and A,

$$\mathsf{J5} \vdash s(t) : (A \;\rightarrow\; ?t : \neg t : \neg A) \;\;. \tag{22}$$

It follows from propositional reasoning and Internalization Property 3.5 that there exists a ground term p such that

$$\mathsf{J5} \vdash p : \big((x_1 : \neg P_1 \rightarrow \neg P_1) \;\rightarrow\; P_1 \;\rightarrow\; \neg x_1 : \neg P_1\big) \;\;.$$

By Lemma 5.11, for the term $\mathrm{fact}(x_1)$ constructed there, $\mathsf{J5} \vdash \mathrm{fact}(x_1) : (x_1 : \neg P_1 \rightarrow \neg P_1)$. By app and MP,

$$\mathsf{J5} \vdash \big(p \cdot \mathrm{fact}(x_1)\big) : (P_1 \;\rightarrow\; \neg x_1 : \neg P_1) \;\;.$$

For a fixed arbitrary constant c^1_i of level 1, $c^1_i : (\neg x_1 : \neg P_1 \rightarrow ?x_1 : \neg x_1 : \neg P_1)$ is provable in $\mathsf{J5}$ by $\mathsf{j5}$ and iAN. Hence, by Lemma 5.10,

$$\mathsf{J5} \vdash \mathrm{syl}\big(p \cdot \mathrm{fact}(x_1), c^1_i\big) : (P_1 \;\rightarrow\; ?x_1 : \neg x_1 : \neg P_1) \;\;.$$

Now (22) follows from the Substitution Lemma for $s(x_1) := \mathrm{syl}\big(p \cdot \mathrm{fact}(x_1), c^1_i\big)$.

By Lemma 5.12, for the term $\mathrm{invnegint}(x_1)$ constructed there,

$$\mathsf{J5} \vdash ??x_1 : \neg ?x_1 : \neg x_1 : P_1 \;\rightarrow\; \mathrm{invnegint}(??x_1) : x_1 : P_1 \;\;.$$

Then, by Internalization Property 3.5, there exists a ground term p_2 such that

$$\mathsf{J5} \vdash p_2 : \big(??x_1 : \neg ?x_1 : \neg x_1 : P_1 \;\rightarrow\; \mathrm{invnegint}(??x_1) : x_1 : P_1\big) \;\;.$$

By (22), for $t = ?x_1$ and $A = x_1 : P_1$,

$$\mathsf{J5} \vdash s(?x_1) : (x_1 : P_1 \;\rightarrow\; ??x_1 : \neg ?x_1 : \neg x_1 : P_1) \;\;.$$

Hence, by Lemma 5.10,

$$\mathsf{J5} \vdash \mathrm{syl}\big(s(?x_1), p_2\big) : \big(x_1 : P_1 \;\rightarrow\; \mathrm{invnegint}(??x_1) : x_1 : P_1\big) \;\;.$$

For $\mathrm{posint}(x_1) := \mathrm{syl}\big(s(?x_1), p_2\big)$ and $\mathrm{t}_!(x_1) := \mathrm{invnegint}(??x_1)$, the statement of the lemma now follows by the Substitution Lemma. Note that $\mathrm{t}_!(x_1)$ and $\mathrm{posint}(x_1)$ depend on neither s nor A. □

Corollary 5.14 (Internalized Inverse Positive Introspection). There exists a term $\mathrm{invposint}(x_1)$ such that for any term s, for any formula A, and for the term $\mathrm{t}_!(x_1)$ constructed in Lemma 5.13,

$$\mathsf{J5} \vdash \mathrm{invposint}(s) : \bigl(\neg\, \mathrm{t}_!(s) : s : A \;\to\; \neg s : A\bigr) \enspace .$$

Proof. By Lemma 5.13, for the terms $\mathrm{posint}(x_1)$ and $\mathrm{t}_!(x_1)$ constructed there,

$$\mathsf{J5} \vdash \mathrm{posint}(x_1) : \bigl(x_1 : P_1 \;\to\; \mathrm{t}_!(x_1) : x_1 : P_1\bigr) \enspace .$$

By propositional reasoning and Internalization Property 3.5, there exists a ground term p such that

$$\mathsf{J5} \vdash p : \Bigl(\bigl(x_1 : P_1 \to \mathrm{t}_!(x_1) : x_1 : P_1\bigr) \;\to\; \neg\, \mathrm{t}_!(x_1) : x_1 : P_1 \;\to\; \neg x_1 : P_1\Bigr) \enspace .$$

For $\mathrm{invposint}(x_1) := p \cdot \mathrm{posint}(x_1)$, by app and MP,

$$\mathsf{J5} \vdash \mathrm{invposint}(x_1) : \bigl(\neg\, \mathrm{t}_!(x_1) : x_1 : P_1 \;\to\; \neg x_1 : P_1\bigr) \enspace .$$

The statement of the lemma now follows from the Substitution Lemma. Note that $\mathrm{invposint}(x_1)$ depends on neither s nor A. □

Lemma 5.15 (Modal Rules). Let $\rho \in \{\mathsf{d}, \mathsf{t}, \mathsf{b}, \mathsf{4}, \mathsf{5a}, \mathsf{5b}, \mathsf{5c}\}$. The shallow version of ρ is realizable in $\mathsf{J}\rho$, where by Jd we mean JD, and so on, except for $\rho \in \{\mathsf{5a}, \mathsf{5b}, \mathsf{5c}\}$, where we mean $\mathsf{J5}$.

Proof. We consider an arbitrary instance of $\mathsf{sh}\text{-}\rho$ for each rule ρ in turn.

Case $\rho = \mathsf{d}$. For an arbitrary instance $\dfrac{[A']}{\Diamond A'}$ of $\mathsf{sh}\text{-}\mathsf{d}$, let $[A]_k$ and $\Diamond_{2m} A$ be properly annotated versions of its premise and conclusion respectively. Then $\dfrac{[A]_k}{\Diamond_{2m} A}$ is an annotated version of this instance. Consider an arbitrary realization function r_1 on $[A]_k$. From the app-instance

$$x_m : (A^{r_1} \to \bot) \;\to\; r_1(k) : A^{r_1} \to\; \bigl(x_m \cdot r_1(k)\bigr) : \bot \enspace ,$$

it follows by propositional reasoning that

$$\mathsf{JD} \vdash r_1(k) : A^{r_1} \to\; x_m : (A^{r_1} \to \bot) \;\to\; \bigl(x_m \cdot r_1(k)\bigr) : \bot \enspace .$$

Using the jd-instance $\bigl(x_m \cdot r_1(k)\bigr) : \bot \to \bot$, we obtain by propositional reasoning

$$\mathsf{JD} \vdash r_1(k) : A^{r_1} \to\; x_m : (A^{r_1} \to \bot) \;\to\; \bot \enspace ,$$

which is identical to $\mathsf{JD} \vdash r_1(k) : A^{r_1} \to \neg x_m : \neg A^{r_1}$. Since $2m$ is even, $r := r_1 \cup \{2m \mapsto x_m\}$ is a realization function on $\Diamond_{2m} A$ by Facts 3.12 (7) and 3.12 (8). Thus, for the identity substitution σ and this r,

$$\mathsf{JD} \vdash \bigl([A]_k\bigr)^{r_1} \sigma \;\to\; (\Diamond_{2m} A)^r \enspace .$$

Case $\rho = \mathsf{t}$. For an arbitrary instance $\dfrac{A'}{\Diamond A'}$ of $\mathsf{sh}\text{-}\mathsf{t}$, let $\Diamond_{2m} A$ be a properly annotated version of its conclusion. Then $\dfrac{A}{\Diamond_{2m} A}$ is an annotated version of this instance. Consider an arbitrary realization function r_1 on A. By taking the contraposition of the jt-instance $x_m {:} \neg A^{r_1} \to \neg A^{r_1}$, we have $\mathsf{JT} \vdash A^{r_1} \to \neg x_m {:} \neg A^{r_1}$. Since $2m$ is even, $r := r_1 \cup \{2m \mapsto x_m\}$ is a realization function on $\Diamond_{2m} A$. Thus, for the identity substitution σ and this r,

$$\mathsf{JT} \vdash A^{r_1} \sigma \;\to\; (\Diamond_{2m} A)^r \enspace .$$

Case $\rho = \mathsf{b}$. For an arbitrary instance $\dfrac{[\Delta'], A'}{[\Delta', \Diamond A']}$ of $\mathsf{sh\text{-}b}$, let $[\Delta]_k, A$ and $[\Delta, \Diamond_{2m} A]_i$ be properly annotated versions of its premise and conclusion respectively. Then $\dfrac{[\Delta]_k, A}{[\Delta, \Diamond_{2m} A]_i}$ is an annotated version of this instance. Consider an arbitrary realization function r_1 on $[\Delta]_k, A$. Since $\Delta^{r_1} \to \Delta^{r_1} \vee \neg x_m : \neg A^{r_1}$ is a propositional tautology, by Internalization Property 3.5, there exists a term $t_1(x_1)$ such that

$$\mathsf{JB} \vdash r_1(k) : \Delta^{r_1} \to \ t_1\big(r_1(k)\big) : (\Delta^{r_1} \vee \ \neg x_m : \neg A^{r_1}) \ . \tag{23}$$

Similarly, for $\neg x_m : \neg A^{r_1} \to \Delta^{r_1} \vee \neg x_m : \neg A^{r_1}$, there exists a term $t_2(x_1)$ such that

$$\mathsf{JB} \vdash \bar{?}\, x_m : \neg x_m : \neg A^{r_1} \to \ t_2(\bar{?}\, x_m) : (\Delta^{r_1} \vee \ \neg x_m : \neg A^{r_1}) \ .$$

From this and (23), it follows by the axiom sum and propositional reasoning that

$$\mathsf{JB} \vdash r_1(k) : \Delta^{r_1} \vee \ \bar{?}\, x_m : \neg x_m : \neg A^{r_1} \to \ t : (\Delta^{r_1} \vee \ \neg x_m : \neg A^{r_1})$$

for $t := t_1\big(r_1(k)\big) + t_2(\bar{?}\, x_m)$. Finally, from the jb-instance $A^{r_1} \to \bar{?}\, x_m : \neg x_m : \neg A^{r_1}$, it follows that

$$\mathsf{JB} \vdash r_1(k) : \Delta^{r_1} \vee \ A^{r_1} \to \ t : (\Delta^{r_1} \vee \ \neg x_m : \neg A^{r_1}) \ .$$

The indices $2m$ and i do not occur in either Δ or A because $[\Delta, \Diamond_{2m} A]_i$ is properly annotated. Hence,

$$r := (r_1 \upharpoonright \Delta, A) \ \cup \ \{i \mapsto t; 2m \mapsto x_m\}$$

is a realization function on $[\Delta, \Diamond_{2m} A]_i$. Thus, for the identity substitution σ and this r,

$$\mathsf{JB} \vdash \big([\Delta]_k, A\big)^{r_1} \sigma \ \to \ \big([\Delta, \Diamond_{2m} A]_i\big)^{r} \ .$$

