On the Realization and Classification of Justification Logics

Inauguraldissertation
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von

Remo Goetschi
von Galmiz FR

Leiter der Arbeit:
Prof. Dr. G. Jäger
Institut für Informatik und angewandte Mathematik

On the Realization and Classification of Justification Logics

Inauguraldissertation
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von

Remo Goetschi
von Galmiz FR

Leiter der Arbeit:
Prof. Dr. G. Jäger
Institut für Informatik und angewandte Mathematik

Von der Philosophisch-naturwissenschaftlichen Fakultät angenommen.

Der Dekan:
Prof. Dr. S. Decurtins

Bern, 25.9.2012

Contents

Introduction

Justification logic is a refinement of modal logic and, as such, studies the concepts of knowledge, belief, and provability. The single modality $\Box$ from the modal language is replaced by a family of *justification terms*. While a modal formula $\Box A$ can be read as *A is known/believed*, or *A is provable*, a justification counterpart $t\!:\!A$ of this formula is read as *A is known/believed for reason t* or *t is evidence for A*, where t is a justification term. By introducing operations on justification terms, justification logic studies the operational content of modality in various modal logics. For example, the justification counterpart of the modal axiom $\Box A \to \Box\Box A$ of positive introspection is $t : A \to\, !t : t : A$, where ! denotes a unary operation on terms. This formula can be read as *if t is evidence for A, then $!t$ is evidence for t being evidence for A*. Therefore, justification logic studies explicit knowledge or belief, while modal logic studies implicit knowledge or belief.

The first justification logic, the *Logic of Proofs* or LP, was introduced by Artemov [Art95, Art01] as a tool for giving an arithmetical semantics for the modal logic $\mathsf{S4}$. Justification logics are also interesting as epistemic logics. For example, as shown in [AK09], justification logics avoid the well-known logical omniscience problem[1] because justification terms have a structure and thus provide a "measure" of how hard it is to obtain knowledge of something. Further, justification logics can be used to analyze epistemic paradoxes such as Gettier problems [Get63] (see Artemov's paper [Art08]). Kuznets showed that self-referentiality in modal logics can be studied through their justification counterparts. See his PhD thesis [Kuz08] for an overview. Recently, Studer [Stu11] presented an application of justification logic to protocol verification.

The formal correspondence between $\mathsf{S4}$ and LP, a *realization theorem*, has two directions. First, it says that each provable formula of $\mathsf{S4}$ can be turned into a provable formula of LP by realizing, i.e., replacing,

[1] Logical omniscience denotes the unrealistic property that an agent knows all the logical consequences of her basic assumptions.

instances of $\Box$ with justification terms. The converse direction says that replacing all terms in a provable formula of LP with $\Box$ results in a modal formula provable in $\mathsf{S4}$. Similar correspondences have been established for several other modal logics by means of various proof methods.

Each modal axiom schema d, t, b, $\mathsf{4}$, or $\mathsf{5}$ has a natural corresponding justification axiom schema jd, jt, jb, $\mathsf{j4}$, or $\mathsf{j5}$ respectively. And like one can build modal logics by adding to the basic modal logic K modal axiom schemas in various combinations, one can build justification logics by adding to the basic justification logic J these justification axiom schemas in various combinations. However, a modal logic may have several axiomatizations and thus may have several justification counterparts, supposedly one for each axiomatization. These counterparts mainly differ in the set of operations on terms they employ. There is a total of 24 so obtained justification logics, as opposed to only 15 modal logics.

In this thesis, we study various aspects of these 24 justification logics: we classify them via notions of embedding and equivalence (Chapter 2), we develop a general method for proving realization theorems and use this method to prove a uniform and modular realization theorem that involves all of the 24 logics (Chapter 3), we introduce Gentzen systems for some justification logics and study the property of *inversed internalization* (Chapter 4), and we analyze which justification logics are conservative extensions of others (Chapter 5). All the results in this thesis are obtained by purely syntactic proofs.

In **Chapter 1**, we introduce modal and justification logics and state some basic properties of justification logics, the most well-known of which is internalization: if a formula A is provable, then so is $t : A$ for some term t.

In **Chapter 2**, we classify the 24 justification logics by introducing an embedding relation on them that extends that of Fitting [Fit08]. This embedding gives rise to an equivalence relation, which is natural in the sense that two equivalent justification logics realize the same modal logic. The modular realization theorem in Chapter 3 will also establish the converse direction of this statement, thereby demonstrating that the embedding provides the right level of granularity among justification logics.

The machinery of embeddings also enables us to study minimal sets of operations on terms. For instance, we show that some logics do have positive introspection even though the *operation* ! of positive introspection (and its defining axiom schema) is not present in the logic. This leads—in Chapter 3—to minimal justification logics (in the sense of number of operations used) that realize a modal logic.

Chapter 3 is the most involved part of this thesis. We first develop a general constructive method for proving realization theorems. The method applies to all modal logics that can be captured by cut-free nested sequent systems. We apply it to prove a realization theorem that uniformly connects every normal modal logic formed from the axiom schemas d, t, b, 4, and 5 with one of its justification counterparts. The notion of embedding from Chapter 2 then enables us to extend this realization theorem to all natural justification counterparts, i.e., we obtain a realization theorem that is modular in the following sense: given a modal logic ML, all its justification counterparts realize ML. We therefore present a systematic study of the effects of variant axiomatizations of a modal logic on its realizations and provide realizations that are based on alternative modal axiomatizations.

To date, there are essentially two approaches of establishing realization theorems: the syntactic approach due to Artemov [Art95, Art01] and the semantic approach due to Fitting [Fit05]. The syntactic approach makes use of cut-free sequent systems for modal logics, while the semantic approach makes use of a Kripke-style semantics for justification logics. In contrast to the semantic approach, the syntactic approach is constructive: it provides an algorithm for computing justification terms that realize all the occurrences of modalities in a given modal theorem. The semantic approach has been used to prove several realization theorems: for S4, S5, K45, and KD45 [Art08, Fit05, Rub06]. Prior to the publication of [BGK10], constructive realizations, via the syntactic approach, were available only for K, D, T, K4, D4, S4, and S5 [Art95, Art01, AKS99, Bre00, Fit09, Fit11]. In the case of S5, for which no cut-free sequent system is available, two methods have been used: via a translation from S5 to K45 [Fit11] in conjunction with the *realization merging* technique developed in [Fit09] and via a cut-free hypersequent system [AKS99]. However, neither method can be applied to other modal logics that lack cut-free sequent systems, such as K5 and KB.

Our method is of the syntactic type. It 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* [Brü03, Gug07], 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 [Brü09] that capture all the 15 normal modal logics formed by the axiom schemas d, t, b, $\mathsf{4}$, and $\mathsf{5}$, which gives us a uniform constructive realization theorem for these logics. In particular, this proves Pacuit's conjecture implicit in [Pac05] that $\mathsf{J5}$[2] is a justification counterpart of $\mathsf{K5}$. Our method also helps provide justification counterparts for the modal logics $\mathsf{D5}$, KB, DB, TB, and $\mathsf{KB5}$, which, to our knowledge, did not have justification counterparts prior to the publication of [BGK10].

In **Chapter 4**, we introduce cut-free Gentzen systems for so-called justification logics of belief, which are logics that lack axiom schemas jd and jt (the counterparts of modal axiom schemas d and t respectively). Although these systems have the drawback of not being analytic, i.e., they violate the subformula property, they help us study the property of *inversed internalization*: if a formula $t : A$ is provable, then so is A. This property holds for logics J and $\mathsf{J4}$ (the counterparts of modal logics K and $\mathsf{K4}$ respectively), but fails for all other logics of belief.

In **Chapter 5**, we study which justification logics are conservative extensions of others. For logics that contain axiom schema jt we use a method developed by Fitting [Fit08] to show that every extension of such a logic is conservative. For justification logics of belief we use the Gentzen systems from Chapter 4 to prove a partial conservativity result for various logics. The idea how to do this is due to Milnikel [Mil12], who proved a partial conservativity result for the extension $\mathsf{J4}$ of J. Apart from the syntactic proof based on Gentzen systems, he also gave a semantic proof of this result. For the so-called logics of *consistent* belief, i.e., logics that contain axiom schema jd but lack jt, we generalize a counterexample by Milnikel—used to show non-conservativity of $\mathsf{JD4}$ over JD—to prove that extensions of logics of consistent belief are *not* conservative.

[2] Pacuit used the name $\mathsf{LP(K5)}$.

Relationship to Previous Work

The results of Chapter 2—without Section 2.3—and the results of Chapter 3 are published in [GK12]. Note, however, that some proofs have been omitted in [GK12]. The definition of justification logics used there slightly differs from the one used in this thesis. Here, we allow justification logics to be restricted by so-called constant specifications. Further, for simplicity, in [GK12] a uniform, full language was used for all the justification logics, while in this thesis we use restricted languages, which allows us to study conservativity. In [BGK10], a uniform realization theorem for all the 15 normal modal logics formed by the axiom schemas d, t, b, $\mathsf{4}$, and $\mathsf{5}$ was proved. The proof of the realization theorem presented there is a special case of the general method described in Chapter 3. Further, the realization theorem in [BGK10] was not modular, i.e., it did not involve all the justification counterparts of a modal logic. In [BGK10], in order to minimize the number of operations on terms, the negative introspection operation ? was used to realize both the modal axiom schemas $\mathsf{5}$ and b. However, because of the new definition of embedding for justification logics (introduced in Chapter 2), 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 $\mathsf{5}$, and this new $\bar{?}$ by exploring the conditions under which one can be replaced by the other. Another difference from [BGK10] is that, in this thesis, justification constants are assigned levels to make the results on embedding in Chapter 2 stronger (see Remark 1.2.2 for details).

Acknowledgments

First of all, I express my gratitude to Professor Gerhard Jäger for his support, for raising my interest in mathematical logic and for giving me the opportunity to do research in this field.

I am also especially thankful to Roman Kuznets. I learned a lot from him and working with him was always a great pleasure. Many ideas and results presented in this thesis evolved from the collaboration with him.

Special thanks also go to Kai Brünnler for a successful and fun collaboration and for patiently answering my questions on proof theory.

I am also very thankful to Bettina Choffat, Thomas Strahm, and Thomas Studer, who all provided me with valuable advice whenever needed.

Further, I thank all the current and former members of our research group for many interesting discussions during work or work breaks.

Special thanks go to Galina Savukova for correcting many English mistakes in the journal paper [GK12], thereby indirectly reducing the number of mistakes in this thesis.

Last but not least, I thank Samuel Bucheli for providing me with a template for the title page, Patrick Liniger for pointing out some mistakes in this introduction and Alessio Guglielmi for the Virginia Lake LaTeX macros, of which I made extensive use in this thesis.

During my time as a PhD student, I was supported by Swiss National Science Foundation grants 200021–117699 and 200020–134740.

Bern, May 2012, Remo Goetschi

1 Modal and Justification Logics

In Section 1.1, we introduce modal logics and their axiom systems, and we briefly discuss Kripke semantics for these logics. We then introduce languages and axiom systems of the justification logics we work with (Sections 1.2 and 1.3), and define forgetful projection and realization theorems, which provide a formal connection between modal and justification logics and languages (Section 1.4). We also explain in detail the naming conventions for axiom systems and logics to be employed throughout the thesis. Finally, in Section 1.5, we state some basic properties of justification logics, such as internalization and substitution. A reader already familiar with these basics is still encouraged to skim through the chapter because the justification language we use is not entirely standard (e.g., constants are divided into levels, and there is a new operation $\bar{?}$).

1.1 Modal Logics and Axiom Systems

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 (cf. Chapter 3), which is more common and also minimizes the number of propositional sequent rules, thereby shortening many proofs.

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

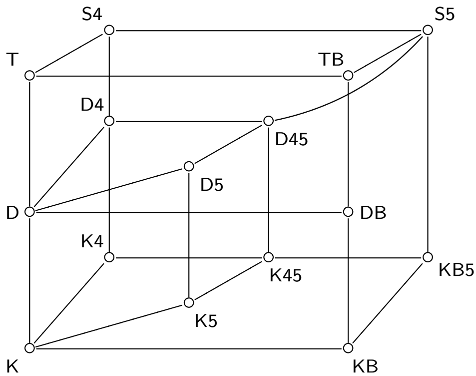

Figure 1.1: The *modal cube*

propositions to all formulas by means of the usual De Morgan laws:

$$
\begin{aligned}
\neg\neg P_i &:= P_i \ , \\
\neg(A \vee B) &:= (\neg A \wedge \neg B) \ , \\
\neg(A \wedge B) &:= (\neg A \vee \neg B) \ , \\
\neg\Box A &:= \Diamond\neg A \ , \\
\neg\Diamond A &:= \Box\neg A \ .
\end{aligned}
$$

Using this negation operation, we define implication, equivalence and falsum as usual:

$$
\begin{aligned}
(A \to B) &:= (\neg A \vee B) \ , \\
(A \leftrightarrow B) &:= \big((A \to B) \wedge (B \to A)\big) \ , \\
\bot &:= (P_1 \wedge \neg P_1) \ .
\end{aligned}
$$

Modal logics and their axiom systems. We first recall the notion of a (Hilbert-style) axiom system. The following definition applies to both modal and justification logics.

Definition 1.1.1 (Axiom System and Logic)**.** An *axiom system* (modal or justification) consists of a set of axiom schemas and a finite set of inference rules. A *proof* in an axiom system is a finite list of formulas that are either axioms, i.e., instances of axiom schemas, or follow from earlier formulas in the list by application of an inference rule.

taut:	A fixed complete set of propositional axiom schemas
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 1.2: The axiom system for the basic normal modal logic K

$\Box\bot \to \bot$	d (consistency)
$\Box A \to A$	t (factivity)
$A \to \Box\neg\Box\neg A$	b
$\Box A \to \Box\Box A$	4 (positive introspection)
$\neg\Box A \to \Box\neg\Box A$	5 (negative introspection)

Figure 1.3: Modal axiom schemas

A formula A is *provable* in an axiom system provided there exists a proof in it whose last element is A. Every axiom system *yields* a logic, namely the set of all formulas provable in the axiom system. We call a formula A provable in a logic L provided A is provable in an axiom system that yields L.

One of our goals is to prove a uniform realization theorem for all modal logics in the so-called *modal cube* from [Gar09] (see Figure 1.1). All these logics are extensions of the basic normal modal logic K that are obtained by taking its axiom system from Figure 1.2 and adding to it the modal axiom schemas d, t, b, 4, and 5 from Figure 1.3 in various combinations. Figure 1.1 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 schema k, followed by the names of the axiom schemas added to the axiom system for K from Figure 1.2, with all letters capitalized. For example, KD45 is the axiom system with additional axiom schemas 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 [Gar09], we call KB5. Thus, there is a small ambiguity between the logic KB5 and the axiom system KB5, which is resolved by explicit typification, as in this sentence. Finally, the 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.

Unless stated otherwise, from this point on, 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 logic by ML.

Kripke semantics. Proving the non-validity of a formula is a lot easier in modal logic than in justification logic. Because modal and justification logics are closely connected by realization theorems, we can use Kripke semantics for modal logic to indirectly show the non-validity of justification formulas. We briefly introduce Kripke semantics for modal logics. See, e.g., [Che80] or [HC96] for further reading.

A *Kripke frame* is a pair (W, R), where W is a nonempty set and R is a binary relation on W. The elements of W are often called *possible worlds* and R is called an *accessibility relation*. A *Kripke model* $\mathcal{M}$ is a triple (W, R, V), where (W, R) is a Kripke frame and V is a function that maps every element of W to a set of propositions. If $P_i \in V(w)$, we say proposition P_i is true at world w, and false otherwise. For a Kripke model $\mathcal{M} = (W, R, V)$ the truth relation $\models$ is defined as follows:

- $\mathcal{M}, w \models P_i$ iff $P_i \in V(w)$;
- $\mathcal{M}, w \models \neg P_i$ iff $P_i \notin V(w)$;
- $\mathcal{M}, w \models (A \vee B)$ iff $\mathcal{M}, w \models A$ or $\mathcal{M}, w \models B$;
- $\mathcal{M}, w \models (A \wedge B)$ iff $\mathcal{M}, w \models A$ and $\mathcal{M}, w \models B$;

- $\mathcal{M}, w \models \Box A$ iff $\mathcal{M}, v \models A$ for every $v \in W$ with wRv;
- $\mathcal{M}, w \models \Diamond A$ iff $\mathcal{M}, v \models A$ for some $v \in W$ with wRv.

Let A be a modal formula. We say that A is valid in $\mathcal{M}$, if $\mathcal{M}, w \models A$ for every $w \in W$. We say A is valid in a Kripke frame (W, R), if A is valid in every Kripke model (W, R, V) based on (W, R). We say that a modal logic ML is *sound and complete* with respect to a class $\mathcal{C}$ of Kripke frames, provided that any modal formula A is provable in ML iff A is valid in every member of $\mathcal{C}$.

k	(no condition)	
d	seriality	$\forall w \exists v.\ wRv$
t	reflexivity	$\forall w.\ wRw$
b	symmetry	$\forall vw.\ wRv \Longrightarrow vRw$
$\mathsf{4}$	transitivity	$\forall uvw.\ wRv \wedge vRu \Longrightarrow wRu$
$\mathsf{5}$	euclideanness	$\forall uvw.\ wRu \wedge wRv \Longrightarrow uRv$

Figure 1.4: Modal frame conditions

We identify various classes of Kripke frames by defining conditions on accessibility relations. For example, the class of reflexive Kripke frames consists of all the Kripke frames with a reflexive accessibility relation. Every axiom schema from d, t, b, $\mathsf{4}$, and $\mathsf{5}$ corresponds to a frame condition according to Figure 1.4. The basic modal logic K is sound and complete with respect to the class of all Kripke frames. Every proper extension ML of K is sound and complete with respect to the class of Kripke frames determined by the axiom schemas of ML, e.g., $\mathsf{S4}$ is sound and complete with respect to the class of reflexive and transitive Kripke frames.

1.2 Justification Logics and Axiom Systems

The languages of justification logics are given in a more traditional format with falsum and implication as primary propositional connectives.

Justification languages. Apart from formulas, the languages of justification logics have another type of syntactic objects called *justification terms*, or simply *terms*, that are given by the grammar

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

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 sub- or superscripts, whereas arbitrary terms are denoted by t, s, and q, with or without sub- or superscripts. We use the notation

$$t(x_{i_1}, \ldots, x_{i_n})$$

for terms that do not contain variables other than $x_{i_1}, \ldots, x_{i_n}$.

Constants, variables, and the operations $\cdot$ and $+$ are present in all justification logics we consider, whereas $!$, $?$, and $\bar{?}$ are optional and may or may not be present, depending on the axioms of the logic. For $S \subseteq \{!, ?, \bar{?}\}$ we denote by $\mathcal{L}(S)$ the justification language whose optional operations are those in S. For simplicity, we omit curly braces and write, e.g., $\mathcal{L}(!, ?)$ instead of $\mathcal{L}(\{!, ?\})$.

For $S \subseteq \{!, ?, \bar{?}\}$ *Justification formulas* over $\mathcal{L}(S)$ 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 in the language $\mathcal{L}(S)$. The remaining Boolean connectives $\neg, \vee, \wedge, \leftrightarrow$, and the Boolean constant $\top$ are defined as usual:

$$\begin{aligned}
\neg A &:= (A \to \bot) \ , \\
(A \vee B) &:= (\neg A \to B) \ , \\
(A \wedge B) &:= \neg(A \to \neg B) \ , \\
(A \leftrightarrow B) &:= \big((A \to B) \wedge (B \to A)\big) \ , \\
\top &:= (P_1 \vee \neg P_1) \ .
\end{aligned}$$

While writing formulas, we assume that implication is right-associative and that the colon : binds stronger than implication, conjunction, and disjunction; the last two bind stronger than implication. For example, $t : E \wedge A \to B \to C \vee D$ stands for $(t : E) \wedge \big(A \to \big(B \to (C \vee D)\big)\big)$.

Justification axiom systems. The basic justification logic is called J. Its language is $\mathcal{L}(\varnothing)$ and its axiom system—also denoted J—consists of the axioms and rules given in Figure 1.5. The zero-premise iAN-rule

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

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

Figure 1.5: The axiom system for the basic justification logic J

is called *iterated axiom necessitation*. We define it as a rule and not as an axiom schema to prevent it from referring to iself. The finiteness of the set of propositional axiom schemas in taut is required for the results on embedding in Chapter 2 (it is also a standard requirement for proving decidability and estimating complexity of justification logics).

To define *extensions of system* J we add to its axiom system axiom schemas jd, jt, jb, j4, and j5 from Figure 1.6 in various combinations. Note that these extensions also have extended languages. For example, the language of the axiom system resulting from adding j4 to J is $\mathcal{L}(!)$.

$t : \bot \to \bot$	jd (consistency)
$t : A \to A$	jt (factivity)
$A \to \bar{?}\, t : (\neg t : \neg A)$	jb (weak negative introspection)
$t : A \to !\, t : t : A$	j4 (positive introspection)
$\neg t : A \to ?\, t : (\neg t : A)$	j5 (negative introspection)

Figure 1.6: Justification axiom schemas

The axiom schemas j4 and jt occur already in Artemov [Art95]; jd and j5 were introduced by Brezhnev [Bre00] and by Pacuit [Pac05] respectively. The axiom schema jb, as presented here, was first published in [GK12]—it has also been independently proposed by Meghdad Ghari in an unpublished manuscript. The idea to use a new operation $\bar{?}$ rather than reuse ? to mimic the modal axiom schema b is consistent with the general policy that incomparable axiom schemas should be realized via different operations (cf. Remark 2.2.8).

Remark 1.2.1 (Alternative axiom necessitation)**.** In the literature, axiomatizations of justification logics that contain the axiom schema j4 often use a simpler version of the iAN-rule:

$$\frac{A \text{ is an axiom}}{c_i^1 : A}$$

The above rule is called *axiom necessitation* instead of iterated axiom necessitation. In the presence of j4, axiom necessitation is sufficient to prove the Internalization Property (cf. Section 1.5). Since we are interested in the relationships among justification logics, it is more natural to use the rule iAN suitable for all justification logics rather than switch between different versions of the rule (cf. also [Art08, Fit08]). Moreover, logics formulated with the above rule instead of iAN would not be extensions of J because, in them, no formula of the form $c_{i_n}^n : \ldots : c_{i_1}^1 : A$ would be provable. Instead of iAN, some authors use the rule

$$\frac{A \text{ is an axiom}}{\underbrace{!!\ldots!}_{n} c_i^1 : \underbrace{!\ldots!}_{n-1} c_i^1 : \ldots : c_i^1 : A}$$

This rule is not suitable for our purpose because it gives the !-operation a peculiar meaning—different from the one given by j4—and would break our results on embedding in Chapter 2.

Remark 1.2.2 (Levels of constants)**.** The assignment of levels to constants is needed for proving the results on embedding in Chapter 2—the other chapters, however, do not rely on levels. Without levels, we would be forced to use the weaker notion of local embedding (cf. Section 2.3) instead of embedding. A similar concept of levels was introduced in [Kuz10] (see also the definition of constant specification in [Art08]). Levels would not be needed for justification logics that contain the axiom schema j4 if we had chosen the first rule from Remark 1.2.1 instead of iAN.

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 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 jd are also instances of jt. Hence, adding the axiom schema 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 8 logics with two axiom systems each. For instance, the logic $\mathsf{JT5}$ is the logic of the axiom systems $\mathsf{JT5}$ and $\mathsf{JDT5}$. 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 and c_i^1 is a constant of level 1. Note that this gives us 24 extensions of J, as opposed to only 15 extensions of the modal logic K.

Unless stated otherwise, from this point on by a *justification logic*, usually denoted as JL with or without a subscript, we mean the logic of either the axiom system J or one of its extensions.

Logics that prove the axiom schema jt are usually called *logics of knowledge*[1] whereas logics that do not prove jt are called *logics of belief*. Logics of belief that additionally prove jd are sometimes called *logics of consistent belief*.

We often do not explicitly mention a formula's language when it is clear from the context. For example, when we write $\mathsf{J45} \vdash A$, it is clear that A is a formula in the language $\mathcal{L}(!, ?)$.