Case $\rho = \mathsf{4}$. For an arbitrary instance $\dfrac{[\Diamond A', \Delta']}{\Diamond A', [\Delta']}$ of $\mathsf{sh\text{-}4}$, let $[\Diamond_{2m} A, \Delta]_k$ and $\Diamond_{2m} A, [\Delta]_i$ be properly annotated versions of its premise and conclusion respectively. Then $\dfrac{[\Diamond_{2m} A, \Delta]_k}{\Diamond_{2m} A, [\Delta]_i}$ is an annotated version of this instance. Consider an arbitrary realization function r_1 on $[\Diamond_{2m} A, \Delta]_k$. Since

$$x_m : \neg A^{r_1} \to \ \neg x_m : \neg A^{r_1} \vee \ \Delta^{r_1} \to \ \Delta^{r_1}$$

is a propositional tautology, it follows from Internalization Property 3.5 that there is a term $s(x_1)$ such that

$$\mathsf{J4} \vdash !\, x_m : x_m : \neg A^{r_1} \to \ s(!\, x_m) : (\neg x_m : \neg A^{r_1} \vee \ \Delta^{r_1} \to \ \Delta^{r_1}) \ .$$

From the $\mathsf{j4}$-instance $x_m : \neg A^{r_1} \to !\, x_m : x_m : \neg A^{r_1}$, it then follows by propositional reasoning that

$$\mathsf{J4} \vdash x_m : \neg A^{r_1} \to \ s(!\, x_m) : (\neg x_m : \neg A^{r_1} \vee \ \Delta^{r_1} \to \ \Delta^{r_1}) \ .$$

By the axiom app and propositional reasoning,

$$\mathsf{J4} \vdash r_1(k) : (\neg x_m : \neg A^{r_1} \vee \ \Delta^{r_1}) \ \to \ \neg x_m : \neg A^{r_1} \vee \ \big(s(!\, x_m) \cdot r_1(k)\big) : \Delta^{r_1} \ .$$

The index i does not occur in either Δ or $\Diamond_{2m} A$ because $\Diamond_{2m} A, [\Delta]_i$ is properly annotated. Hence,

$$r := (r_1 \upharpoonright \Diamond_{2m} A, \Delta) \ \cup \ \{i \mapsto s(!\, x_m) \cdot r_1(k)\}$$

is a realization function on $\Diamond_{2m} A, [\Delta]_i$. Thus, for the identity substitution σ and this r,

$$\mathsf{J4} \vdash \big([\Diamond_{2m} A, \Delta]_k\big)^{r_1} \sigma \ \to \ \big(\Diamond_{2m} A, [\Delta]_i\big)^{r} \ .$$

Case $\rho = \mathsf{5a}$. For an arbitrary instance $\dfrac{[\Delta'], \Diamond A'}{[\Delta', \Diamond A']}$ of $\mathsf{sh\text{-}5a}$, let $[\Delta]_k, \Diamond_{2m} A$ and $[\Delta, \Diamond_{2m} A]_i$ be properly annotated versions of its premise and conclusion respectively. Then $\dfrac{[\Delta]_k, \Diamond_{2m} A}{[\Delta, \Diamond_{2m} A]_i}$ is an annotated version of this instance. Consider an arbitrary realization function r_1 on $[\Delta]_k, \Diamond_{2m} A$. By the propositional tautology $\Delta^{r_1} \to \Delta^{r_1} \vee \neg x_m : \neg A^{r_1}$ and Internalization Property 3.5, there exists a term $t_1(x_1)$ such that

$$\mathsf{J5} \vdash r_1(k) : \Delta^{r_1} \to\ t_1\bigl(r_1(k)\bigr) : (\Delta^{r_1} \vee\ \neg x_m : \neg A^{r_1})\ . \tag{24}$$

Similarly, for the propositional tautology $\neg x_m : \neg A^{r_1} \to \Delta^{r_1} \vee \neg x_m : \neg A^{r_1}$, there exists $t_2(x_1)$ such that

$$\mathsf{J5} \vdash ?\, x_m : \neg x_m : \neg A^{r_1} \to\ t_2(?\, x_m) : (\Delta^{r_1} \vee\ \neg x_m : \neg A^{r_1})\ .$$

From the $\mathsf{j5}$-instance $\neg x_m : \neg A^{r_1} \to ?\, x_m : \neg x_m : \neg A^{r_1}$, by propositional reasoning,

$$\mathsf{J5} \vdash \neg x_m : \neg A^{r_1} \to\ t_2(?\, x_m) : (\Delta^{r_1} \vee\ \neg x_m : \neg A^{r_1})\ . \tag{25}$$

It follows from (24) and (25) by the axiom sum and propositional reasoning that for $t := t_1\bigl(r_1(k)\bigr) + t_2(?\, x_m)$,

$$\mathsf{J5} \vdash r_1(k) : \Delta^{r_1} \vee\ \neg x_m : \neg A^{r_1} \to\ t : (\Delta^{r_1} \vee\ \neg x_m : \neg A^{r_1})\ .$$

The index i does not occur in either Δ or $\Diamond_{2m} A$ because $[\Delta, \Diamond_{2m} A]_i$ is properly annotated. Hence,

$$r := (r_1 \upharpoonright \Delta, \Diamond_{2m} A)\ \cup\ \{i \mapsto t\}$$

is a realization function on $[\Delta, \Diamond_{2m} A]_i$. For the identity substitution σ and this r,

$$\mathsf{J5} \vdash \bigl([\Delta]_k, \Diamond_{2m} A\bigr)^{r_1} \sigma\ \to\ \bigl([\Delta, \Diamond_{2m} A]_i\bigr)^{r}\ .$$

Case $\rho = \mathsf{5b}$. For an arbitrary instance $\dfrac{[\Delta'], [\Pi', \Diamond A']}{[\Delta', \Diamond A'], [\Pi']}$ of $\mathsf{sh\text{-}5b}$, let $[\Delta]_k, [\Pi, \Diamond_{2m} A]_i$ and $[\Delta, \Diamond_{2m} A]_l, [\Pi]_j$ be properly annotated versions of its premise and conclusion respectively. Then $\dfrac{[\Delta]_k, [\Pi, \Diamond_{2m} A]_i}{[\Delta, \Diamond_{2m} A]_l, [\Pi]_j}$ is an annotated version of this instance. Consider an arbitrary realization function r_1 on $[\Delta]_k, [\Pi, \Diamond_{2m} A]_i$. By Corollary 5.14, for the term $\mathrm{invposint}(x_1)$ constructed there and the term $\mathrm{t}_!(x_1)$ from Lemma 5.13,

$$\mathsf{J5} \vdash \mathrm{invposint}(x_m) : \bigl(\neg\, \mathrm{t}_!(x_m) : x_m : \neg A^{r_1} \to\ \neg x_m : \neg A^{r_1}\bigr)\ .$$

Thus, by app and MP,

$$\mathsf{J5} \vdash ?\, \mathrm{t}_!(x_m) : \neg\, \mathrm{t}_!(x_m) : x_m : \neg A^{r_1} \to\ \bigl(\mathrm{invposint}(x_m) \cdot ?\, \mathrm{t}_!(x_m)\bigr) : \neg x_m : \neg A^{r_1}\ .$$

From the $\mathsf{j5}$-instance $\neg\, \mathrm{t}_!(x_m) : x_m : \neg A^{r_1} \to ?\, \mathrm{t}_!(x_m) : \neg\, \mathrm{t}_!(x_m) : x_m : \neg A^{r_1}$, it follows that

$$\mathsf{J5} \vdash \neg\, \mathrm{t}_!(x_m) : x_m : \neg A^{r_1} \to\ \bigl(\mathrm{invposint}(x_m) \cdot ?\, \mathrm{t}_!(x_m)\bigr) : \neg x_m : \neg A^{r_1}\ . \tag{26}$$

By propositional reasoning and Internalization Property 3.5, for some ground term p_1,

$$\mathsf{J5} \vdash p_1 : \bigl(x_m : \neg A^{r_1} \to\ \Pi^{r_1} \vee\ \neg x_m : \neg A^{r_1} \to\ \Pi^{r_1}\bigr)\ .$$

Thus, by app and MP,

$$\mathsf{J5} \vdash \mathrm{t}_!(x_m) : x_m : \neg A^{r_1} \to\ \bigl(p_1 \cdot \mathrm{t}_!(x_m)\bigr) : (\Pi^{r_1} \vee\ \neg x_m : \neg A^{r_1} \to\ \Pi^{r_1})\ .$$

By app and propositional reasoning,

$$\mathsf{J5} \vdash \mathrm{t}_!(x_m) : x_m : \neg A^{r_1} \to\ r_1(i) : (\Pi^{r_1} \vee\ \neg x_m : \neg A^{r_1})\ \to\ \bigl(p_1 \cdot \mathrm{t}_!(x_m) \cdot r_1(i)\bigr) : \Pi^{r_1}\ ,$$

which is propositionally equivalent to

$$\mathsf{J5} \vdash r_1(i) : (\Pi^{r_1} \vee \neg x_m : \neg A^{r_1}) \;\to\; \neg\, \mathrm{t}_!(x_m) : x_m : \neg A^{r_1} \vee \; s : \Pi^{r_1}$$

for $s := p_1 \cdot \mathrm{t}_!(x_m) \cdot r_1(i)$. From this and (26), by propositional reasoning,

$$\mathsf{J5} \vdash r_1(i) : (\Pi^{r_1} \vee \neg x_m : \neg A^{r_1}) \;\to\; \big(\mathrm{invposint}(x_m) \cdot ?\, \mathrm{t}_!(x_m)\big) : \neg x_m : \neg A^{r_1} \vee \; s : \Pi^{r_1} \quad . \tag{27}$$

By Internalization Property 3.5, for the propositional tautology $\neg x_m : \neg A^{r_1} \to \Delta^{r_1} \vee \neg x_m : \neg A^{r_1}$, there exists a term $t_3(x_1)$ such that from (27), by propositional reasoning, we obtain the provability in $\mathsf{J5}$ of

$$r_1(i) : (\Pi^{r_1} \vee \neg x_m : \neg A^{r_1}) \;\to\; t_3\big(\mathrm{invposint}(x_m) \cdot ?\, \mathrm{t}_!(x_m)\big) : (\Delta^{r_1} \vee \neg x_m : \neg A^{r_1}) \;\vee\; s : \Pi^{r_1} \quad . \tag{28}$$

Since $\Delta^{r_1} \to \Delta^{r_1} \vee \neg x_m : \neg A^{r_1}$ is a propositional tautology, by Internalization Property 3.5, there exists a term $t_4(x_1)$ such that

$$\mathsf{J5} \vdash r_1(k) : \Delta^{r_1} \;\to\; t_4\big(r_1(k)\big) : (\Delta^{r_1} \vee \neg x_m : \neg A^{r_1}) \quad .$$

Therefore, by the axiom sum,

$$\mathsf{J5} \vdash r_1(k) : \Delta^{r_1} \;\to\; t : (\Delta^{r_1} \vee \neg x_m : \neg A^{r_1}) \tag{29}$$

for $t := t_3\big(\mathrm{invposint}(x_m) \cdot ?\, \mathrm{t}_!(x_m)\big) + t_4\big(r_1(k)\big)$. Similarly, by (28) and sum,

$$\mathsf{J5} \vdash r_1(i) : (\Pi^{r_1} \vee \neg x_m : \neg A^{r_1}) \;\to\; t : (\Delta^{r_1} \vee \neg x_m : \neg A^{r_1}) \;\vee\; s : \Pi^{r_1} \quad . \tag{30}$$

Finally, by propositional reasoning from (29) and (30),

$$\mathsf{J5} \vdash r_1(k) : \Delta^{r_1} \vee \; r_1(i) : (\Pi^{r_1} \vee \neg x_m : \neg A^{r_1}) \;\to\; t : (\Delta^{r_1} \vee \neg x_m : \neg A^{r_1}) \;\vee\; s : \Pi^{r_1} \quad .$$

The indices l and j do not occur in Δ, Π, or $\Diamond_{2m} A$ because $[\Delta, \Diamond_{2m} A]_l, [\Pi]_j$ is properly annotated. Hence,

$$r := (r_1 \upharpoonright \Delta, \Diamond_{2m} A, \Pi) \;\cup\; \{l \mapsto t; j \mapsto s\}$$

is a realization function on $[\Delta, \Diamond_{2m} A]_l, [\Pi]_j$. For the identity substitution σ and this r,

$$\mathsf{J5} \vdash \big([\Delta]_k, [\Pi, \Diamond_{2m} A]_i\big)^{r_1} \sigma \;\to\; \big([\Delta, \Diamond_{2m} A]_l, [\Pi]_j\big)^{r} \quad .$$

Case $\rho = \mathsf{5c}$**.** For an arbitrary instance $\dfrac{[\Delta', [\Pi', \Diamond A']]}{[\Delta', \Diamond A', [\Pi']]}$ of $\mathsf{sh\text{-}5c}$, let $[\Delta, [\Pi, \Diamond_{2m} A]_i]_k$ and $[\Delta, \Diamond_{2m} A, [\Pi]_j]_l$ be properly annotated versions of its premise and conclusion respectively. Then $\dfrac{[\Delta, [\Pi, \Diamond_{2m} A]_i]_k}{[\Delta, \Diamond_{2m} A, [\Pi]_j]_l}$ is an annotated version of this instance. Consider an arbitrary realization function r_1 on $[\Delta, [\Pi, \Diamond_{2m} A]_i]_k$. As in the case $\rho = \mathsf{5b}$, cf. (27),

$$\mathsf{J5} \vdash r_1(i) : (\Pi^{r_1} \vee \neg x_m : \neg A^{r_1}) \;\to\; \big(\mathrm{invposint}(x_m) \cdot ?\, \mathrm{t}_!(x_m)\big) : \neg x_m : \neg A^{r_1} \vee \; s : \Pi^{r_1} \quad .$$

Thus, by propositional reasoning,

$$\mathsf{J5} \vdash \Delta^{r_1} \vee \; r_1(i) : (\Pi^{r_1} \vee \neg x_m : \neg A^{r_1}) \;\to\; \Delta^{r_1} \vee \; \big(\mathrm{invposint}(x_m) \cdot ?\, \mathrm{t}_!(x_m)\big) : \neg x_m : \neg A^{r_1} \vee \; s : \Pi^{r_1} \quad .$$

By Internalization Property 3.5, there exists a term $s_1(x_1)$ such that

$$\mathsf{J5} \vdash r_1(k) : \big(\Delta^{r_1} \vee \; r_1(i) : (\Pi^{r_1} \vee \neg x_m : \neg A^{r_1})\big) \;\to\; s_1\big(r_1(k)\big) : (\Delta^{r_1} \vee \; t_3 : \neg x_m : \neg A^{r_1} \vee \; s : \Pi^{r_1}) \quad , \tag{31}$$

where $t_3 := \mathrm{invposint}(x_m) \cdot ?\, \mathrm{t}_!(x_m)$. By Lemma 5.11, for the term $\mathrm{fact}(x_1)$ constructed there,

$$\mathsf{J5} \vdash \mathrm{fact}(t_3) : (t_3 : \neg x_m : \neg A^{r_1} \;\to\; \neg x_m : \neg A^{r_1}) \quad . \tag{32}$$

By propositional reasoning and Internalization Property 3.5, for some ground term p_2,

$$\mathsf{J5} \vdash p_2 : \big((t_3 : \neg x_m : \neg A^{r_1} \to \neg x_m : \neg A^{r_1}) \to \Delta^{r_1} \vee t_3 : \neg x_m : \neg A^{r_1} \vee s : \Pi^{r_1} \to \Delta^{r_1} \vee \neg x_m : \neg A^{r_1} \vee s : \Pi^{r_1}\big) .$$

From this and (32), by app and MP, it follows that

$$\mathsf{J5} \vdash \big(p_2 \cdot \mathrm{fact}(t_3)\big) : (\Delta^{r_1} \vee t_3 : \neg x_m : \neg A^{r_1} \vee s : \Pi^{r_1} \to \Delta^{r_1} \vee \neg x_m : \neg A^{r_1} \vee s : \Pi^{r_1}) .$$

It follows by app and MP that

$$\mathsf{J5} \vdash s_1\big(r_1(k)\big) : (\Delta^{r_1} \vee t_3 : \neg x_m : \neg A^{r_1} \vee s : \Pi^{r_1}) \to t_4 : (\Delta^{r_1} \vee \neg x_m : \neg A^{r_1} \vee s : \Pi^{r_1})$$

for $t_4 := p_2 \cdot \mathrm{fact}(t_3) \cdot s_1\big(r_1(k)\big)$. From this and (31), by propositional reasoning,

$$\mathsf{J5} \vdash r_1(k) : \big(\Delta^{r_1} \vee r_1(i) : (\Pi^{r_1} \vee \neg x_m : \neg A^{r_1})\big) \to t_4 : (\Delta^{r_1} \vee \neg x_m : \neg A^{r_1} \vee s : \Pi^{r_1}) .$$

The indices l and j do not occur in Δ, Π, or $\Diamond_{2m}A$ because $[\Delta, \Diamond_{2m}A, [\Pi]_j]_l$ is properly annotated. Hence,

$$r := (r_1 \restriction \Delta, \Diamond_{2m}A, \Pi) \cup \{j \mapsto s; l \mapsto t_4\}$$

is a realization function on $[\Delta, \Diamond_{2m}A, [\Pi]_j]_l$. For the identity substitution σ and this r,

$$\big([\Delta, [\Pi, \Diamond_{2m}A]_i]_k\big)^{r_1}\sigma \to \big([\Delta, \Diamond_{2m}A, [\Pi]_j]_l\big)^{r} . \qquad \square$$

Theorem 5.16 (Realization). Let a modal logic ML and a justification logic JL be chosen respectively from the first and the second row of the same column of the following table:

K	D	T	KB	K4	K5	DB	D4	D5	TB	S4	K45	D45	S5	KB5
J	JD	JT	JB	J4	J5	JDB	JD4	JD5	JTB	JT4	J45	JD45	JT45	JB45

Note that the first row contains all the 15 modal logics from the modal cube. Then $\mathsf{JL}^\circ = \mathsf{ML}$. Moreover, for each $A' \in \mathsf{ML}$, there exists a properly annotated version A of it and a realization function r on A such that $\mathsf{JL} \vdash A^r$.

Proof. The inclusion $\mathsf{JL}^\circ \subseteq \mathsf{ML}$ is easy to prove by induction on a proof in JL since the forgetful projections of axioms of any justification logic are derivable and the forgetful projections of its rules are admissible in the modal logic with the corresponding axiom system.

Let us now turn to the more interesting opposite inclusion. As discussed at the beginning of this section, with the exception of the case of the modal logic K, whose sequent system is denoted by SK, ML also denotes the sequent system (an extension of SK according to Figure 7) for the modal logic ML. Be it SK or ML for $\mathsf{ML} \neq \mathsf{K}$, this sequent system is complete with respect to the modal logic ML by Theorem 5.3. By Lemmas 5.4–5.9, the rules $\mathsf{sh\text{-}id}$, $\mathsf{sh\text{-}\vee}$, $\mathsf{sh\text{-}\wedge}$, $\mathsf{sh\text{-}ctr}$, $\mathsf{sh\text{-}exch}$, $\mathsf{sh\text{-}\Box}$, and $\mathsf{sh\text{-}k}$, i.e., the shallow versions of all the rules of the sequent system SK for the modal logic K, are realizable in J. If $\mathsf{ML} \neq \mathsf{K}$, then JL is an extension of J, so the shallow versions of these nested rules are also realizable in JL by Fact 4.9. Let $\rho \in \{\mathsf{d}, \mathsf{t}, \mathsf{b}, \mathsf{4}, \mathsf{5a}, \mathsf{5b}, \mathsf{5c}\}$ be one of the remaining rules of the sequent system ML. It is easy to see from the table above and Figure 7 that JL is an extension of the justification logic, the realizability of $\mathsf{sh}\text{-}\rho$ in which is stated in Lemma 5.15. Thus, $\mathsf{sh}\text{-}\rho$ is realizable in JL by Fact 4.9. Again, the shallow versions of all the rules of ML are realizable in JL. Let $A' \in \mathsf{ML}$, i.e., $\mathsf{ML} \vdash A'$, for a modal formula A'. By completeness of our sequent system, the sequent A' is provable in it. Therefore, by Theorem 4.12, for some properly annotated version A of A', there exists a realization function r on A such that $\mathsf{JL} \vdash A^r$. Clearly, $(A^r)^\circ = A'$. Hence, $A' \in \mathsf{JL}^\circ$. $\square$

6. Embedding and the Modular Realization Theorem

So far, we have introduced 24 justification logics. However, only 15 of them are connected to a modal logic by Theorem 5.16. In this section, we define what it means for one logic to embed in another and show that the justification counterparts (as defined in Section 2) of a modal logic all mutually embed in each other and, hence, are pairwise equivalent. This enables us to prove a modular realization theorem that connects every modal logic to all of its justification counterparts, thus yielding a realization theorem that involves all of the 24 justification logics.

The notion of embedding we introduce is quite natural. Consider the situation in modal logic first. It is common to formulate modal logics with a fixed but unspecified complete set of propositional axioms. This creates no ambiguity because the set of theorems resulting from different axiomatizations remains the same. The only change is that, in general, the proof of a formula depends on the given axiomatization; in particular, an axiom under one axiomatization may require a more involved proof under another axiomatization. The situation with justification logics is more nuanced because proofs are represented in the object language. Therefore, for justification logic, different proofs due to alternative axiomatizations become different theorems of the logic, the difference being in the terms used. In the above mentioned case of an axiom turned theorem, a constant that justifies the axiom needs to be replaced with a more complicated term. As a result, an insignificant change in the propositional axiomatization leads to a different set of theorems, i.e., to a different logic.