1.3 Constant Specifications

By the rule iAN, every sequence $c_{i_n}^n, \ldots, c_{i_1}^1$ of constants justifies every axiom of a logic. However, it can be useful to restrict the use of constants, for example, to reserve particular constants for a particular axiom or axiom schema. Such restrictions, called *constant specifications*, were used to establish complexity results [Mil07, BK12], to demonstrate potential undecidability [Kuz05], and to study the logical omniscience problem [AK06, AK09] and self-referentiality [Kuz10].

Definition 1.3.1 (Constant Specification)**.** For a justification logic JL, any set

$$\mathsf{CS} \subseteq \left\{ c_{i_n}^n : \ldots : c_{i_1}^1 : A \;\middle|\; \begin{array}{l} A \text{ is an axiom of } \mathsf{JL} \text{ and } c_{i_n}^n, \ldots, c_{i_1}^1 \text{ are} \\ \text{constants.} \end{array} \right\}$$

is called a *constant specification for* JL.

Note that constant specifications depend on the axioms of a logic. Therefore, a constant specification for one logic may not be a constant specification for another (smaller) logic.

[1] This additionally includes $\mathsf{JDB4}$, $\mathsf{JDB5}$, and $\mathsf{JDB45}$, as we show in Section 2.2.

Definition 1.3.2 ($\mathsf{JL_{CS}}$)**.** Let CS be a constant specification for a justification logic JL. We denote by $\mathsf{JL_{CS}}$ the logic obtained from JL by replacing the iAN-rule of its axiom system(s) with the rule

$$\mathsf{iAN_{CS}}\ \frac{c^n_{i_n} : \ldots : c^1_{i_1} : A \in \mathsf{CS}}{c^n_{i_n} : \ldots : c^1_{i_1} : A}$$

Using the notation $\mathsf{JL_{CS}}$, the (unrestricted) justification logic JL could be denoted $\mathsf{JL_{TCS}}$, where TCS is the *total constant specification for* JL:

$$\mathsf{TCS} := \left\{ c^n_{i_n} : \ldots : c^1_{i_1} : A \;\middle|\; \begin{array}{l} A \text{ is an axiom of } \mathsf{JL} \text{ and } c^n_{i_n}, \ldots, c^1_{i_1} \text{ are} \\ \text{constants.} \end{array} \right\} .$$

The following types of constant specifications are relevant in this thesis:

Definition 1.3.3 (Types of Constant Specifications)**.** Let CS be a constant specification for a justification logic JL.

- CS is called *axiomatically appropriate* if for every axiom A of JL and for every number $n > 0$ there exist constants $c^n_{i_n}, \ldots, c^1_{i_1}$ such that $c^n_{i_n} : \ldots : c^1_{i_1} : A \in \mathsf{CS}$. In other words: for every axiom there exist constants that justify it.
- CS is called *schematic* if whenever $c^n_{i_n} : \ldots : c^1_{i_1} : A \in \mathsf{CS}$, where A is an axiom of JL, then for every instance B of the same axiom schema, $c^n_{i_n} : \ldots : c^1_{i_1} : B \in \mathsf{CS}$. In other words: a sequence of constants justifies entire axiom schemas.
- CS is called *downward closed* if $c^n_{i_n} : c^{n-1}_{i_{n-1}} : \ldots : c^1_{i_1} : A \in \mathsf{CS}$ implies $c^{n-1}_{i_{n-1}} : \ldots : c^1_{i_1} : A \in \mathsf{CS}$ for every $n > 1$.

Note that the total constant specification TCS for a logic JL is axiomatically appropriate, schematic, and downward closed.

Also note that, e.g., $\mathsf{J4_{CS}}$ is an extension of $\mathsf{J_{CS'}}$ iff $\mathsf{CS} \supseteq \mathsf{CS'}$.

1.4 Justification Counterparts and Realization Theorems

We have named and formulated modal and justification axioms in such a way that each justification axiom schema from Figure 1.6 has a corresponding modal axiom schema from Figure 1.3. Consequently, 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. Based on this naming convention, we define the *justification counterparts* of a modal logic:

Definition 1.4.1 (Justification Counterparts)**.** Every modal logic ML has one or several *justification counterparts*, namely 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 (recall from page 16 that KB5 and S5 have more than one axiom system). Every other modal logic has only one axiom systen and therefore exactly one justification counterpart, e.g., JD45 for D45.

A deeper *correspondance* between modal logics and their justification counterparts is established by realization theorems. The first realization theorem was proved by Artemov [Art95, Art01] for the modal logic S4. It connects S4 with a justification logic that he called LP, or the *Logic of Proofs*, and that we 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 1.4.2 (Forgetful projection and realization)**.** Given a justification formula A, its *forgetful projection* A° is defined by induction on the structure of A:

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

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.

Using the above notation, the aforementioned realization theorem for $\mathsf{S4}$ can be formulated as $\mathsf{S4} = \mathsf{JT4}^\circ$.

While the inclusion $\mathsf{ML} \subseteq \mathsf{JL}^\circ$ of a realization theorem is difficult to prove, the forgetul projection part $\mathsf{JL}^\circ \subseteq \mathsf{ML}$ is trivial:

Lemma 1.4.3 (Forgetful Projection)**.** *Let* ML *be a modal logic and let* JL *be one of its justification counterparts. Let* CS *be an arbitrary constant specification for* JL*. Then* $(\mathsf{JL}_{\mathsf{CS}})^\circ \subseteq \mathsf{ML}$*.*

Proof. By a straightforward induction on a proof in $\mathsf{JL}_{\mathsf{CS}}$. Forgetful projections of JL-axioms are derivable in ML and forgetful projections of the rules MP and $\mathsf{iAN}_{\mathsf{CS}}$ are admissible in ML. □

The realization theorems we prove in Chapter 3 have an additional restriction: diamonds, i.e., negative boxes, should be realized by distinct variables.

1.5 Basic Properties of Justification Logics

We state some properties of justification logics that we use extensively in later chapters. Probably the most well-known of these properties 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 Lemma 1.5.2, originally proved for $\mathsf{JT4}$ [Art01].

Definition 1.5.1 ($A_1, \dots, A_n \vdash_{\mathsf{JL}_{\mathsf{CS}}} B$)**.** For any justification logic JL and constant specification CS for JL and for arbitrary formulas $A_1, \dots, A_n$, and B we write $A_1, \dots, A_n \vdash_{\mathsf{JL}_{\mathsf{CS}}} B$ provided there exists a $\mathsf{JL}_{\mathsf{CS}}$-proof of B, where $A_1, \dots, A_n$ may serve as additional axioms.

Lemma 1.5.2 (Internalization)**.** *For any justification logic* JL *and axiomatically appropriate constant specification* CS *for* JL*, if*

$$A_1, \dots, A_n \vdash_{\mathsf{JL}_{\mathsf{CS}}} B \ , \tag{1.1}$$

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

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

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

Proof. By induction on a $\mathsf{JL}_{\mathsf{CS}}$-proof (1.1). For an axiom, the term t is taken to be a constant of level 1. For an instance of $\mathsf{iAN}_{\mathsf{CS}}$ with the outermost constant of level n, the term t is taken to be a constant of level $n+1$. Note that, in both cases, such a constant exists because CS is assumed to be axiomatically appropriate. For a hypothesis A_i, we take $t(x_i) := x_i$. For a conclusion D of MP with premises $C \to D$ and C, by induction hypothesis, there must exist terms t_1 for $C \to D$ and t_2 for C. The term for D is taken to be $t := t_1 \cdot t_2$. $\square$

We mostly use another version of internalization, stated in Corollary 1.5.4. It can be obtained by using rule MP and the *Deduction Theorem* for justification logics. The proof of the Deduction Theorem can be almost literally adopted from that for classical propositional logic since MP remains the only rule with premises.

Lemma 1.5.3 (Deduction Theorem)**.** *For any justification logic* JL *and constant specification* CS *for* JL*, if*

$$A_1, \ldots, A_n \vdash_{\mathsf{JL}_{\mathsf{CS}}} B \ ,$$

then

$$A_1, \ldots, A_{n-1} \vdash_{\mathsf{JL}_{\mathsf{CS}}} A_n \to B \ .$$

Corollary 1.5.4 (Internalization)**.** *For any justification logic* JL *and axiomatically appropriate constant specification* CS *for* JL*, if*

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

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

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

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

For logics with positive introspection, a stronger variant of the Internalization Property holds, usually called *Lifting Lemma.*

Lemma 1.5.5 (Lifting Lemma)**.** *For any justification logic* $\mathsf{JL} \supseteq \mathsf{J4}$ *and axiomatically appropriate constant specification* CS *for* JL*, if*

$$A_1, \ldots, A_n,\ t_1 : B_1, \ldots, t_m : B_m \vdash_{\mathsf{JL}_{\mathsf{CS}}} C \ , \tag{1.2}$$

then there exists a term $t(x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n+m})$ *such that*

$$s_1 : A_1, \ldots, s_n : A_n,\ t_1 : B_1, \ldots, t_m : B_m \vdash_{\mathsf{JL}_{\mathsf{CS}}} t(s_1, \ldots, s_n, t_1, \ldots, t_m) : C$$

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

Proof. By induction on a $\mathsf{JL}_{\mathsf{CS}}$-proof (1.2). For an axiom, the term t is taken to be a constant of level 1. For an instance of $\mathsf{iAN}_{\mathsf{CS}}$ with the outermost constant of level n, the term t is taken to be a constant of level $n+1$. Note that, in both cases, such a constant exists because CS is assumed to be axiomatically appropriate. For a hypothesis A_i, $1 \leq i \leq n$, we take $t(x_i) := x_i$. For a hypothesis $t_i : B_i$, $1 \leq i \leq m$, we take $t(x_{n+i}) := !\, x_{n+i}$. For a conclusion D of MP with premises $C \to D$ and C, by induction hypothesis, there must exist terms t_1 for $C \to D$ and t_2 for C. The term for D is taken to be $t := t_1 \cdot t_2$. $\square$

Remark 1.5.6 (Lifting Lemma and Positive Introspection)**.** In Section 2.2, we show that the Lifting Lemma also holds for logics that do not contain axiom schema $\mathsf{j4}$ but still prove a schema of positive introspection: $t : A \to s(t) : t : A$ for some term $s(x_1)$.

Another important and well-known feature of justification logics is the so-called Substitution Property, by which we can substitute arbitary terms for variables and arbitrary formulas for propositions. To formulate it we introduce the concept of a substitution, which also plays an important role in the realization procedure in Chapter 3. The following definition is mostly standard, see, e.g., [BN98].

Definition 1.5.7 (Substitution)**.** A *substitution* (in a justification language $\mathcal{L}$), denoted by σ, is a total mapping from variables to $\mathcal{L}$-terms. For any $\mathcal{L}$-term t, the $\mathcal{L}$-term $t\sigma$ is then inductively defined as follows:

$$
\begin{array}{rcll}
c_i^j\sigma & := & c_i^j & \text{for any constant } c_i^j, \\
x_i\sigma & := & \sigma(x_i) & \text{for any variable } x_i, \\
(*t)\sigma & := & *(t\sigma) & \text{for any unary operation } *, \text{ and} \\
(t_1 * t_2)\sigma & := & (t_1\sigma) * (t_2\sigma) & \text{for any binary operation } *.
\end{array}
$$

We write $A\sigma$ for the formula that is obtained from A by simultaneously replacing every term t in A with $t\sigma$.

Remark 1.5.8 (n-ary Operations)**.** In Chapter 2, we work with arbitrary n-ary operations, $n \geq 0$. For such an operation $*$ the term $*(t_1, \ldots, t_n)\sigma$ is defined as $*(t_1\sigma, \ldots, t_n\sigma)$.