The idea that this change of the logic is not significant has been captured by Fitting [15], who was the first to introduce the notions of embedding and equivalence of justification logics. In his opinion, the change of a propositional axiomatization leads to a different but *equivalent* logic, where equivalence is defined as a two-way *embedding.* A logic JL_1 *embeds* in a logic JL_2, provided there is a mapping from constants of JL_1 to terms of JL_2 that converts each theorem of JL_1 into a theorem of JL_2.

Fitting's notion of embedding is also sufficient to demonstrate that changing the non-propositional part of the axiomatization in a provably equivalent way and/or changing the primary Boolean connectives of the logic would lead to an equivalent logic (in the latter case, provided the embedding also does the appropriate Boolean conversions). However, as we will soon show, there are justification logics that realize the same modal logic but are not equivalent with respect to Fitting's definition. These logics differ in their sets of operations on justifications. For instance, we will demonstrate that both $\mathsf{JT45}$ and $\mathsf{JT5}$ realize $\mathsf{S5}$, even though $\mathsf{JT5}$ lacks the operation of positive introspection: although ! is present in the language, the axiom $\mathsf{j4}$ describing its properties is not a theorem of $\mathsf{JT5}$.

To explain in which sense $\mathsf{JT5}$ is equivalent to $\mathsf{JT45}$, consider an analogous situation when Boolean connectives are changed. If conjunction is not present in the language, it can be defined via primary connectives. We propose to do the same with operations on justifications. In particular, ! missing in $\mathsf{JT5}$ can be defined via the remaining operations. In other words, $\mathsf{j4}$ can be proved in $\mathsf{JT5}$ if $!s$ is replaced with another term $\mathrm{t}_!(s)$. Hence, to obtain a sufficiently general notion of equivalence, we generalize Fitting's definition of an embedding from 0-ary operations (i.e., constants) to arbitrary n-ary operations. Informally, we say that JL_1 embeds in JL_2, provided there is a mapping from operations of JL_1 to terms of JL_2 that maps each n-ary operation to a term with n distinct variables such that each theorem of JL_1 is converted into a theorem of JL_2. We call such a mapping an *operation translation.*

Remark 6.1 (Avoiding trivial equivalences). To see why the property of realizing the same modal logic by itself does not qualify as a definition of equivalence, imagine a "justification logic" that is obtained from $\mathsf{JT45}$ by replacing all the terms with a single constant. Such a logic trivially realizes $\mathsf{S5}$, but intuitively it should not be considered equivalent to $\mathsf{JT45}$.

Many definitions and results in this section apply to a more general class of justification logics than the one discussed in this paper. Everything up to Fact 6.11 is general enough to be applicable to logics with any collection of justification terms. Lemma 6.13 and Theorem 6.14 hold for justification logics that satisfy the Internalization Property and can prove sum. The remaining results are specific to what we call the extensions of J. Note that all the results also apply to logics with different languages (recall that all the extensions of J have the same language).

Even though the operations of our logics are at most binary, we want to keep the following definitions as general as possible. Note that, in this general setting, we use prefix notation also for binary operations.

Definition 6.2 (Operation translation). Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be two justification languages. An *operation translation* ω (from $\mathcal{L}_1$ to $\mathcal{L}_2$) is a total function that for each $n \geq 0$, maps every n-ary operation $*$ of $\mathcal{L}_1$ to an $\mathcal{L}_2$-term $\omega(*) = \omega_*(x_1, \dots, x_n)$. In particular, constants of $\mathcal{L}_1$ are mapped to ground terms of $\mathcal{L}_2$. For any $\mathcal{L}_1$-term t, the term $t\omega$ is inductively defined as follows: for any variable x_i, $x_i\omega := x_i$; if $*$ is an n-ary operation of $\mathcal{L}_1$, $n \geq 0$, then $\big(*(t_1, \dots, t_n)\big)\omega := \omega_*(t_1\omega, \dots, t_n\omega)$. Similarly, for any $\mathcal{L}_1$-formula A, the formula $A\omega$ is inductively defined as follows: for any proposition P_i, $P_i\omega := P_i$; ω distributes through all Boolean connectives; finally, $(t : B)\omega := (t\omega) : (B\omega)$.

Whenever safe, we omit parentheses and write, e.g., $*(t_1, \dots, t_n)\omega$ instead of $\big(*(t_1, \dots, t_n)\big)\omega$.
As an example, let ? be a unary operation in the language of $\mathcal{L}_1$ and $\omega(?) = \omega_?(x_1)$. Then

$$\big(\neg s : A \to ?\, s : \neg s : A\big)\omega \;=\; \neg(s\omega) : (A\omega) \to \omega_?(s\omega) : \neg(s\omega) : (A\omega)\ .$$

Fact 6.3 (Properties of Operation Translation). Let ω be an operation translation from $\mathcal{L}_1$ to $\mathcal{L}_2$ and let t and A be an $\mathcal{L}_1$-term and an $\mathcal{L}_1$-formula respectively. Then

(1) $t\omega$ is an $\mathcal{L}_2$-term and $A\omega$ is an $\mathcal{L}_2$-formula;

(2) $A^\circ = (A\omega)^\circ$;

(3) for any justification variable x, we have that x occurs in $A\omega$ iff x occurs in A.

Definition 6.4 (Embedding and equivalence). Let JL_1 and JL_2 be justification logics over languages $\mathcal{L}_1$ and $\mathcal{L}_2$ respectively. We say that JL_1 *embeds* in JL_2, written $\mathsf{JL}_1 \mathrel{\tilde{\subseteq}} \mathsf{JL}_2$, if there exists an operation translation ω from $\mathcal{L}_1$ to $\mathcal{L}_2$ such that $\mathsf{JL}_1 \vdash A$ implies $\mathsf{JL}_2 \vdash A\omega$ for any $\mathcal{L}_1$-formula A. We call JL_1 and JL_2 *equivalent*, written $\mathsf{JL}_1 \equiv \mathsf{JL}_2$, if $\mathsf{JL}_1 \mathrel{\tilde{\subseteq}} \mathsf{JL}_2$ and $\mathsf{JL}_2 \mathrel{\tilde{\subseteq}} \mathsf{JL}_1$.

By the following two lemmas, equivalent logics realize the same modal logic.

Lemma 6.5 (Equivalence and Forgetful Projection). Let JL_1 and JL_2 be justification logics over languages $\mathcal{L}_1$ and $\mathcal{L}_2$ respectively. $\mathsf{JL}_1 \equiv \mathsf{JL}_2$ implies $(\mathsf{JL}_1)^\circ = (\mathsf{JL}_2)^\circ$.[2]

Proof. We show that $\mathsf{JL}_1 \mathrel{\tilde{\subseteq}} \mathsf{JL}_2$ implies $(\mathsf{JL}_1)^\circ \subseteq (\mathsf{JL}_2)^\circ$. The opposite inclusion is analogous. Let ω be an operation translation that witnesses the embedding $\mathsf{JL}_1 \mathrel{\tilde{\subseteq}} \mathsf{JL}_2$. Each modal formula $B \in (\mathsf{JL}_1)^\circ$ has the form A° for some $\mathcal{L}_1$-formula A such that $\mathsf{JL}_1 \vdash A$. By Fact 6.3 (1), $A\omega$ is an $\mathcal{L}_2$-formula. By definition of embedding, $\mathsf{JL}_2 \vdash A\omega$. By Fact 6.3 (2), $(A\omega)^\circ = A^\circ = B$. Hence, $B \in (\mathsf{JL}_2)^\circ$. ☐

The realization theorem from the previous section has an additional requirement that different occurrences of $\Diamond$ be realized by distinct variables. This requirement can also be preserved under embeddings:

Lemma 6.6 (Embedding and Realization). Let JL_1 and JL_2 be justification logics over languages $\mathcal{L}_1$ and $\mathcal{L}_2$ respectively. Let $\mathsf{JL}_1 \mathrel{\tilde{\subseteq}} \mathsf{JL}_2$ and $\mathsf{JL}_1 \vdash A^{r_1}$ for some properly annotated formula A and an $\mathcal{L}_1$-realization function r_1 on A. Then there exists an $\mathcal{L}_2$-realization function r_2 on A such that $\mathsf{JL}_2 \vdash A^{r_2}$.

Proof. Let ω be an operation translation that witnesses the embedding $\mathsf{JL}_1 \mathrel{\tilde{\subseteq}} \mathsf{JL}_2$. Then $\mathsf{JL}_2 \vdash A^{r_1}\omega$. Define $r_2(i) := r_1(i)\omega$ so that $r_2(i)$ is undefined whenever $r_1(i)$ is. Since, by Fact 6.3 (1), $r_1(i)\omega$ is an $\mathcal{L}_2$-term whenever $r_1(i)$ is defined, r_2 is an $\mathcal{L}_2$-prerealization function on A. Whenever $r_2(2k)$ is defined, $r_1(2k) = x_k$ since r_1 is a realization function. Hence, $r_2(2k) = r_1(2k)\omega = x_k\omega = x_k$. Thus, r_2 is also a realization function. It is easy to check by induction on the structure of A that $A^{r_1}\omega = A^{r_2}$. ☐

[2]Note that the definition of forgetful projection does not depend on which justification terms are used in the logic.

Lemma 6.7 (Extension and Embedding). Let JL_1 and JL_2 be justification logics over languages $\mathcal{L}_1 \subseteq \mathcal{L}_2$. If $\mathsf{JL}_1 \subseteq \mathsf{JL}_2$, then $\mathsf{JL}_1 \mathrel{\tilde{\subseteq}} \mathsf{JL}_2$.

Proof. Let ω^{id} denote the *identity operation translation* such that $\omega^{id}(*) := *(x_1, \ldots, x_n)$ for each n-ary $\mathcal{L}_1$-operation $*$. Since $\mathcal{L}_1 \subseteq \mathcal{L}_2$, clearly ω^{id} is an operation translation from JL_1 to JL_2. It is easy to show by induction on the structure of an $\mathcal{L}_1$-term t and of an $\mathcal{L}_1$-formula A that $t = t\omega^{id}$ and $A = A\omega^{id}$. Hence, if $\mathsf{JL}_1 \vdash A$, then $\mathsf{JL}_1 \vdash A\omega^{id}$, and, consequently, $\mathsf{JL}_2 \vdash A\omega^{id}$. Thus, $\mathsf{JL}_1 \mathrel{\tilde{\subseteq}} \mathsf{JL}_2$. □

In order to show that $\equiv$ is indeed an equivalence relation, we need the following auxiliary lemma.

Lemma 6.8 (Operation Translation and Substitution). Let ω be an operation translation from a language $\mathcal{L}_1$ to a language $\mathcal{L}_2$ and let σ be an $\mathcal{L}_1$-substitution. Then for any $\mathcal{L}_1$-term t, we have $(t\sigma)\omega = (t\omega)\sigma'$, where σ' is the $\mathcal{L}_2$-substitution defined by $\sigma'(x) := \sigma(x)\omega$ for any variable x.

Proof. By induction on the structure of t. If t is a variable x, then $(x\omega)\sigma' = x\sigma' = \sigma'(x) = \sigma(x)\omega = (x\sigma)\omega$. If $t = *(t_1, \ldots, t_n)$ for some n-ary $\mathcal{L}_1$-operation $*$, $n \geq 0$, then

$$\big(*(t_1, \ldots, t_n)\sigma\big)\omega \;=\; *(t_1\sigma, \ldots, t_n\sigma)\omega \;=\; \omega_*\big((t_1\sigma)\omega, \ldots, (t_n\sigma)\omega\big) \enspace .$$

By induction hypothesis, this is the same as

$$\omega_*\big((t_1\omega)\sigma', \ldots, (t_n\omega)\sigma'\big) \;=\; \big(\omega_*(t_1\omega, \ldots, t_n\omega)\big)\sigma' \;=\; \big(*(t_1, \ldots, t_n)\omega\big)\sigma' \enspace .$$

The penultimate equality holds because the only variables that occur in $\omega_*(t_1\omega, \ldots, t_n\omega)$ are those that occur in one of $t_1\omega, \ldots, t_n\omega$. □

Lemma 6.9 (Equivalence Relation). The relation $\tilde{\subseteq}$ is a preorder. Accordingly, $\equiv$ is an equivalence relation.

Proof. Since each logic is a trivial extension of itself, it follows from Lemma 6.7 that each logic embeds in itself. Hence, $\tilde{\subseteq}$ is reflexive.

Let JL_1, JL_2, and JL_3 be justification logics over languages $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_3$ respectively. Let operation translations ω' and ω'' witness the embeddings $\mathsf{JL}_1 \mathrel{\tilde{\subseteq}} \mathsf{JL}_2$ and $\mathsf{JL}_2 \mathrel{\tilde{\subseteq}} \mathsf{JL}_3$ respectively. We show $\mathsf{JL}_1 \mathrel{\tilde{\subseteq}} \mathsf{JL}_3$. For every $\mathcal{L}_1$-formula A, $\mathsf{JL}_1 \vdash A$ implies $\mathsf{JL}_2 \vdash A\omega'$. Accordingly, for every $\mathcal{L}_2$-formula B, $\mathsf{JL}_2 \vdash B$ implies $\mathsf{JL}_3 \vdash B\omega''$. Let $\mathsf{JL}_1 \vdash A$ for an $\mathcal{L}_1$-formula A. It follows that $\mathsf{JL}_3 \vdash (A\omega')\omega''$. Let ω be defined by $\omega(*) := \omega'(*)\omega''$ for every n-ary $\mathcal{L}_1$-operation $*$, $n \geq 0$. Since $\omega'(*)$ is an $\mathcal{L}_2$-term with $x_1, \ldots, x_n$ as its only variables, it follows from Facts 6.3 (1) and 6.3 (3) that $\omega'(*)\omega''$ is an $\mathcal{L}_3$-term with the same variables. Hence, ω is an operation translation from $\mathcal{L}_1$ to $\mathcal{L}_3$. It is now sufficient to show that $(A\omega')\omega'' = A\omega$. To this end, we show that $(t\omega')\omega'' = t\omega$ for every $\mathcal{L}_1$-term t by induction on the structure of t.

If t is a variable x, then $(x\omega')\omega'' = x\omega'' = x = x\omega$. Let $t = *(t_1, \ldots, t_n)$ for some n-ary $\mathcal{L}_1$-operation $*$, $n \geq 0$. Then

$$\big(*(t_1, \ldots, t_n)\omega'\big)\omega'' \;=\; \big(\omega'_*(t_1\omega', \ldots, t_n\omega')\big)\omega'' \enspace ;$$

in other words, for the $\mathcal{L}_2$-substitution $\sigma := \{x_i \mapsto t_i\omega' \mid 1 \leq i \leq n\} \cup \{x_i \mapsto x_i \mid i > n\}$,

$$\big(*(t_1, \ldots, t_n)\omega'\big)\omega'' \;=\; \big(\omega'(*)\sigma\big)\omega'' \enspace . \tag{33}$$

By definition, $*(t_1, \ldots, t_n)\omega = \omega_*(t_1\omega, \ldots, t_n\omega)$. By induction hypothesis, this is the same as

$$\omega_*\big((t_1\omega')\omega'', \ldots, (t_n\omega')\omega''\big) \;=\; \omega(*)\sigma'$$

for the $\mathcal{L}_3$-substitution $\sigma' := \{x_i \mapsto (t_i\omega')\omega'' \mid 1 \leq i \leq n\} \cup \{x_i \mapsto x_i \mid i > n\}$. By definition of ω, we have $\omega(*)\sigma' = \big(\omega'(*)\omega''\big)\sigma'$. Altogether,

$$*\,(t_1, \ldots, t_n)\omega \;=\; \big(\omega'(*)\omega''\big)\sigma' \enspace . \tag{34}$$

Note that $\sigma'(x) = \sigma(x)\omega''$ for any variable x. Therefore, by Lemma 6.8, we have $\big(\omega'(*)\sigma\big)\omega'' = \big(\omega'(*)\omega''\big)\sigma'$ and, by (33) and (34), $\big(*(t_1, \ldots, t_n)\omega'\big)\omega'' = *(t_1, \ldots, t_n)\omega$. Hence, $\tilde{\subseteq}$ is transitive.

Thus, $\tilde{\subseteq}$ is a preorder. The definition of $\equiv$ is a standard definition of the equivalence relation induced by the preorder $\tilde{\subseteq}$. □

Our goal is to find sufficient conditions for two logics to embed in each other. Axioms (formula schemas in general) and constants play a fundamental role in this respect.

Definition 6.10 (Formula schema). Let $\mathcal{L}_1$ be any justification language. Any $\mathcal{L}_1$-formula of the form $A(x_1, \ldots, x_n, P_1, \ldots, P_k)$, with $n, k \geq 0$ and with all variables and propositions indicated, is called a *formula representation* of an $\mathcal{L}_1$*-formula schema* S. Then for arbitrary $\mathcal{L}_1$-terms $t_1, \ldots, t_n$ and $\mathcal{L}_1$-formulas $B_1, \ldots, B_k$, the formula $A(t_1, \ldots, t_n, B_1, \ldots, B_k)$ is called an *instance* of S. For a justification logic JL over the language $\mathcal{L}_1$, an $\mathcal{L}_1$-schema S is called *provable* in JL if the formula representation of S is a theorem of JL. For an operation translation ω from $\mathcal{L}_1$ to a justification language $\mathcal{L}_2$, the $\mathcal{L}_2$-formula schema represented by the formula $A(x_1, \ldots, x_n, P_1, \ldots, P_k)\omega$ is denoted by $S\omega$.

All the justification axioms from Figures 3 and 4 are, in fact, schemas written with variables over terms and variables over formulas. From now on, we write them using their formula representations instead. For instance, the axiom $\mathsf{j4}$ is now written as $x_1 : P_1 \to\, !x_1 : x_1 : P_1$ instead of $t : A \to\, !t : t : A$, with a variable over terms t and a variable over formulas A.

Fact 6.11 (Properties of Formula Schemas). Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be justification languages, S be an $\mathcal{L}_1$-formula schema with formula representation $A(x_1, \ldots, x_n, P_1, \ldots, P_k)$, JL be a justification logic over $\mathcal{L}_1$, and ω be an operation translation from $\mathcal{L}_1$ to $\mathcal{L}_2$. Then

(1) S is provable in JL iff all instances of S are theorems of JL;

(2) if A is an instance of S, then $A\omega$ is an instance of $S\omega$.

Now we can finally explain why constants are assigned levels. Suppose we want to embed $\mathsf{JTB4}$ in $\mathsf{JDB4}$. Without levels, $\underbrace{c : c : \ldots : c}_{m} : (x{:}P \to P)$ would be provable in $\mathsf{JTB4}$ for any constant c and $m \geq 0$. Therefore, we would have to provide an operation translation ω that maps c to some ground term p such that, in particular, $\mathsf{JDB4} \vdash \underbrace{p : p : \ldots : p}_{m} : (x : P \to P)$. It can be shown that for each m, there exists such a term p. However, according to Definition 6.4, we have to choose ω in such a way that it maps c to a single term p that works for *every* number m. This is not possible because such a p would have to be infinite. The assignment of levels to constants enables us to map constants of different levels to different ground terms.

Alternatively, we could drop the levels and change Definition 6.4 in such a way that in order to embed a logic in another one, instead of having a global operation translation, it would be enough to provide a separate operation translation for every formula. Following Fitting [14], such an embedding could be called *local*.

Lemma 6.12 (Iterated Internalization of Schemas). Let $\mathcal{L}$ be a justification language that has a binary operation $+$. Let JL be a justification logic over $\mathcal{L}$ that enjoys Internalization Property 3.5 and Substitution Lemma 3.8, in which MP is an admissible rule and $x_1{:}P_1 \to (x_1{+}x_2){:}P_1$ and $x_2{:}P_1 \to (x_1{+}x_2){:}P_1$ are provable formula schemas, collectively referred to as sum.[3] Let $S_1, \ldots, S_n$ be $\mathcal{L}$-formula schemas provable in JL. There exists an infinite sequence of ground $\mathcal{L}$-terms $p_1, p_2, \ldots$ such that for any $m > 0$ and for any $\mathcal{L}$-instance A of one of S_i, $1 \leq i \leq n$, we have $\mathsf{JL} \vdash p_m : p_{m-1} : \ldots : p_1 : A$.

Proof. Let $A_i(x_1, \ldots, x_{k_i}, P_1, \ldots, P_{l_i})$ be a formula representation of S_i, $i = 1, \ldots, n$. We show how to construct the term p_j, $j = 1, 2, \ldots$ by induction on j. For each $1 \leq i \leq n$, by Internalization Property 3.5, there exists a ground $\mathcal{L}$-term p_i^1 such that

$$\mathsf{JL} \vdash p_i^1 : A_i(x_1, \ldots, x_{k_i}, P_1, \ldots, P_{l_i}) \ .$$

[3] Earlier in this paper, sum denoted one of the axioms of J. We are using the same name here because these two formula schemas coincide with that axiom. The only difference is that instead of requiring them to be axioms as before, here we only postulate that all their instances are theorems.