Lemma 1.5.9 (Substitution Property)**.** *For any justification logic* JL *and schematic constant specification* CS *for* JL*, if* $\mathsf{JL}_{\mathsf{CS}} \vdash A$*, then*

(1) $\mathsf{JL} \vdash A\sigma$ *for any substitution* σ *(in the language of* JL*) 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 distinct propositions* $P_{i_1}, \dots, P_{i_n}$ *in* A *with the formulas* $B_1, \dots, B_n$ *respectively.*

Proof. By induction on a $\mathsf{JL}_{\mathsf{CS}}$-proof of A. For an axiom, (1) and (2) hold because if C is an axiom, then $C\sigma$ and $C[P_{i_1} \mapsto B_1, \dots, P_{i_n} \mapsto B_n]$ are instances of the same axiom schema. For a conclusion $c^n_{i_n} : \dots : c^1_{i_1} : C$ of $\mathsf{iAN}_{\mathsf{CS}}$, since C, $C\sigma$, and $C[P_{i_1} \mapsto B_1, \dots, P_{i_n} \mapsto B_n]$ are instances of the same axiom schema and CS is schematic, by the $\mathsf{iAN}_{\mathsf{CS}}$-rule we have $\mathsf{JL}_{\mathsf{CS}} \vdash c^n_{i_n} : \dots : c^1_{i_1} : C\sigma$ and

$$\mathsf{JL}_{\mathsf{CS}} \vdash c^n_{i_n} : \dots : c^1_{i_1} : C[P_{i_1} \mapsto B_1, \dots, P_{i_n} \mapsto B_n] \ .$$

For a conclusion D of MP with premises $C \to D$ and C, by induction hypothesis, $\mathsf{JL}_{\mathsf{CS}} \vdash C\sigma \to D\sigma$ and $\mathsf{JL}_{\mathsf{CS}} \vdash C\sigma$. By an application of MP, $\mathsf{JL}_{\mathsf{CS}} \vdash D\sigma$. The inductive step for statement (2) is analogous. □

Remark 1.5.10 (Simultaneous substitution)**.** In Lemma 1.5.9 (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 2.2.3. 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 1.5.9 (1).

In this thesis, we make extensive use of the Internalization Property 1.5.4 and the Substitution Property 1.5.9. As a consequence, it is often required that constant specifications be axiomatically appropriate and schematic.

2 Classification via Embedding

We have introduced 24 justification logics and 15 modal logics, some of which have several justification counterparts according to Definition 1.4.1. In this chapter, we define what it means for one justification logic to embed in another and show that the justification counterparts of a modal logic all mutually embed in each other and, hence, are pairwise equivalent. This notion of equivalence is also useful to extend known realization theorems to other, equivalent logics: if two logics are equivalent and one realizes a modal logic ML, then so does the other. This enables us—in Chapter 3—to strengthen the realization theorem proved there in the sense that it connects every modal logic to all of its justification counterparts.

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 axiom schemas. 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 [Fit08], who was the first to introduce 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*. Informally, Fitting's definition of embedding is as follows: 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 terms. 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.

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 terms. 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 2.0.11 (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 Section 2.1 not only apply to what we call the extensions of J but to a more general class of justification logics. Everything up to Definition 2.1.10 is general enough to be applicable to logics with any collection of justification terms. Fact 2.1.11 holds for justification logics that satisfy the Substitution Property whereas

Lemma 2.1.13 and Theorem 2.1.15 hold for justification logics that satisfy the Internalization and Substitution Properties and prove sum. The remaining results are specific to the extensions of J.

In Section 2.1, we introduce the concept of embedding. In Section 2.2, we classify all our logics with respect to embedding. In Section 2.3, we present an alternative, weaker concept of embedding that we call *local embedding* and that allows to dispense with the levels of constants.

2.1 The Concept of Embedding

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 2.1.1 (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 $\omega(*) = \omega_*(x_1, \ldots, x_n)$. 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:

$$\begin{aligned} x_i\omega &:= x_i \enspace, \\ \bigl(*(t_1, \ldots, t_n)\bigr)\omega &:= \omega_*(t_1\omega, \ldots, t_n\omega) \enspace, \end{aligned}$$

for any variable x_i and n-ary operation $*$ of $\mathcal{L}_1$, $n \geq 0$. Similarly, for any $\mathcal{L}_1$-formula A, the formula $A\omega$ is inductively defined as follows:

$$\begin{aligned} P_i\omega &:= P_i \enspace, \\ \bot\omega &:= \bot \enspace, \\ (A \to B)\omega &:= A\omega \to B\omega \enspace, \\ (t : B)\omega &:= (t\omega) : (B\omega) \enspace . \end{aligned}$$

Whenever safe, we omit parentheses and write, e.g., $*(t_1, \ldots, t_n)\omega$ instead of $\bigl(*(t_1, \ldots, t_n)\bigr)\omega$.

As an example, let $?$ be a unary operation in the language $\mathcal{L}_1$ and $\omega(?) = \omega_?(x_1)$. Then

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

Fact 2.1.2 (Properties of Operation Translations). *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_i, we have that x_i occurs in $A\omega$ iff x_i occurs in A.

Proof. (1): By induction on the structure of t. The statement is trivial for variables. Let $t = *(t_1, \ldots, t_n)$ for an n-ary $\mathcal{L}_1$-operation $*$, $n \geq 0$. By definition, $\omega(*)$ is an $\mathcal{L}_2$-term $\omega_*(x_1, \ldots, x_n)$. Since $t_1\omega, \ldots, t_n\omega$ are $\mathcal{L}_2$-terms by induction hypothesis, $t\omega = \omega_*(t_1\omega, \ldots, t_n\omega)$ is also an $\mathcal{L}_2$-term.

By induction on the structure of A. The propositional cases are trivial. Let $A = t : B$. As shown above, $t\omega$ is an $\mathcal{L}_2$-term. By induction hypothesis, $B\omega$ is an $\mathcal{L}_2$-formula. Hence, $A\omega = (t\omega) : (B\omega)$ is also an $\mathcal{L}_2$-formula.

(2): By induction on the structure of A. The propositional cases are trivial. If $A = t : B$, then $A^\circ = (t : B)^\circ = \Box B^\circ$ and $(A\omega)^\circ = ((t\omega){:}(B\omega))^\circ = \Box(B\omega)^\circ$, which is the same as $\Box B^\circ$ since, by induction hypothesis, $B^\circ = (B\omega)^\circ$.

(3): It is enough to show by induction on the structure of t that $\mathrm{vars}(t) = \mathrm{vars}(t\omega)$ for an arbitrary term t. If t is a variable x_i, then $x_i\omega = x_i$. If $t = *(t_1, \ldots, t_n)$ for $n \geq 0$, then $\omega(*)$ is a term $\omega_*(x_1, \ldots, x_n)$. By induction hypothesis, $\mathrm{vars}(t_i) = \mathrm{vars}(t_i\omega)$ for $1 \leq i \leq n$. Therefore we have

$$\mathrm{vars}(t\omega) = \mathrm{vars}(\omega_*(t_1\omega, \ldots, t_n\omega)) = \bigcup_{i=1}^{n} \mathrm{vars}(t_i\omega) = \bigcup_{i=1}^{n} \mathrm{vars}(t_i) = \\ \mathrm{vars}(*(t_1, \ldots, t_n)) = \mathrm{vars}(t) \ .$$

□

Definition 2.1.3 (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

$$\mathsf{JL}_1 \vdash A \quad \Longrightarrow \quad \mathsf{JL}_2 \vdash A\omega$$

for any $\mathcal{L}_1$-formula A. We say that JL_1 and JL_2 are *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 lemma, equivalent logics realize the same modal logic.

Lemma 2.1.4 (Equivalence and Forgetful Projection)**.** *Let JL_1 and JL_2 be justification logics over languages $\mathcal{L}_1$ and $\mathcal{L}_2$ respectively. Then $\mathsf{JL}_1 \equiv \mathsf{JL}_2$ implies $(\mathsf{JL}_1)^\circ = (\mathsf{JL}_2)^\circ$.* [1]

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 2.1.2 (1), $A\omega$ is an $\mathcal{L}_2$-formula. By definition of the embedding relation, $\mathsf{JL}_2 \vdash A\omega$. By Fact 2.1.2 (2), $(A\omega)^\circ = A^\circ = B$. Hence, $B \in (\mathsf{JL}_2)^\circ$. □

Definition 2.1.5 (Identity Operation Translation)**.** The identity operation translation ω^{id} (for a language $\mathcal{L}$) is the operation translation that for each $n \geq 0$, maps every n-ary $\mathcal{L}$-operation $*$ to $*(x_1, \ldots, x_n)$.

Lemma 2.1.6 (Identity Operation Translation)**.** *Let ω^{id} be the identity operation translation for a language $\mathcal{L}$. Then for every $\mathcal{L}$-term t and $\mathcal{L}$-formula A, $t = t\omega^{id}$ and $A = A\omega^{id}$.*

Proof. By straightforward inductions on the structure of an $\mathcal{L}_1$-term t and of an $\mathcal{L}_1$-formula A respectively. □

Lemma 2.1.7 (Extension and Embedding)**.** *Let JL_1 and JL_2 be justification logics with $\mathsf{JL}_1 \subseteq \mathsf{JL}_2$. Then $\mathsf{JL}_1 \mathrel{\tilde{\subseteq}} \mathsf{JL}_2$.*

Proof. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be the languages of JL_1 and JL_2 respectively. Let ω^{id} be the *identity operation translation* for $\mathcal{L}_1$. We have $\mathcal{L}_1 \subseteq \mathcal{L}_2$ and, therefore, ω^{id} is an operation translation from $\mathcal{L}_1$ to $\mathcal{L}_2$. By Lemma 2.1.6, 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.

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

Lemma 2.1.8 (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 $\sigma'(x_i) := \sigma(x_i)\omega$ for any variable x_i.*

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

$$\bigl(*(t_1, \dots, t_n)\sigma\bigr)\omega \;=\; *(t_1\sigma, \dots, t_n\sigma)\omega \;=\; \omega_*\bigl((t_1\sigma)\omega, \dots, (t_n\sigma)\omega\bigr) \;.$$

By induction hypothesis, this is the same as

$$\begin{aligned} \omega_*\bigl((t_1\omega)\sigma', \dots, (t_n\omega)\sigma'\bigr) \;&=\; \bigl(\omega_*(t_1\omega, \dots, t_n\omega)\bigr)\sigma' \\ &=\; \bigl(*(t_1, \dots, t_n)\omega\bigr)\sigma' \;. \end{aligned}$$

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

Recall that a preorder (also called quasi-order) is a binary relation that is reflexive and transitive.

Lemma 2.1.9 (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 2.1.7 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 $\mathsf{JL}_1 \,\tilde{\subseteq}\, \mathsf{JL}_2$ and $\mathsf{JL}_2 \,\tilde{\subseteq}\, \mathsf{JL}_3$ respectively. We show $\mathsf{JL}_1 \,\tilde{\subseteq}\, \mathsf{JL}_3$. 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 $x_1, \dots, x_n$ as its only variables, it follows from Facts 2.1.2 (1) and 2.1.2 (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_i, then $(x_i\omega')\omega'' = x_i\omega'' = x_i = x_i\omega$.

Let $t = *(t_1, \ldots, t_n)$ for some n-ary $\mathcal{L}_1$-operation $*$, $n \geq 0$. Then

$$\bigl(*(t_1, \ldots, t_n)\omega'\bigr)\omega'' \;=\; \bigl(\omega'_*(t_1\omega', \ldots, t_n\omega')\bigr)\omega'' \;;$$

in other words, for the $\mathcal{L}_2$-substitution

$$\sigma := \{x_i \mapsto t_i\omega' \mid 1 \leq i \leq n\} \cup \{x_i \mapsto x_i \mid i > n\}$$

we have

$$\bigl(*(t_1, \ldots, t_n)\omega'\bigr)\omega'' \;=\; \bigl(\omega'(*)\sigma\bigr)\omega'' \;. \tag{2.1}$$

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

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

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 $\omega(*)\sigma' = \bigl(\omega'(*)\omega''\bigr)\sigma'$. Altogether,

$$*\,(t_1, \ldots, t_n)\omega \;=\; \bigl(\omega'(*)\omega''\bigr)\sigma' \;. \tag{2.2}$$

Note that $\sigma'(x_i) = \sigma(x_i)\omega''$ for any variable x_i. Therefore, it follows by Lemma 2.1.8 that $\bigl(\omega'(*)\sigma\bigr)\omega'' = \bigl(\omega'(*)\omega''\bigr)\sigma'$ and, by (2.1) and (2.2),

$$\bigl(*(t_1, \ldots, t_n)\omega'\bigr)\omega'' = *(t_1, \ldots, t_n)\omega \;.$$

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. Axiom schemas (formula schemas in general) and constants play a fundamental role in this respect. It is convenient to write formula schemas in the form of their *formula representation*:

Definition 2.1.10 (Formula Representation of Schemas)**.** Let $\mathcal{L}_1$ be a justification language.