Let $p_1 := p_1^1 + \ldots + p_n^1$. For each $1 \le i \le n$, by using appropriate instances of the schemas sum and the rule MP, we obtain

$$\mathsf{JL} \vdash p_1 : A_i(x_1, \ldots, x_{k_i}, P_1, \ldots, P_{l_i}) \ .$$

Assume that for some $m > 0$, we have already constructed ground $\mathcal{L}$-terms $p_1, p_2, \ldots, p_m$ such that

$$\mathsf{JL} \vdash p_m : p_{m-1} : \ldots : p_1 : A_i(x_1, \ldots, x_{k_i}, P_1, \ldots, P_{l_i})$$

for all $i = 1, \ldots, n$. We show how to construct p_{m+1}. For each $1 \le i \le n$, by Internalization Property 3.5, there exists a ground $\mathcal{L}$-term p_i^{m+1} such that

$$\mathsf{JL} \vdash p_i^{m+1} : p_m : \ldots : p_1 : A_i(x_1, \ldots, x_{k_i}, P_1, \ldots, P_{l_i}) \ .$$

Let $p_{m+1} := p_1^{m+1} + \ldots + p_n^{m+1}$. Again, for each $1 \le i \le n$, by instances of sum and MP,

$$\mathsf{JL} \vdash p_{m+1} : p_m : \ldots : p_1 : A_i(x_1, \ldots, x_{k_i}, P_1, \ldots, P_{l_i}) \ .$$

Thus, we have constructed an infinite sequence of ground $\mathcal{L}$-terms $p_1, p_2, \ldots$ such that for all $m > 0$ and for all $i = 1, \ldots, n$,

$$\mathsf{JL} \vdash p_m : p_{m-1} : \ldots : p_1 : A_i(x_1, \ldots, x_{k_i}, P_1, \ldots, P_{l_i}) \ .$$

Therefore, $\mathsf{JL} \vdash p_m : p_{m-1} : \ldots : p_1 : A$ for every $\mathcal{L}$-instance A of one of S_i, $1 \le i \le n$, by the Substitution Lemma. ☐

Since all the extensions of J have the rule MP and the axiom sum and satisfy both Internalization Property 3.5 and Substitution Lemma 3.8, we obtain the following

Corollary 6.13 (Iterated Internalization of Schemas for the Extensions of J). Lemma 6.12 holds for any extension JL of J as defined on p. 5.

Theorem 6.14 (Embedding). Let JL_1 and JL_2 be two justification logics over languages $\mathcal{L}_1$ and $\mathcal{L}_2$ respectively. Let the set of constants of $\mathcal{L}_1$ be divided into levels (cf. page 3), let $\rightarrow$ be one of binary Boolean connectives, and let MP and iAN be the only rules of JL_1. Let JL_2 and $\mathcal{L}_2$ satisfy all the conditions of Lemma 6.12. Assume the following:

(1) JL_1 is axiomatized by finitely many axioms, i.e., formula schemas;

(2) the formula representations of the axioms of JL_1 do not contain constants;[4]

(3) there exists an operation translation ω from $\mathcal{L}_1$ to $\mathcal{L}_2$ such that for every axiom S of JL_1, the $\mathcal{L}_2$-formula schema $S\omega$ is provable in JL_2.

Then JL_1 embeds in JL_2.

Proof. We have to show that there exists an operation translation ω' from $\mathcal{L}_1$ to $\mathcal{L}_2$ such that $\mathsf{JL}_1 \vdash A$ implies $\mathsf{JL}_2 \vdash A\omega'$ for any $\mathcal{L}_1$-formula A.

Let $S_1, \ldots, S_n$ be the axioms of JL_1. By assumption (3), the $\mathcal{L}_2$-schemas $S_1\omega, \ldots, S_n\omega$ are provable in JL_2. By Lemma 6.12, there exists an infinite sequence of ground $\mathcal{L}_2$-terms $p_1, p_2, \ldots$ such that for every $m > 0$ and for every $\mathcal{L}_2$-instance B of one of $S_i\omega$ for $1 \le i \le n$,

$$\mathsf{JL}_2 \vdash p_m : p_{m-1} : \ldots : p_1 : B \ . \tag{35}$$

Let the operation translation ω' be defined as follows:

$$\omega'(*) := \begin{cases} p_j & \text{if } * \text{ is an } \mathcal{L}_1\text{-constant } c_i^j \text{ of level } j > 0, \\ \omega(*) & \text{otherwise.} \end{cases}$$

[4] Naturally, the axiom *instances* can contain constants.

Clearly, ω' is an operation translation from $\mathcal{L}_1$ to $\mathcal{L}_2$.

Let A be an arbitrary theorem of JL_1. We show by induction on a JL_1-proof of A that $\mathsf{JL}_2 \vdash A\omega'$. Note that $A\omega'$ is an $\mathcal{L}_2$-formula by Fact 6.3 (1).

If A is an instance of an axiom S_i of JL_1, $1 \leq i \leq n$, then, by Fact 6.11 (2), $A\omega'$ is an instance of the $\mathcal{L}_2$-formula schema $S_i\omega'$. The latter coincides with the $\mathcal{L}_2$-formula schema $S_i\omega$ because the formula representation of S_i does not contain any constants by assumption (2) and ω agrees with ω' on operations of positive arity. Thus, $A\omega'$ is an instance of the provable schema $S_i\omega$ and is itself provable in JL_2 by Fact 6.11 (1).

If A is obtained by the rule iAN, then it is of the form $c^m_{i_m} : c^{m-1}_{i_{m-1}} : \ldots : c^1_{i_1} : B$, where B is an instance of an axiom S_i for some $1 \leq i \leq n$. As shown in the previous paragraph, $B\omega'$ is then an instance of the formula schema $S_i\omega$. By (35), we have $\mathsf{JL}_2 \vdash p_m : p_{m-1} : \ldots : p_1 : B\omega'$, which is the same as $\mathsf{JL}_2 \vdash A\omega'$ by definition of ω'.

Finally, if A is obtained by MP from $B \to A$ and B, then, by induction hypothesis, $\mathsf{JL}_2 \vdash B\omega' \to A\omega'$ and $\mathsf{JL}_2 \vdash B\omega'$, and, therefore, $\mathsf{JL}_2 \vdash A\omega'$ follows by MP. □

Since conditions (1) and (2) of this theorem hold for any extension of J,

Corollary 6.15 (Embedding for the Extensions of J). Let JL_1 and JL_2 be two extensions of J as defined on p. 5. If there exists an operation translation ω from the common language of JL_1 and JL_2 to the same language such that for every axiom S of JL_1, the formula schema $S\omega$ is provable in JL_2, then JL_1 embeds in JL_2.

We now return to our restricted set of justification logics that we call the extensions of J, which all have the same language. Corollary 6.15 can be used to prove that for every modal logic, its justification counterparts are pairwise equivalent. It will be sufficient to provide appropriate operation translations. Moreover, for all such operation translations ω, we can set $\omega(+) := x_1 + x_2$ and $\omega(\cdot) := x_1 \cdot x_2$, as in the identity operation translation, because the axioms sum and app are present in all the extensions of J.

We proceed to prove that all the justification counterparts of $\mathsf{KB5}$ (among the extensions of J) are pairwise equivalent and so are those of $\mathsf{S5}$. The following is an auxiliary lemma to be used in the proof of Lemma 6.17.

Lemma 6.16 (Consistency). For arbitrary terms t and s and an arbitrary formula A,

$$\mathsf{JD} \vdash t : A \to \neg s : \neg A \ .$$

Proof. From the app-instance $s : (A \to \bot) \to t : A \to (s \cdot t) : \bot$, we obtain by propositional reasoning and the jd-instance $(s \cdot t) : \bot \to \bot$

$$\mathsf{JD} \vdash t : A \ \to \ s : (A \to \bot) \ \to \ \bot \ ,$$

which is the same as $\mathsf{JD} \vdash t : A \to \neg s : \neg A$. □

The following lemma is the main ingredient for the construction of operation translations that witness the embeddings between justification logics.

Lemma 6.17 (Operation Replacement). There exist terms $\mathrm{t}'_!(x_1)$ and $\mathrm{t}_?(x_1)$ such that for the term $\mathrm{t}_!(x_1)$ constructed in Lemma 5.13 and for any term s and any formula A,

(1) $\mathsf{JT5} \vdash A \to ?\,s : \neg s : \neg A$;

(2) $\mathsf{JT5} \vdash s : A \to \mathrm{t}_!(s) : s : A$;

(3) $\mathsf{JB5} \vdash s : A \to \mathrm{t}'_!(s) : s : A$;

(4) $\mathsf{JB4} \vdash \neg s : A \to \mathrm{t}_?(s) : \neg s : A$;

(5) $\mathsf{JDB4} \vdash s : A \to A$;

(6) $\mathsf{JDB5} \vdash s : A \to A$.

Proof. (1) The formula $\neg s : \neg A \to ?\, s : \neg s : \neg A$ is an instance of $\mathsf{j5}$. Hence, $A \to ?\, s : \neg s : \neg A$ follows by syllogism with $A \to \neg s : \neg A$, which is the contraposition of an instance of jt.

(2) By Lemma 5.13, for the terms $\mathrm{posint}(x_1)$ and $\mathrm{t}_!(x_1)$ constructed there and for any term s and any formula A, $\mathsf{J5} \vdash \mathrm{posint}(s) : \big(s : A \to \mathrm{t}_!(s) : s : A\big)$. Since $\mathsf{J5} \subseteq \mathsf{JT5}$, also $\mathsf{JT5} \vdash \mathrm{posint}(s) : \big(s : A \to \mathrm{t}_!(s) : s : A\big)$. The desired statement now follows by MP from an instance of jt.

(3) By Lemma 5.12, for the ground term $\mathrm{invnegint}(x_1)$ constructed there,

$$\mathsf{J5} \vdash \bar{?}\, ?\, x_1 : \neg\, ?\, x_1 : \neg x_1 : P_1 \;\to\; \mathrm{invnegint}(\bar{?}\, ?\, x_1) : x_1 : P_1 \ .$$

The same formula is provable in $\mathsf{JB5}$. Since $x_1 : P_1 \to \bar{?}\, ?\, x_1 : \neg\, ?\, x_1 : \neg x_1 : P_1$ is an instance of jb, for $\mathrm{t}'_!(x_1) := \mathrm{invnegint}(\bar{?}\, ?\, x_1)$, we have $\mathsf{JB5} \vdash x_1 : P_1 \to \mathrm{t}'_!(x_1) : x_1 : P_1$ by syllogism. The desired statement now follows by the Substitution Lemma. Note that $\mathrm{t}'_!(x_1)$ depends on neither s nor A.

(4) By a propositional tautology and Internalization Property 3.5, there exists a ground term p_1 such that $\mathsf{JB4} \vdash p_1 : (x_1 : P_1 \to \neg\neg x_1 : P_1)$. By the axiom app,

$$\mathsf{JB4} \vdash !\, x_1 : x_1 : P_1 \;\to\; (p_1 \cdot !\, x_1) : \neg\neg x_1 : P_1 \ .$$

By syllogism and the $\mathsf{j4}$-instance $x_1 : P_1 \to !\, x_1 : x_1 : P_1$,

$$\mathsf{JB4} \vdash x_1 : P_1 \;\to\; (p_1 \cdot !\, x_1) : \neg\neg x_1 : P_1 \ .$$

By contraposition and Internalization Property 3.5, there exists a term $s_1(x_1)$ such that

$$\mathsf{JB4} \vdash \bar{?}(p_1 \cdot !\, x_1) : \neg (p_1 \cdot !\, x_1) : \neg\neg x_1 : P_1 \;\to\; s_1\big(\bar{?}(p_1 \cdot !\, x_1)\big) : \neg x_1 : P_1 \ .$$

From the jb-instance

$$\neg x_1 : P_1 \;\to\; \bar{?}(p_1 \cdot !\, x_1) : \neg (p_1 \cdot !\, x_1) : \neg\neg x_1 : P_1 \ ,$$

it follows by syllogism that $\neg x_1 : P_1 \to \mathrm{t}_?(x_1) : \neg x_1 : P_1$ for $\mathrm{t}_?(x_1) := s_1\big(\bar{?}(p_1 \cdot !\, x_1)\big)$. The desired statement now follows by the Substitution Lemma. Note that $\mathrm{t}_?(x_1)$ depends on neither s nor A.

(5) By the propositional tautology $P_1 \to \neg\neg P_1$ and Internalization Property 3.5, there exists a term $t(x_1)$ such that $\mathsf{J} \vdash x_1 : P_1 \to t(x_1) : \neg\neg P_1$. By contraposition and Internalization Property 3.5, there exists a term $s_2(x_1)$ such that

$$\mathsf{J} \vdash x_2 : \neg t(x_1) : \neg\neg P_1 \;\to\; s_2(x_2) : \neg x_1 : P_1 \ .$$

Again by contraposition,

$$\mathsf{J} \vdash \neg s_2(x_2) : \neg x_1 : P_1 \;\to\; \neg x_2 : \neg t(x_1) : \neg\neg P_1 \ . \tag{36}$$

Since $\mathsf{JDB4} \supseteq \mathsf{JD}$, we have $\mathsf{JDB4} \vdash !\, x_1 : x_1 : P_1 \to \neg s_2(x_2) : \neg x_1 : P_1$ by Lemma 6.16. By the $\mathsf{j4}$-instance $x_1 : P_1 \to !\, x_1 : x_1 : P_1$ and syllogism,

$$x_1 : P_1 \;\to\; \neg s_2(x_2) : \neg x_1 : P_1 \tag{37}$$

is provable in $\mathsf{JDB4}$. By syllogism with (36), $\mathsf{JDB4} \vdash x_1 : P_1 \to \neg x_2 : \neg t(x_1) : \neg\neg P_1$. By the Substitution Lemma, $\mathsf{JDB4} \vdash x_1 : P_1 \to \neg\, \bar{?}\, t(x_1) : \neg t(x_1) : \neg\neg P_1$. It follows by the contrapositive $\neg\, \bar{?}\, t(x_1) : \neg t(x_1) : \neg\neg P_1 \to P_1$ of a jb-instance and by syllogism that $\mathsf{JDB4} \vdash x_1 : P_1 \to P_1$. The desired statement now follows by the Substitution Lemma.

(6) Since $\mathsf{JDB5} \supseteq \mathsf{JB5}$, we have $\mathsf{JDB5} \vdash x_1 : P_1 \to \mathrm{t}'_!(x_1) : x_1 : P_1$ by Lemma 6.17 (3) for the term $\mathrm{t}'_!(x_1)$ constructed there. Since $\mathsf{JDB5} \supseteq \mathsf{JD}$, by Lemma 6.16,

$$\mathsf{JDB5} \vdash \mathrm{t}'_!(x_1) : x_1 : P_1 \;\to\; \neg s_2(x_2) : \neg x_1 : P_1 \ .$$

By syllogism, (37) is provable in $\mathsf{JDB5}$. It remains to repeat the final steps of the proof of Lemma 6.17 (5). □

Corollary 6.18 (Realizability of Modal Rules). The following nested rules are realizable:

(1) the b-rule in JT5;

(2) the 4-rule in JT5 and JB5;

(3) the 5-rule in JB4;

(4) the t-rule in JDB4 and JDB5.

Proof. Using Lemma 6.17, we can prove realizability of the shallow rules in the respective justification logics by repeating the proof of Lemma 5.15, replacing each use of the axiom jb with $A \to ?s : \neg s : \neg A$, of the axiom j4 for JT5 with $s : A \to t_!(s) : s : A$, of the axiom j4 for JB5 with $s : A \to t'_!(s) : s : A$, and of the axiom j5 with $\neg s : A \to t_?(s) : \neg s : A$. Note also that the axiom jt is derivable in both JDB4 and JDB5. The realizability of the nested rules follows from Lemma 4.11. □

Remark 6.19 (Why $\bar{?}$ is not ?). In [10], we used a single operation ? to formulate both the axioms j5 and jb. This decision was motivated by a desire to minimize the number of operations on justifications. It was possible to use ? to realize the modal axiom b in JT5 because $A \to ?s : \neg s : \neg A$ is provable in JT5 (cf. Lemma 6.17 (1)). Hence, JB embeds in JT5 by an operation translation that replaces $\bar{?}$ with ?. However, the same operation translation does not embed JB in J5; nor does the inverse operation translation that replaces ? with $\bar{?}$ embed J5 in JB. In fact, no operation translation embeds JB in J5 or J5 in JB. Indeed, if JB $\tilde{\subseteq}$ J5, then, by the proof of Lemma 6.5 and by Theorem 5.16, KB = JB$^\circ$ $\subseteq$ J5° = K5, which is not the case since the modal axiom b is not provable in K5. An analogous argument shows that J5 does not embed in JB. Since each of J5 and JB can be viewed as J supplied with the definition of ? and of $\bar{?}$ respectively, the argument just given shows that ? and $\bar{?}$ are different operations.

Theorem 6.20 (Equivalences).

(1) JB4 $\equiv$ JB5 $\equiv$ JB45.

(2) JT5 $\equiv$ JT45 $\equiv$ JTB45 $\equiv$ JTB4 $\equiv$ JDB4 $\equiv$ JDB45 $\equiv$ JDB5 $\equiv$ JTB5.

Proof. To show each embedding, according to Corollary 6.15, it is sufficient to provide an operation translation ω such that for every axiom S of one logic, the formula schema $S\omega$ is provable in the other. In the following proof, we provide such an ω for each embedding. Recall that all the extensions of J have common language.

(1) Since $\equiv$ is an equivalence relation induced by $\tilde{\subseteq}$, it is sufficient to show a circular chain of three embeddings: JB4 $\tilde{\subseteq}$ JB5 $\tilde{\subseteq}$ JB45 $\tilde{\subseteq}$ JB4.

JB4 $\tilde{\subseteq}$ JB5: Let $\omega^{!\text{-elim}}$ agree with the identity operation translation ω^{id} (see the proof of Lemma 6.7), except that $\omega^{!\text{-elim}}(!) := t'_!(x_1)$. Since each axiom S of JB4, except for j4, is also an axiom of JB5 and since its formula representation does not contain !, $S\omega^{!\text{-elim}} = S$ is provable in JB5. For the only remaining axiom, j4,

$$(x_1 : P_1 \to \, !x_1 : x_1 : P_1)\omega^{!\text{-elim}} = x_1 : P_1 \to t'_!(x_1) : x_1 : P_1 \ , \tag{38}$$

which is provable in JB5 by Lemma 6.17 (3).

JB5 $\tilde{\subseteq}$ JB45: Follows from Lemma 6.7.

JB45 $\tilde{\subseteq}$ JB4: Let $\omega^{?\text{-elim}}$ agree with ω^{id}, except that $\omega^{?\text{-elim}}(?) := t_?(x_1)$. Since each axiom S of JB45, except for j5, is also an axiom of JB4 and since its formula representation does not contain ?, $S\omega^{?\text{-elim}} = S$ is provable in JB4. For the only remaining axiom, j5,

$$(\neg x_1 : P_1 \to \, ?x_1 : \neg x_1 : P_1)\omega^{?\text{-elim}} = \neg x_1 : P_1 \to t_?(x_1) : \neg x_1 : P_1 \ , \tag{39}$$

which is provable in JB4 by Lemma 6.17 (4).

(2) Again, it is sufficient to demonstrate a circular chain of eight embeddings:

$$\mathsf{JT5} \ \tilde{\subseteq} \ \mathsf{JT45} \ \tilde{\subseteq} \ \mathsf{JTB45} \ \tilde{\subseteq} \ \mathsf{JTB4} \ \tilde{\subseteq} \ \mathsf{JDB4} \ \tilde{\subseteq} \ \mathsf{JDB45} \ \tilde{\subseteq} \ \mathsf{JDB5} \ \tilde{\subseteq} \ \mathsf{JTB5} \ \tilde{\subseteq} \ \mathsf{JT5} \ .$$

Among these, four are immediate from Lemma 6.7:

$$\mathsf{JT5} \mathrel{\tilde{\subseteq}} \mathsf{JT45}, \quad \mathsf{JT45} \mathrel{\tilde{\subseteq}} \mathsf{JTB45}, \quad \mathsf{JDB4} \mathrel{\tilde{\subseteq}} \mathsf{JDB45}, \quad \text{and} \quad \mathsf{JDB5} \mathrel{\tilde{\subseteq}} \mathsf{JTB5} \ .$$

We now prove the remaining four embeddings.

$\mathsf{JTB45} \mathrel{\tilde{\subseteq}} \mathsf{JTB4}$: The operation translation $\omega^{?\text{-elim}}$ defined above witnesses the embedding. Indeed, as in the case of $\mathsf{JB45} \mathrel{\tilde{\subseteq}} \mathsf{JB4}$, all the axioms of $\mathsf{JTB45}$, except for $\mathsf{j5}$, remain axioms in $\mathsf{JTB4}$ and their formula representations do not contain $?$. As noted above, the operation translation (39) of the formula representation of $\mathsf{j5}$ is provable in $\mathsf{JB4}$ and, hence, in its extension $\mathsf{JTB4}$.

$\mathsf{JTB4} \mathrel{\tilde{\subseteq}} \mathsf{JDB4}$: The identity operation translation ω^{id} witnesses the embedding. Indeed, since for each axiom S of $\mathsf{JTB4}$, we have $S\omega^{id} = S$, it remains to note that all but one axiom of $\mathsf{JTB4}$ remain axioms in $\mathsf{JDB4}$. The only remaining axiom, jt, with a formula representation $x_1 : P_1 \to P_1$, is provable in $\mathsf{JDB4}$ by Lemma 6.17 (5).

$\mathsf{JDB45} \mathrel{\tilde{\subseteq}} \mathsf{JDB5}$: The operation translation $\omega^{!\text{-elim}}$ defined above witnesses the embedding. Indeed, as in the case of $\mathsf{JB4} \mathrel{\tilde{\subseteq}} \mathsf{JB5}$, all the axioms of $\mathsf{JDB45}$, except for $\mathsf{j4}$, remain axioms in $\mathsf{JDB5}$ and their formula representations do not contain $!$. As noted above, the operation translation (38) of the formula representation of $\mathsf{j4}$ is provable in $\mathsf{JB5}$ and, hence, in its extension $\mathsf{JDB5}$.