(1) 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.

(2) 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.

(3) For a justification logic JL over 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.

(4) 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 $S\omega$.

For the rest of this chapter we write the justification axiom schemas from Figures 1.5 and 1.6 using their formula representations. For instance, the axiom schema $\mathsf{j4}$ 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 t over terms and a variable A over formulas.

Fact 2.1.11 (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 $\mathcal{L}_1$ that satisfies the Substitution Property, 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 $S\omega$.

Proof. (1): If S is provable in JL, by the Substitution Lemma 1.5.9, so is every instance $A(t_1, \ldots, t_n, B_1, \ldots, B_k)$ of S. The converse is trivial since $A(x_1, \ldots, x_n, P_1, \ldots, P_k)$ is itself an instance of S.

(2): By Fact 2.1.2 (3), the $\mathcal{L}_2$-formula $A(x_1, \ldots, x_n, P_1, \ldots, P_k)\omega$ that represents the schema $S\omega$ has all variables and propositions indicated.

Therefore, it can be written as $B(x_1, \ldots, x_n, P_1, \ldots, P_k)$. For any instance $A(t_1, \ldots, t_n, B_1, \ldots, B_k)$ of S the formula

$$A(t_1, \ldots, t_n, B_1, \ldots, B_k)\omega \;=\; B(t_1\omega, \ldots, t_n\omega, B_1\omega, \ldots, B_k\omega)$$

is clearly an instance of $S\omega$. $\square$

We now finally explain why constants are assigned levels. Suppose, as an example, we want to embed JTB4 in JDB4 and that constants do not have levels. Then, for a single constant c

$$\underbrace{c : c : \ldots : c}_{m} : (x_1 : P_1 \to P_1)$$

would be provable in JTB4 for all $m > 0$. Note that the above is not provable in JDB4 because $x_1 : P_1 \to P_1$ is not an axiom in JDB4 (it is, however, provable in JDB4 as a theorem; cf. Lemma 2.2.6). Showing that JTB4 embeds in JDB4 would thus require us, according to Definition 2.1.3, to provide an operation translation ω that, in particular, maps c to some ground term p such that

$$\underbrace{p : p : \ldots : p}_{m} : (x_1 : P_1 \to P_1) \tag{2.3}$$

is provable in JDB4 for all $m > 0$. It can be shown (cf. Lemma 2.3.3) that for a particular m there exists a term p_m that satisfies (2.3). However, because terms are finite objects, there is no term p that satisfies (2.3) for *every* number m. As a consequence, the desired ω does not exist. The assignment of levels to constants is a solution to this problem because it enables us to map constants of different levels to different ground terms.

Remark 2.1.12 (Levels and Constant Specifications)**.** Even though Fitting's definition of embedding from [Fit08] is less general than ours, the problem just described still potentially occurs. Fitting avoided it by imposing *injective* constant specifications: at most one formula is justified by each (sequence of) constant(s). Note that—strictly speaking—our solution of assigning levels to constants can also be viewed as a restriction on the usage of constants, i.e., a constant specification. But the assignment of levels is less restrictive than injective constant specifications.

Recall that an inference rule is called *admissible* in a logic L provided that for any instance of the rule, if its premises are provable in L, then so is its conclusion.

Lemma 2.1.13 (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 1.5.4 and Substitution Lemma 1.5.9, 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.*[2] *Let $S_1, \dots, S_n$ be $\mathcal{L}$-formula schemas provable in JL. There exists an infinite sequence of ground $\mathcal{L}$-terms $p_1, p_2, \dots$ such that for any $m > 0$ and for any $\mathcal{L}$-instance A of one of S_i, $1 \le i \le n$,*

$$\mathsf{JL} \vdash p_m : p_{m-1} : \ldots : p_1 : A \enspace .$$

Proof. Let $A_i(x_1, \dots, x_{k_i}, P_1, \dots, P_{l_i})$ be a formula representation of schema S_i, $i = 1, \dots, n$. We show how to construct the term p_j, $j = 1, 2, \dots$ by induction on j. For each $1 \le i \le n$, by Internalization Property 1.5.4, there exists a ground $\mathcal{L}$-term p_i^1 such that

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

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

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

Assume that for some $m > 0$, we have already constructed ground $\mathcal{L}$-terms $p_1, p_2, \dots, p_m$ such that for all $i = 1, \dots, n$

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

We show how to construct p_{m+1}. For each $1 \le i \le n$, by Internalization Property 1.5.4, there exists a ground $\mathcal{L}$-term p_i^{m+1} such that

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

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,

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

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

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, $1 \leq i \leq n$, by the Substitution Lemma 1.5.9. □

Because all the extensions of J have the rule MP, the two sum-axiom schemas, and satisfy both Internalization Property 1.5.4 and Substitution Lemma 1.5.9, we obtain the following

Corollary 2.1.14 (Iterated Internalization of Schemas for the Extensions of J)**.** *Let JL be an extension of J as defined on page 19. Then Lemma 2.1.13 holds for $\mathsf{JL}_{\mathsf{CS}}$, provided CS is an axiomatically appropriate and schematic constant specification for JL.*

Theorem 2.1.15 (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 17), let $\rightarrow$ be one of binary Boolean connectives, and let MP and $\mathsf{iAN}_{\mathsf{CS}}$ be the only rules of JL_1. Let JL_2 and $\mathcal{L}_2$ satisfy all the conditions of Lemma 2.1.13. Assume the following:*

(1) JL_1 is axiomatized by finitely many axiom schemas;

(2) the formula representations of the axiom schemas of JL_1 do not contain constants;[3]

(3) there exists an operation translation ω from $\mathcal{L}_1$ to $\mathcal{L}_2$ such that for every axiom schema 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 axiom schemas 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 2.1.13, 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 $1 \leq i \leq n$,

$$\mathsf{JL}_2 \vdash p_m : p_{m-1} : \ldots : p_1 : B \ . \tag{2.4}$$

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

Let the operation translation ω' be defined as follows:

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

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 2.1.2 (1).

If A is an instance of an axiom schema S_i of JL_1, $1 \leq i \leq n$, then, by Fact 2.1.11 (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 2.1.11 (1).

If A is obtained by the rule $\mathsf{iAN}_{\mathsf{CS}}$, then it is of the form $c_{i_m}^m : \ldots : c_{i_1}^1 : B$, where B is an instance of an axiom schema 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 (2.4), we have $\mathsf{JL}_2 \vdash p_m : \ldots : p_1 : B\omega'$, 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. $\square$

Since conditions (1) and (2) of the above theorem hold for any extension of J, we have

Corollary 2.1.16 (Embedding for the Extensions of J). *Let JL_1 and JL_2 be two extensions of J as defined on page 19 and let CS and CS' be an arbitrary constant specification for JL_1 and an axiomatically appropriate and schematic constant specification for JL_2 respectively. If there exists an operation translation ω from the language of JL_1 to the language of JL_2 such that for every axiom schema S of JL_1, the formula schema $S\omega$ is provable in $\mathsf{JL}_{2\mathsf{CS}'}$, then $\mathsf{JL}_{1\mathsf{CS}}$ embeds in $\mathsf{JL}_{2\mathsf{CS}'}$.*

Additionally, by Corollary 2.1.16 and Lemma 2.1.6:

Corollary 2.1.17 (Equivalence and Constant Specifications). *Let JL be an extension of J and let CS_1 and CS_2 be constant specifica-*

tions for JL *that are axiomatically appropriate and schematic. Then* $\mathsf{JL}_{\mathsf{CS}_1}$ *and* $\mathsf{JL}_{\mathsf{CS}_2}$ *are equivalent.*

2.2 Classification of Logics

We now return to our justification logics that we call the extensions of J. Corollary 2.1.16 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 axiom schemas sum and app are present in all the extensions of J.

We proceed to prove that all the justification counterparts (as defined on page 23) of $\mathsf{KB5}$ are pairwise equivalent and so are those of $\mathsf{S5}$. Lemma 2.2.6 is the main ingredient for the construction of operation translations that witness these embeddings between justification logics. To prove it, we need five auxiliary lemmas, some of which are later used in the realization proof in Section 3.3.

Lemma 2.2.1 (Consistency)**.** *Let* $\mathsf{JL} \supseteq \mathsf{JD}$ *and let* CS *be an arbitrary constant specification for* JL*. For arbitrary terms* t *and* s *and an arbitrary formula* A*,*

$$\mathsf{JL}_{\mathsf{CS}} \vdash t : A \to \neg s : \neg A \ .$$

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 $(s \cdot t) : \bot \to \bot$

$$\mathsf{JL}_{\mathsf{CS}} \vdash t : A \ \to \ s : (A \to \bot) \ \to \ \bot \ ,$$

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

The following lemma provides a uniform realization of the theorem $\Box(A \to B) \to \Box(B \to C) \to \Box(A \to C)$ of K.

Lemma 2.2.2 (Syllogism)**.** *Let* JL *be a justification logic and let* CS *be an axiomatically appropriate and schematic constant specification for* JL*. 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*,*

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

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 1.5.4, there exists a term $\mathrm{syl}(x_1, x_2)$ such that for arbitrary terms t_1 and t_2,

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

The desired result now follows from the Substitution Lemma 1.5.9. Note that $\mathrm{syl}(x_1, x_2)$ does not depend on t_1, t_2, A, B, or C. □

Lemma 2.2.3 (Internalized Factivity)**.** *Let* $\mathsf{JL} \supseteq \mathsf{J5}$ *and let* CS *be an axiomatically appropriate and schematic constant specification for* JL*. There exists a term* $\mathrm{fact}(x_1)$ *such that for any term* s *and any justification formula* A*,*

$$\mathsf{JL}_{\mathsf{CS}} \vdash \mathrm{fact}(s) : (s : A \to A) \ .$$

Proof. From the propositional tautology $P_1 \to P_2 \to P_1$, by Internalization Property 1.5.4, there exists a term $t_1(x_1)$ such that $\mathsf{JL}_{\mathsf{CS}} \vdash s : P_1 \to t_1(s) : (P_2 \to P_1)$ for any term s. Hence, by the Substitution Lemma 1.5.9, for any formula A,

$$\mathsf{JL}_{\mathsf{CS}} \vdash s : A \ \to \ t_1(s) : (s : A \to A) \ . \tag{2.5}$$

Similarly, for $\neg P_2 \to P_2 \to P_1$, there exists a term $t_2(x_1)$ such that

$$\mathsf{JL}_{\mathsf{CS}} \vdash ?\, s : \neg P_2 \to t_2(?\, s) : (P_2 \to P_1)$$

for any term s. By the Substitution Lemma 1.5.9,

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

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

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

From this, (2.5), and sum, we have $\mathsf{JL}_{\mathsf{CS}} \vdash \mathrm{fact}(s) : (s : A \to A)$ 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. □

The following auxiliary lemma is used in the proofs of Lemmas 2.2.5 and 2.2.6.

Lemma 2.2.4 (Inverse to Negative Introspection, Internalized). *Let* $\mathsf{JL} \supseteq \mathsf{J5}$ *and let* CS *be an axiomatically appropriate and schematic constant specification for* JL*. There exists a term* $\mathrm{invnegint}(x_1)$ *such that for arbitrary terms* t *and* s *and for any justification formula* A*,*

$$\mathsf{JL}_{\mathsf{CS}} \vdash s : \neg\, ?\, t : \neg t : A \;\to\; \mathrm{invnegint}(s) : t : A \;\;.$$

Proof. By a propositional tautology and Internalization Property 1.5.4 there exists a ground term p such that

$$\mathsf{JL}_{\mathsf{CS}} \vdash p : \big((\neg x_2 : P_1 \to ?\, x_2 : \neg x_2 : P_1) \to \neg\, ?\, x_2 : \neg x_2 : P_1 \to x_2 : P_1 \big) \;\;.$$

Since CS is axiomatically appropriate, by $\mathsf{j5}$ and $\mathsf{iAN}_{\mathsf{CS}}$,

$$\mathsf{JL}_{\mathsf{CS}} \vdash c_j^1 : \big(\neg x_2 : P_1 \to ?\, x_2 : \neg x_2 : P_1 \big)$$

for some constant c_j^1 of level 1. Therefore, by app and MP,

$$\mathsf{JL}_{\mathsf{CS}} \vdash (p \cdot c_j^1) : (\neg\, ?\, x_2 : \neg x_2 : P_1 \;\to\; x_2 : P_1) \;\;.$$

Also by app and MP,

$$\mathsf{JL}_{\mathsf{CS}} \vdash x_1 : \neg\, ?\, x_2 : \neg x_2 : P_1 \;\to\; (p \cdot c_j^1 \cdot x_1) : x_2 : P_1 \;\;.$$

The statement of the lemma for $\mathrm{invnegint}(x_1) := p \cdot c_j^1 \cdot x_1$ now follows from the Substitution Lemma 1.5.9. Note that $\mathrm{invnegint}(x_1)$ does not depend on t, s, or A. □

Lemma 2.2.5 (Internalized Positive Introspection). *Let* $\mathsf{JL} \supseteq \mathsf{J5}$ *and let* CS *be an axiomatically appropriate and schematic constant specification for* JL*. There exist terms* $\mathrm{posint}(x_1)$ *and* $\mathrm{t}_!(x_1)$ *such that for any term* s *and any justification formula* A*,*

$$\mathsf{JL}_{\mathsf{CS}} \vdash \mathrm{posint}(s) : \big(s : A \;\to\; \mathrm{t}_!(s) : s : A \big) \;\;.$$

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

$$\mathsf{JL}_{\mathsf{CS}} \vdash s(t) : (A \;\to\; ?\, t : \neg t : \neg A) \;\;. \tag{2.6}$$

By a propositional tautology and Internalization Property 1.5.4 there exists a ground term p such that

$$\mathsf{JL}_{\mathsf{CS}} \vdash p : \big((x_1 : \neg P_1 \to \neg P_1) \;\to\; P_1 \;\to\; \neg x_1 : \neg P_1 \big) \;\;.$$

By Lemma 2.2.3, for the term $\mathrm{fact}(x_1)$ constructed there, $\mathsf{JL_{CS}} \vdash \mathrm{fact}(x_1) : (x_1 : \neg P_1 \to \neg P_1)$. By app and MP,

$$\mathsf{JL_{CS}} \vdash \big(p \cdot \mathrm{fact}(x_1)\big) : (P_1 \ \to \ \neg x_1 : \neg P_1) \ .$$

Since CS is axiomatically appropriate, for some constant c_i^1 of level 1, $c_i^1 : (\neg x_1 : \neg P_1 \to ?\, x_1 : \neg x_1 : \neg P_1)$ is provable in $\mathsf{JL_{CS}}$ by $\mathsf{j5}$ and $\mathsf{iAN_{CS}}$. Hence, by Lemma 2.2.2,

$$\mathsf{JL_{CS}} \vdash \mathrm{syl}\big(p \cdot \mathrm{fact}(x_1), c_i^1\big) : (P_1 \ \to \ ?\, x_1 : \neg x_1 : \neg P_1) \ .$$

Now (2.6) follows from the Substitution Lemma 1.5.9 for $s(x_1) := \mathrm{syl}\big(p \cdot \mathrm{fact}(x_1), c_i^1\big)$.

By Lemma 2.2.4, for the term $\mathrm{invnegint}(x_1)$ constructed there,

$$\mathsf{JL_{CS}} \vdash ?\,?\, x_1 : \neg\, ?\, x_1 : \neg x_1 : P_1 \ \to \ \mathrm{invnegint}(?\,?\, x_1) : x_1 : P_1 \ .$$

Then, by Internalization Property 1.5.4, there exists a ground term p_2 such that

$$\mathsf{JL_{CS}} \vdash p_2 : \big(?\,?\, x_1 : \neg\, ?\, x_1 : \neg x_1 : P_1 \ \to \ \mathrm{invnegint}(?\,?\, x_1) : x_1 : P_1\big) \ .$$

By (2.6), for $t = ?\, x_1$ and $A = x_1 : P_1$,

$$\mathsf{JL_{CS}} \vdash s(?\, x_1) : (x_1 : P_1 \ \to \ ?\,?\, x_1 : \neg\, ?\, x_1 : \neg x_1 : P_1) \ .$$

Hence, by Lemma 2.2.2,

$$\mathsf{JL_{CS}} \vdash \mathrm{syl}\big(s(?\, x_1), p_2\big) : \big(x_1 : P_1 \ \to \ \mathrm{invnegint}(?\,?\, x_1) : x_1 : P_1\big) \ .$$

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 1.5.9. Note that $\mathrm{t}_!(x_1)$ and $\mathrm{posint}(x_1)$ depend on neither s nor A. $\square$

Lemma 2.2.6 (Operation Replacement)**.** *Let* CS_1 *be an arbitrary constant specification for* $\mathsf{JT5}$ *and let* CS_2*,* CS_3*,* CS_4*,* CS_5*, and* CS_6 *be constant specifications for* $\mathsf{JT5}$*,* $\mathsf{JB5}$*,* $\mathsf{JB4}$*,* $\mathsf{JDB4}$*, and* $\mathsf{JDB5}$ *respectively that are axiomatically appropriate and schematic. Then there exist an* $\mathcal{L}(?, \bar{?})$*-term* $\mathrm{t}'_!(x_1)$ *and an* $\mathcal{L}(!, \bar{?})$*-term* $\mathrm{t}_?(x_1)$ *such that*

(1) $\mathsf{JT5}_{\mathsf{CS}_1} \vdash A \to ?\, s : \neg s : \neg A$*;*

(2) $\mathsf{JT5}_{\mathsf{CS}_2} \vdash s : A \to \mathrm{t}_!(s) : s : A$*;*

(3) $\mathsf{JB5}_{\mathsf{CS}_3} \vdash s : A \to \mathrm{t}'_!(s) : s : A$*;*

(4) $\mathsf{JB4}_{\mathsf{CS}_4} \vdash \neg s : A \to \mathrm{t}_?(s) : \neg s : A$;

(5) $\mathsf{JDB4}_{\mathsf{CS}_5} \vdash s : A \to A$;

(6) $\mathsf{JDB5}_{\mathsf{CS}_6} \vdash s : A \to A$,

where $\mathrm{t}_!(x_1)$ *is the* $\mathcal{L}(?)$*-term constructed in Lemma 2.2.5, and* s *and* A *denote an arbitrary term and formula respectively, in each case in the language of the given logic.*

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 2.2.5, for the terms $\mathrm{posint}(x_1)$ and $\mathrm{t}_!(x_1)$ constructed there,

$$\mathsf{JT5}_{\mathsf{CS}_2} \vdash \mathrm{posint}(s) : \bigl(s : A \to \mathrm{t}_!(s) : s : A\bigr) \ .$$

The desired statement now follows from an instance of jt.

(3) By Lemma 2.2.4, for the ground term $\mathrm{invnegint}(x_1)$ constructed there,

$$\mathsf{JB5}_{\mathsf{CS}_3} \vdash \bar{?}\, ?\, x_1 : \neg\, ?\, x_1 : \neg x_1 : P_1 \ \to\ \mathrm{invnegint}(\bar{?}\, ?\, x_1) : x_1 : P_1 \ .$$

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 $\mathsf{JB5}_{\mathsf{CS}_3} \vdash x_1 : P_1 \to \mathrm{t}'_!(x_1) : x_1 : P_1$ by syllogism. The desired statement now follows by the Substitution Lemma 1.5.9. Note that $\mathrm{t}'_!(x_1)$ depends on neither s nor A.

(4) By a propositional tautology and Internalization Property 1.5.4, there exists a ground term p_1 such that $\mathsf{JB4}_{\mathsf{CS}_4} \vdash p_1 : (x_1 : P_1 \to \neg\neg x_1 : P_1)$. By the axiom schema app,

$$\mathsf{JB4}_{\mathsf{CS}_4} \vdash\ !\, x_1 : x_1 : P_1 \ \to\ (p_1 \cdot !\, x_1) : \neg\neg x_1 : P_1 \ .$$

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

$$\mathsf{JB4}_{\mathsf{CS}_4} \vdash x_1 : P_1 \ \to\ (p_1 \cdot !\, x_1) : \neg\neg x_1 : P_1 \ .$$

By contraposition and Internalization Property 1.5.4, there exists a term $t_1(x_1)$ such that

$$\mathsf{JB4}_{\mathsf{CS}_4} \vdash \bar{?}(p_1 \cdot !\, x_1) : \neg(p_1 \cdot !\, x_1) : \neg\neg x_1 : P_1 \ \to\ t_1\bigl(\bar{?}(p_1 \cdot !\, x_1)\bigr) : \neg x_1 : P_1 \ .$$

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 $\mathsf{JB4}_{\mathsf{CS}_4} \vdash \neg x_1 : P_1 \to \mathrm{t}_?(x_1) : \neg x_1 : P_1$ for

$$\mathrm{t}_?(x_1) := t_1\big(\bar{?}(p_1 \cdot !\, x_1)\big) \ .$$

The desired statement now follows by the Substitution Lemma 1.5.9. 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 1.5.4, there exists a term $t_1(x_1)$ such that

$$\mathsf{JDB4}_{\mathsf{CS}_5} \vdash x_1 : P_1 \ \to \ t_1(x_1) : \neg\neg P_1 \ .$$

By contraposition and Internalization Property 1.5.4, there exists a term $t_2(x_1)$ such that

$$\mathsf{JDB4}_{\mathsf{CS}_5} \vdash x_2 : \neg t_1(x_1) : \neg\neg P_1 \ \to \ t_2(x_2) : \neg x_1 : P_1 \ .$$

Again by contraposition,

$$\mathsf{JDB4}_{\mathsf{CS}_5} \vdash \neg t_2(x_2) : \neg x_1 : P_1 \ \to \ \neg x_2 : \neg t_1(x_1) : \neg\neg P_1 \ . \tag{2.7}$$

By Lemma 2.2.1, $\mathsf{JDB4}_{\mathsf{CS}_5} \vdash !\, x_1 : x_1 : P_1 \ \to \ \neg t_2(x_2) : \neg x_1 : P_1$. By the j4-instance $x_1 : P_1 \to !\, x_1 : x_1 : P_1$ and syllogism,

$$\mathsf{JDB4}_{\mathsf{CS}_5} \vdash x_1 : P_1 \ \to \ \neg t_2(x_2) : \neg x_1 : P_1 \ .$$

By syllogism with (2.7),

$$\mathsf{JDB4}_{\mathsf{CS}_5} \vdash x_1 : P_1 \ \to \ \neg x_2 : \neg t_1(x_1) : \neg\neg P_1 \ .$$

By the Substitution Lemma 1.5.9,

$$\mathsf{JDB4}_{\mathsf{CS}_5} \vdash x_1 : P_1 \ \to \ \neg\, \bar{?}\, t_1(x_1) : \neg t_1(x_1) : \neg\neg P_1 \ .$$

It follows by the contrapositive $\neg\, \bar{?}\, t_1(x_1) : \neg t_1(x_1) : \neg\neg P_1 \to P_1$ of a jb-instance and by syllogism that $\mathsf{JDB4}_{\mathsf{CS}_5} \vdash x_1 : P_1 \to P_1$. The desired statement now follows by the Substitution Lemma 1.5.9.