$\mathsf{JTB5} \mathrel{\tilde{\subseteq}} \mathsf{JT5}$: Let $\omega^{\bar{?}\text{-elim}}$ agree with ω^{id}, except that $\omega^{\bar{?}\text{-elim}}(\bar{?}) := {?}\,x_1$. Since each axiom S of $\mathsf{JTB5}$, except for jb, is also an axiom of $\mathsf{JT5}$ and since its formula representation does not contain $\bar{?}$, $S\omega^{\bar{?}\text{-elim}} = S$ is provable in $\mathsf{JT5}$. For the only remaining axiom, jb,

$$(P_1 \ \to \ \bar{?}\,x_1 : \neg x_1 : \neg P_1)\omega^{\bar{?}\text{-elim}} = \ P_1 \ \to \ ?\,x_1 : \neg x_1 : \neg P_1 \ ,$$

which is provable in $\mathsf{JT5}$ by Lemma 6.17 (1). ☐

Now we are ready to prove the modular realization theorem. It states that any modal logic can be realized by each of its justification counterparts, as defined on p. 6.

Theorem 6.21 (Modular Realization). Let ML be a modal logic and let JL be one of its justification counterparts. Then $\mathsf{JL}^\circ = \mathsf{ML}$. Moreover, for each $A' \in \mathsf{ML}$, there exists a properly annotated version A of it and a realization function r on A such that $\mathsf{JL} \vdash A^r$.

Proof. All the modal logics, except for $\mathsf{KB5}$ and $\mathsf{S5}$, have only one justification counterpart, for which the statement of the theorem was proved in Theorem 5.16.

Let $\mathsf{S5} \vdash A'$. By Theorem 5.16, there exists a properly annotated version A of A' and a realization function r on A such that $\mathsf{JT45} \vdash A^r$. Let JL be any justification counterpart of $\mathsf{S5}$. By Theorem 6.20, $\mathsf{JL} \equiv \mathsf{JT45}$; hence, by Lemma 6.6, there exists a realization function r_2 on A such that $\mathsf{JL} \vdash A^{r_2}$. Clearly, $(A^{r_2})^\circ = A'$; hence, $A' \in \mathsf{JL}^\circ$. For the converse, $\mathsf{JT45}^\circ = \mathsf{S5}$ by Theorem 5.16. Since $\mathsf{JL} \equiv \mathsf{JT45}$, it follows from Lemma 6.5 that $\mathsf{JL}^\circ = \mathsf{S5}$.

The proof for $\mathsf{KB5}$ is analogous, except that $\mathsf{JB45}$ is used in place of $\mathsf{JT45}$. ☐

Corollary 6.22. For two justification logics JL_1 and JL_2, we have $\mathsf{JL}_1 \equiv \mathsf{JL}_2$ iff $(\mathsf{JL}_1)^\circ = (\mathsf{JL}_2)^\circ$. In particular, there exist distinct justification logics that are equivalent. It then follows that one logic may embed in the other without being its subset.

Remark 6.23. Alternatively, the modular realization theorem can be obtained by using the fact that, by Corollary 6.18 and Fact 4.9, each rule of the sequent system $\mathsf{S5}$ ($\mathsf{KB5}$) is realizable in every justification counterpart of the logic $\mathsf{S5}$ ($\mathsf{KB5}$). The modular realization theorem can thus be proved similarly to Theorem 5.16, using Theorem 4.12.

7. Conclusions

We have presented a general method to prove realization theorems constructively and uniformly. It can be applied to any modal logic captured by a cut-free nested sequent system. Proving a realization theorem is reduced to dealing with the non-nested versions of rules, which are essentially ordinary sequent rules without side formulas. In particular, the method has enabled us to realize the 15 modal logics of the modal cube. In the process, we have reproved in a uniform way several known realization theorems and have realized modal logics that did not have justification counterparts before.

We have demonstrated that the realization for these 15 modal logics can be made modular, independent of whether the modal sequent systems are. Our realization theorem is modular in the sense that we produce a justification counterpart for each axiomatization of a modal logic. This modularity has been achieved by introducing an equivalence relation on justification logics that is based on translations of justification operations. This equivalence relation is natural in that justification logics are equivalent iff they realize the same modal logic. Although the modular systems from [11] have turned out to be incomplete, our method should be easily applicable to the corrected versions of these systems that Brünnler and Straßburger are working on.

Since we have introduced new justification logics, an obvious next step is to look for appropriate semantics and proof systems and to investigate the decidability and complexity of these logics. Further, it could be interesting to explore the connections between the equivalence of justification logics and their decidability and complexity, e.g., whether equivalent logics are necessarily in the same complexity class.

It remains an open problem whether each valid annotated formula A can be realized with the additional restriction on a realization function r that whenever $\Diamond_{2n}B$ is a subformula of A, the variable x_n should not occur in B^r. This restriction, called *non-self-referentiality on variables*, was introduced by Fitting in [16]. The main difficulty of obtaining this extra condition via our realization method lies in the contraction rule.

In this paper, we have only considered justification logics with unrestricted axiom necessitation rule. This rule is often restricted by so-called *constant specifications*. We are confident that our results can be extended to logics with arbitrary schematic and axiomatically appropriate constant specifications.

A major open problem is to establish realizability of the cut-rule, or equivalently of modus ponens. It is not known whether cut is realizable with respect to the definition we have given or with respect to some other suitable definition of a realizable rule. A positive answer to this question would allow for direct realization proofs via Hilbert systems and, thus, would probably lead to new realization theorems—for modal logics that lack cut-free systems even in nested calculi, e.g., for logics of common knowledge (cf. [12]).

Acknowledgments

Remo Goetschi is supported by Swiss National Science Foundation grants 200021–117699 and 200020–134740. Roman Kuznets is supported by Swiss National Science Foundation grant PZ00P2–131706. The authors are grateful to Kai Brünnler for the enjoyable collaboration on the conference version of this paper and for useful suggestions, to Lutz Straßburger for selflessly working with us at the time of a crisis, and to Richard McKinley for notational suggestions. We also thank the audiences of AiML'10, of the Logic and Theory Group of the University of Bern, of the Münchenwiler seminar, of CUNY Graduate Center, and of the 2011 "Gentzen Systems and Beyond" workshop in Bern for useful comments. We owe a debt of gratitude to the anonymous reviewer for the valuable feedback that greatly helped to improve the transparency of the structure of the paper. We thank Galina Savukova for helping to make the language of the paper more readable.

[1] S.N. Artemov, Operational modal logic, Technical Report MSI 95–29, Cornell University, 1995.
[2] S.N. Artemov, Explicit provability and constructive semantics, Bulletin of Symbolic Logic 7 (2001) 1–36.
[3] S.N. Artemov, The logic of justification, The Review of Symbolic Logic 1 (2008) 477–513.
[4] S.N. Artemov, E.L. Kazakov, D. Shapiro, Logic of knowledge with justifications, Technical Report CFIS 99–12, Cornell University, 1999.
[5] S.N. Artemov, R. Kuznets, Logical omniscience as a computational complexity problem, in: A. Heifetz (Ed.), Theoretical Aspects of Rationality and Knowledge, Proceedings of the Twelfth Conference (TARK 2009), ACM, Stanford University, California, 2009, pp. 14–23.
[6] F. Baader, T. Nipkow, Term *Re*writing and *All That*, Cambridge University Press, 1998.

[7] V.N. Brezhnev, On explicit counterparts of modal logics, Technical Report CFIS 2000–05, Cornell University, 2000.
[8] K. Brünnler, Deep Inference and Symmetry in Classical Proofs, Ph.D. thesis, Technische Universität Dresden, 2003.
[9] K. Brünnler, Deep sequent systems for modal logic, Archive for Mathematical Logic 48 (2009) 551–577.
[10] K. Brünnler, R. Goetschi, R. Kuznets, A syntactic realization theorem for justification logics, in: L.D. Beklemishev, V. Goranko, V. Shehtman (Eds.), Advances in Modal Logic, Volume 8, College Publications, 2010, pp. 39–58.
[11] K. Brünnler, L. Straßburger, Modular sequent systems for modal logic, in: M. Giese, A. Waaler (Eds.), Automated Reasoning with Analytic Tableaux and Related Methods, 18th International Conference, TABLEAUX 2009, Oslo, Norway, July 6–10, 2009, Proceedings, volume 5607 of *Lecture Notes in Artificial Intelligence*, Springer, 2009, pp. 152–166.
[12] S. Bucheli, R. Kuznets, T. Studer, Two ways to common knowledge, in: T. Bolander, T. Braüner (Eds.), Proceedings of the 6th Workshop on Methods for Modalities (M4M–6 2009), Copenhagen, Denmark, 12–14 November 2009, number 262 in Electronic Notes in Theoretical Computer Science, Elsevier, 2010, pp. 83–98.
[13] M. Fitting, The logic of proofs, semantically, Annals of Pure and Applied Logic 132 (2005) 1–25.
[14] M. Fitting, Justification Logics and Conservative Extensions, Technical Report TR–2007015, CUNY Ph.D. Program in Computer Science, 2007.
[15] M. Fitting, Justification logics, logics of knowledge, and conservativity, Annals of Mathematics and Artificial Intelligence 53 (2008) 153–167.
[16] M. Fitting, Realizations and LP, Annals of Pure and Applied Logic 161 (2009) 368–387.
[17] M. Fitting, The realization theorem for S5: A simple, constructive proof, in: J. van Benthem, A. Gupta, E. Pacuit (Eds.), Games, Norms and Reasons: Logic at the Crossroads, volume 353 of *Synthese Library*, Springer, 2011, pp. 61–76.
[18] J. Garson, Modal logic, in: E.N. Zalta (Ed.), The Stanford Encyclopedia of Philosophy, 2009, Winter 2009 edition. `http://plato.stanford.edu/archives/win2009/entries/logic-modal/`.
[19] A. Guglielmi, A system of interaction and structure, ACM Transactions on Computational Logic 8 (2007).
[20] R. Kuznets, Complexity Issues in Justification Logic, Ph.D. thesis, CUNY Graduate Center, 2008.
[21] R. Kuznets, Self-referential justifications in epistemic logic, Theory of Computing Systems 46 (2010) 636–661.
[22] E. Pacuit, A note on some explicit modal logics, in: Proceedings of the 5th Panhellenic Logic Symposium, University of Athens, Athens, Greece, 2005, pp. 117–125.
[23] N. Rubtsova, Evidence reconstruction of epistemic modal logic S5, in: D. Grigoriev, J. Harrison, E.A. Hirsch (Eds.), Computer Science — Theory and Applications, First International Computer Science Symposium in Russia, CSR 2006, St. Petersburg, Russia, June 8–12, 2006, Proceedings, volume 3967 of *Lecture Notes in Computer Science*, Springer, 2006, pp. 313–321.