(6) By repeating the steps that lead to (2.7) in the proof of 2.2.6 (5),

$$\mathsf{JDB5}_{\mathsf{CS}_6} \vdash \neg t_2(x_2) : \neg x_1 : P_1 \ \to \ \neg x_2 : \neg t_1(x_1) : \neg\neg P_1 \tag{2.8}$$

for some terms $t_1(x_1)$ and $t_2(x_1)$. Let

$$\mathsf{CS}_6^- := \mathsf{CS}_6 \setminus \{c_{i_n}^n : \ldots : c_{i_1}^1 : A \mid A \text{ is an instance of } \mathsf{jd}\} \ .$$

Since CS_6 is axiomatically appropriate and schematic, $\mathsf{CS}_6^- \subseteq \mathsf{CS}_6$ is an axiomatically appropriate and schematic constant specification for $\mathsf{JB5}$. Therefore, by Lemma 2.2.6 (3), for the term $\mathrm{t}_!'(x_1)$ constructed there,

$$\mathsf{JB5}_{\mathsf{CS}_6^-} \vdash x_1 : P_1 \ \rightarrow \ \mathrm{t}_!'(x_1) : x_1 : P_1 \ .$$

Since $\mathsf{JB5}_{\mathsf{CS}_6^-} \subseteq \mathsf{JDB5}_{\mathsf{CS}_6}$, the same formula is provable in $\mathsf{JDB5}_{\mathsf{CS}_6}$. By Lemma 2.2.1,

$$\mathsf{JDB5}_{\mathsf{CS}_6} \vdash \mathrm{t}_!'(x_1) : x_1 : P_1 \ \rightarrow \ \neg t_2(x_2) : \neg x_1 : P_1 \ .$$

By syllogism,

$$\mathsf{JDB5}_{\mathsf{CS}_6} \vdash x_1 : P_1 \ \rightarrow \ \neg t_2(x_2) : \neg x_1 : P_1 \ .$$

By syllogism with (2.8),

$$\mathsf{JDB5}_{\mathsf{CS}_6} \vdash x_1 : P_1 \ \rightarrow \ \neg x_2 : \neg t_1(x_1) : \neg\neg P_1 \ .$$

It remains to repeat the final steps of the proof of Lemma 2.2.6 (5). □

Remark 2.2.7 (Uniformity of Terms)**.** Note that the constants occurring in the terms constructed in Lemma 2.2.6 depend on the constant specifications used. For example, although the term $\mathrm{t}_!(x_1)$ works in a uniform way for a given constant specification CS—i.e., $\mathsf{JT5}_{\mathsf{CS}} \vdash s : A \to \mathrm{t}_!(s) : s : A$ for arbitrary s and A—the constants in $\mathrm{t}_!(x_1)$ might change if a different constant specification is used.

Remark 2.2.8 (Why $\bar{?}$ is not ?)**.** In [BGK10], a single operation ? was used to formulate both the axiom schemas $\mathsf{j5}$ and jb. This decision was motivated by a desire to minimize the number of operations on terms. It was possible to use ? to realize the modal axiom schema b in $\mathsf{JT5}$ because $A \to ?\, s{:}\neg s{:}\neg A$ is provable in $\mathsf{JT5}$ (cf. Lemma 2.2.6 (1)). Hence, JB embeds in $\mathsf{JT5}$ by an operation translation that replaces $\bar{?}$ with ?. However, the same operation translation does not embed JB in $\mathsf{J5}$; nor does the inverse operation translation that replaces ? with $\bar{?}$ embed $\mathsf{J5}$ in JB. In fact, no operation translation embeds JB in $\mathsf{J5}$ or $\mathsf{J5}$ in JB. Indeed, if $\mathsf{JB} \mathrel{\tilde{\subseteq}} \mathsf{J5}$, then, by the proof of Lemma 2.1.4, $\mathsf{JB}^\circ \subseteq \mathsf{J5}^\circ$ and, by the Realization Theorem (to be proved in Section 3.3), $\mathsf{KB} \subseteq$

K5, which is not the case since the modal axiom schema 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 2.2.9 (Equivalences)**.**

(1) $\mathsf{JB4} \equiv \mathsf{JB5} \equiv \mathsf{JB45}$.

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

The above equivalences also hold if the logics are restricted to axiomatically appropriate and schematic constant specifications, e.g., $\mathsf{JB4}_{\mathsf{CS}} \equiv \mathsf{JB5}_{\mathsf{CS}'}$ *for constant specifications* CS *and* CS' *for* JB4 *and* JB5 *respectively, both axiomatically appropriate and schematic.*

Proof. We prove the theorem for logics with total constant specifications. The lemma for logics with restricted constant specifications then follows from Corollary 2.1.17 and the fact that $\equiv$ is an equivalence relation by Lemma 2.1.9.

To show each embedding, according to Corollary 2.1.16, it is sufficient to provide an operation translation ω such that for every axiom schema S of one logic, the formula schema $S\omega$ is provable in the other. In the following proof, we provide such an ω—based on the terms from Lemma 2.2.6—for each embedding.

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

$\mathsf{JB4} \mathrel{\tilde{\subseteq}} \mathsf{JB5}$: Let $\omega^{!\text{-elim}}$ agree with the identity operation translation for $\mathcal{L}(\bar{?}, !)$ (cf. Definition 2.1.5), except that $\omega^{!\text{-elim}}(!) := \mathrm{t}'_!(x_1)$ for the term $\mathrm{t}'_!(x_1)$ that exists by Lemma 2.2.6 (3). Note that $\omega^{!\text{-elim}}$ is indeed an operation translation from $\mathcal{L}(\bar{?}, !)$ to $\mathcal{L}(\bar{?}, ?)$. Since each axiom schema S of JB4, except for j4, is also an axiom schema of JB5 and since its formula representation does not contain !, $S\omega^{!\text{-elim}} = S$ is provable in JB5. For the only remaining axiom schema, j4,

$$(x_1 : P_1 \ \rightarrow \ !\,x_1 : x_1 : P_1)\omega^{!\text{-elim}} = \ x_1 : P_1 \ \rightarrow \ \mathrm{t}'_!(x_1) : x_1 : P_1 \ , \quad (2.9)$$

which is provable in JB5 by Lemma 2.2.6 (3).

$\mathsf{JB5} \mathrel{\tilde{\subseteq}} \mathsf{JB45}$: The identity operation translation ω^{id} for $\mathcal{L}(\bar{?}, ?)$ witnesses the embedding. Indeed, since for each axiom schema S of JB5, we have $S\omega^{id} = S$, it remains to note that all axiom schemas of $\mathsf{JB5}$ remain axiom schemas in $\mathsf{JB45}$.

$\mathsf{JB45} \mathrel{\tilde{\subseteq}} \mathsf{JB4}$: Let $\omega^{?\text{-elim}}$ agree with the identity operation translation for $\mathcal{L}(\bar{?}, !, ?)$, except that $\omega^{?\text{-elim}}(?) := \mathrm{t}_?(x_1)$ for the term $\mathrm{t}_?(x_1)$ that exists by Lemma 2.2.6 (4). Note that $\omega^{?\text{-elim}}$ is indeed an operation translation from $\mathcal{L}(\bar{?}, !, ?)$ to $\mathcal{L}(\bar{?}, !)$. Since each axiom schema S of $\mathsf{JB45}$, except for $\mathsf{j5}$, is also an axiom schema of $\mathsf{JB4}$ and since its formula representation does not contain ?, $S\omega^{?\text{-elim}} = S$ is provable in $\mathsf{JB4}$. For the only remaining axiom schema, $\mathsf{j5}$,

$$(\neg x_1 : P_1 \to ?\, x_1 : \neg x_1 : P_1)\omega^{?\text{-elim}} = \\ \neg x_1 : P_1 \to \mathrm{t}_?(x_1) : \neg x_1 : P_1 \quad , \tag{2.10}$$

which is provable in $\mathsf{JB4}$ by Lemma 2.2.6 (4).

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

$$\mathsf{JT5} \mathrel{\tilde{\subseteq}} \mathsf{JT45} \mathrel{\tilde{\subseteq}} \mathsf{JTB45} \mathrel{\tilde{\subseteq}} \mathsf{JTB4} \mathrel{\tilde{\subseteq}} \mathsf{JDB4} \\ \mathrel{\tilde{\subseteq}} \mathsf{JDB45} \mathrel{\tilde{\subseteq}} \mathsf{JDB5} \mathrel{\tilde{\subseteq}} \mathsf{JTB5} \mathrel{\tilde{\subseteq}} \mathsf{JT5}_{\mathsf{CS}_1} \quad .$$

Among these embeddings, four are trivially witnessed—as in the embedding $\mathsf{JB5} \mathrel{\tilde{\subseteq}} \mathsf{JB45}$ above—by identity operation translations:

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

We now prove the remaining four embeddings.

$\mathsf{JTB45} \mathrel{\tilde{\subseteq}} \mathsf{JTB4}$: Let $\omega^{?\text{-elim}}$ agree with the identity operation translation for $\mathcal{L}(\bar{?}, !, ?)$, except that $\omega^{?\text{-elim}}(?) := \mathrm{t}_?(x_1)$ for the term $\mathrm{t}_?(x_1)$ that exists by Lemma 2.2.6 (4). As in the case of $\mathsf{JB45} \mathrel{\tilde{\subseteq}} \mathsf{JB4}$, all the axiom schemas of $\mathsf{JTB45}$, except for $\mathsf{j5}$, remain axiom schemas in $\mathsf{JTB4}$ and their formula representations do not contain ?. As noted above (cf. (2.10)), $\mathsf{j5}\omega^{?\text{-elim}}$ is provable in $\mathsf{JB4}$ and, hence, in $\mathsf{JTB4}$.

$\mathsf{JTB4} \mathrel{\tilde{\subseteq}} \mathsf{JDB4}$: The identity operation translation ω^{id} for $\mathcal{L}(\bar{?}, !)$ witnesses the embedding. Indeed, since for each axiom schema S of $\mathsf{JTB4}$, we have $S\omega^{id} = S$, it remains to note that all but one axiom schema of $\mathsf{JTB4}$ remain axiom schemas in $\mathsf{JDB4}$. The only remaining axiom schema, jt, with a formula representation $x_1 : P_1 \to P_1$, is provable in $\mathsf{JDB4}$ by Lemma 2.2.6 (5).

$\mathsf{JDB45} \mathrel{\tilde{\subseteq}} \mathsf{JDB5}$: Let $\omega^{!\text{-elim}}$ agree with the identity operation translation for $\mathcal{L}(\bar{?}, !)$, except that $\omega^{!\text{-elim}}(!) := \mathrm{t}'_!(x_1)$, for the term $\mathrm{t}'_!(x_1)$ that exists by Lemma 2.2.6 (3). As in the case of $\mathsf{JB4} \mathrel{\tilde{\subseteq}} \mathsf{JB5}$, all the axiom schemas of JDB45, except for j4, remain axiom schemas in JDB5 and their formula representations do not contain !. As noted above (cf. (2.9)), $\mathsf{j4}\omega^{!\text{-elim}}$ is provable in JB5 and, hence, in JDB5.

$\mathsf{JTB5} \mathrel{\tilde{\subseteq}} \mathsf{JT5}$: Let $\omega^{\bar{?}\text{-elim}}$ agree with the identity operation translation for $\mathcal{L}(\bar{?}, ?)$, except that $\omega^{\bar{?}\text{-elim}}(\bar{?}) := ?\,x_1$. Since each axiom schema S of JTB5, except for jb, is also an axiom schema of JT5 and since its formula representation does not contain $\bar{?}$, $S\omega^{\bar{?}\text{-elim}} = S$ is provable in JT5. For the only remaining axiom schema, 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 JT5 by Lemma 2.2.6 (1). □

Theorem 2.2.9 allows us to draw a justification analog of the modal cube (cf. Figure 1.1). The Hasse diagram of Figure 2.1 contains all our 24 justification logics; the nodes denote the equivalence classes with respect to $\equiv$. An edge between two nodes means that the logics of the lower left node all embed in every logic of the upper right node. Note that $\tilde{\subseteq}$ is transitive by Lemma 2.1.9 and that for two logics $\mathsf{JL}_1 \subseteq \mathsf{JL}_2$ the embedding $\mathsf{JL}_1 \mathrel{\tilde{\subseteq}} \mathsf{JL}_2$ is witnessed by the identity operation translation (cf. proof of Theorem 2.2.9).

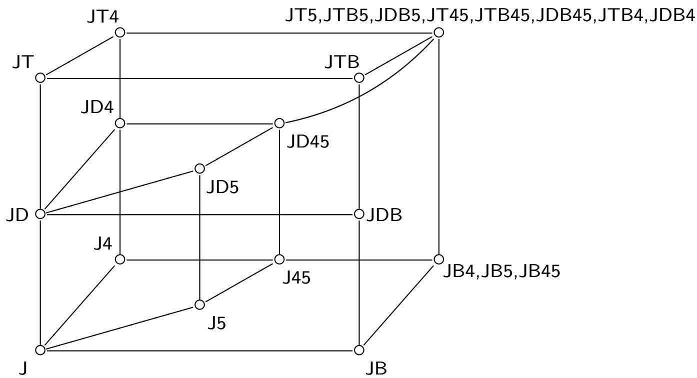

Figure 2.1: The “justification cube”

Using the results of this section, we can restate Lemma 1.5.5 (Lifting) for a bigger class of logics, namely for those logics that contain positive introspection.

Lemma 2.2.10 (Lifting Lemma Extended). *Let* JL *be either one of the logics* $\mathsf{JB5}, \mathsf{JT5}, \mathsf{JTB5}, \mathsf{JDB5}$ *or an extension of* $\mathsf{J4}$. *Let* CS *be a constant specification for* JL *that is axiomatically appropriate (and schematic if* JL *is not an extension of* $\mathsf{J4}$*). If*

$$A_1, \dots, A_n,\ t_1 : B_1, \dots, t_m : B_m \vdash_{\mathsf{JL}_{\mathsf{CS}}} C\ , \tag{2.11}$$

then there exists a term $t(x_1, \dots, x_n, x_{n+1}, \dots, x_{n+m})$ *such that*

$$s_1 : A_1, \dots, s_n : A_n,\ t_1 : B_1, \dots, t_m : B_m \vdash_{\mathsf{JL}_{\mathsf{CS}}} t(s_1, \dots, s_n, t_1, \dots, t_m) : C$$

for all terms $s_1, \dots, s_n$. *The term* t *is ground if* $n = 0$ *and* $m = 0$.

Proof. It is enough to show that $\mathsf{JL}_{\mathsf{CS}}$ contains positive introspection, i.e., that there exists a term $t(x_1)$ such that $\mathsf{JL}_{\mathsf{CS}} \vdash s : A \to t(s) : s : A$, where s and A are arbitrary. Using this schema of positive introspection instead of $\mathsf{j4}$, the lemma then follows by literally repeating the proof of the original Lifting Lemma 1.5.5.

There is nothing to show for extensions of $\mathsf{J4}$.

If JL is $\mathsf{JT5}$, $\mathsf{JL}_{\mathsf{CS}}$ contains positive introspection by Lemma 2.2.6 (2). Assume JL is $\mathsf{JTB5}$. Then

$$\mathsf{CS}^- := \mathsf{CS} \setminus \{c^n_{i_n} : \dots : c^1_{i_1} : A \mid A \text{ is an instance of } \mathsf{jb}\}$$

is an axiomatically appropriate and schematic constant specification for $\mathsf{JT5}$. Therefore, by Lemma 2.2.6 (2), $\mathsf{JT5}_{\mathsf{CS}^-}$ contains positive introspection and, since $\mathsf{JT5}_{\mathsf{CS}^-} \subset \mathsf{JTB5}_{\mathsf{CS}}$, so does $\mathsf{JL}_{\mathsf{CS}}$.

If JL is $\mathsf{JB5}$, $\mathsf{JL}_{\mathsf{CS}}$ contains positive introspection by Lemma 2.2.6 (3). If JL is $\mathsf{JDB5}$, then $\mathsf{JL}_{\mathsf{CS}}$ contains positive introspection by a similar argument as for $\mathsf{JTB5}$. □

2.3 An Alternative: Local Embedding

Theorems 2.1.15 (and, hence, Theorem 2.2.9) depend on the constants being divided into levels (cf. explanations on page 37). These levels can be dropped, but this comes at the price of a weaker definition of embedding and, as a consequence, weaker statements of Theorems 2.1.15 and 2.2.9. In this section, we sketch this alternative approach. Following Fitting [Fit07], the alternative definition of embedding is called *local*: in order to embed a logic in another, instead of providing a global operation translation, it is enough to provide a separate, *local* operation translation for every formula. Local equivalence is defined accordingly. Proving two logics locally equivalent then does not require levels of constants. Lemma 2.3.3 and Theorem 2.3.4 below are the local analogs of Lemma 2.1.13 and Theorem 2.1.15 respectively.

Using Theorem 2.3.4, all the results of Section 2.2 could be reformulated with respect to local embedding instead of embedding and for logics whose constants are not divided into levels—all the proofs could be almost literally adopted. As a consequence, the *modular realization theorem* from Section 3.3 does also hold for logics without levels.

Definition 2.3.1 (Local Embedding and Local 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 *locally embeds* in JL_2, written $\mathsf{JL}_1 \tilde{\subseteq}_{\mathsf{L}} \mathsf{JL}_2$, if for every $\mathcal{L}_1$-formula A there exists an operation translation ω from $\mathcal{L}_1$ to $\mathcal{L}_2$ such that

$$\mathsf{JL}_1 \vdash A \quad \Longrightarrow \quad \mathsf{JL}_2 \vdash A\omega \ .$$

We say that JL_1 and JL_2 are *locally equivalent*, written $\mathsf{JL}_1 \equiv_{\mathsf{L}} \mathsf{JL}_2$, if $\mathsf{JL}_1 \tilde{\subseteq}_{\mathsf{L}} \mathsf{JL}_2$ and $\mathsf{JL}_2 \tilde{\subseteq}_{\mathsf{L}} \mathsf{JL}_1$.

Obviously, two equivalent logics are also locally equivalent (the converse is not true). The proof that $\equiv_{\mathsf{L}}$ is an equivalence relation is similar to the proof of Lemma 2.1.9.

Lemma 2.3.2 ($\equiv_{\mathsf{L}}$ is an Equivalence Relation)**.** *The relation $\tilde{\subseteq}_{\mathsf{L}}$ is a preorder. Accordingly, $\equiv_{\mathsf{L}}$ is an equivalence relation.*

Proof. Since $\tilde{\subseteq}$ is reflexive by Lemma 2.1.9 and $\tilde{\subseteq}_{\mathsf{L}}$ is an extension of $\tilde{\subseteq}$, $\tilde{\subseteq}_{\mathsf{L}}$ is reflexive too.

Let JL_1, JL_2, and JL_3 be justification logics over languages $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_3$ respectively such that $\mathsf{JL}_1 \tilde{\subseteq}_{\mathsf{L}} \mathsf{JL}_2$ and $\mathsf{JL}_2 \tilde{\subseteq}_{\mathsf{L}} \mathsf{JL}_3$. We show $\mathsf{JL}_1 \tilde{\subseteq}_{\mathsf{L}} \mathsf{JL}_3$. Let A be an arbitrary $\mathcal{L}_1$-formula provable in JL_1. By assumption, there exists an operation translation ω' from $\mathcal{L}_1$ to $\mathcal{L}_2$ such that $\mathsf{JL}_2 \vdash A\omega'$. Also by assumption, there exists an operation translation ω'' from $\mathcal{L}_2$ to $\mathcal{L}_3$ such 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 2.1.2 (1) and 2.1.2 (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, the part of the proof of Lemma 2.1.9 that proves $(t\omega')\omega'' = t\omega$ for every $\mathcal{L}_1$-term t can be literally adopted. □

Lemma 2.3.3 (Uniform Iterated Internalization of Schemas)**.** *Let $\mathcal{L}$ be a justification language that has a binary operation $+$ and whose constants are not necessarily divided into levels. Let* JL *be a justification logic over $\mathcal{L}$ that enjoys Internalization Property 1.5.4 and Substitution Lemma 1.5.9, 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. *Let $S_1, \dots, S_n$ be $\mathcal{L}$-formula schemas provable in* JL. *Then for every $m > 0$ there exists a ground term p such that for any $\mathcal{L}$-instance A of one of S_i, $1 \leq i \leq n$ and for any $k \leq m$,*

$$\mathsf{JL} \vdash \underbrace{p : p : \ldots : p}_{k} : A \ .$$

Proof. Let $A_i(x_1, \dots, x_{k_i}, P_1, \dots, P_{l_i})$ be a formula representation of the provable schema S_i, $i = 1, \dots, n$. For each i, by the Internalization Property 1.5.4, there exists a ground $\mathcal{L}$-term p_1^i such that

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

Let $p_1 := p_1^1 + \cdots + p_1^n$. For each $1 \leq i \leq n$, by using appropriate instances of sum and the rule MP,

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

Let $y_1, \dots, y_m$ denote fresh and distinct variables. Let $p_1^+ := p_1 + y_2 + \ldots + y_m$. For each $1 \leq i \leq n$, by using appropriate instances of sum and the rule MP,

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

For each $1 \leq i \leq n$, by the Internalization Lemma 1.5.4, there exists a ground $\mathcal{L}$-term p_2^i such that

$$\mathsf{JL} \vdash p_2^i : p_1^+ : A_i(x_1, \dots, x_{k_i}, P_1, \dots, P_{l_i}) \ .$$

Let $p_2 := p_2^1 + \cdots + p_2^n$. For each $1 \leq i \leq n$, by using appropriate instances of sum and the rule MP,

$$\mathsf{JL} \vdash p_2 : p_1^+ : A_i(x_1, \dots, x_{k_i}, P_1, \dots, P_{l_i}) \ .$$

Let $p_2^+ := y_1 + p_2 + y_3 + \ldots + y_m$. For each $1 \le i \le n$, by using appropriate instances of sum and the rule MP,

$$\mathsf{JL} \vdash p_2^+ : p_1^+ : A_i(x_1, \ldots, x_{k_i}, P_1, \ldots, P_{l_i}) \ .$$

By repeating this procedure, we can construct terms $p_3^+, \ldots, p_m^+$ such that for each $1 \le i \le n$ and for each $k \le m$,

$$\mathsf{JL} \vdash p_k^+ : \ldots : p_2^+ : p_1^+ : A_i(x_1, \ldots, x_{k_i}, P_1, \ldots, P_{l_i}) \ . \tag{2.12}$$

Note that every term p_k^+, $1 \le k \le m$, contains exactly the variables in $\{y_1, \ldots, y_m\} \setminus \{y_k\}$. Let the substitution σ map y_k to p_k, $1 \le k \le m$, and otherwise behave like the identity substitution that maps every variable to itself. It is easy to check that

$$p_1^+\sigma \ = \ \ldots \ = \ p_m^+\sigma \ = \ p_1 + \ldots + p_m \ .$$

Let this term be denoted by p. For each $1 \le i \le n$ and for each $k \le m$, by (2.12) and the Substitution Lemma 1.5.9, we obtain (note that σ has no effect on $A_i(x_1, \ldots, x_{k_i}, P_1, \ldots, P_{l_i})$)

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

Therefore, $\mathsf{JL} \vdash \underbrace{p : p : \ldots : p}_{k} : A$ for every $\mathcal{L}$-instance A of one of S_i, $1 \le i \le n$, by the Substitution Lemma 1.5.9. □

Theorem 2.3.4 (Local Embedding)**.** *Let JL_1 and JL_2 be two justification logics over languages $\mathcal{L}_1$ and $\mathcal{L}_2$ respectively (whose constants are not necessarily divided into levels), let $\rightarrow$ be one of binary Boolean connectives, and let MP and $\mathsf{iAN_{CS}}$ be the only rules of JL_1. Let JL_2 and $\mathcal{L}_2$ satisfy all the conditions of Lemma 2.3.3. Assume the following:*

(1) JL_1 is axiomatized by finitely many axiom schemas;

(2) the formula representations of the axiom schemas 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 schema S of JL_1, the $\mathcal{L}_2$-formula schema $S\omega$ is provable in JL_2.

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

Then JL_1 *locally embeds in* JL_2*.*

Proof. We have to show that for any $\mathcal{L}_1$-formula A there exists an operation translation ω' from $\mathcal{L}_1$ to $\mathcal{L}_2$ such that $\mathsf{JL}_1 \vdash A$ implies $\mathsf{JL}_2 \vdash A\omega'$. So assume $\mathsf{JL}_1 \vdash A$ and let $\mathcal{P}$ be a JL_1-proof of A. Let

$$m := \max\{n \mid c_{i_n} : \ldots : c_{i_1} : B \text{ is a conclusion of } \mathsf{iAN}_{\mathsf{CS}} \text{ in } \mathcal{P}\} \ .$$

Let $S_1, \ldots, S_n$ be the axiom schemas 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 2.3.3, for this m there exists an $\mathcal{L}_2$-term p such that for every $\mathcal{L}_2$-instance C of one of $S_i\omega$, $1 \leq i \leq n$, and for every $k \leq m$,

$$\mathsf{JL}_2 \vdash \underbrace{p : p : \ldots : p}_{k} : C \ . \tag{2.13}$$

Let the operation translation ω' be defined as follows:

$$\omega'(*) := \begin{cases} p & \text{if } * \text{ is an } \mathcal{L}_1\text{-constant } c_i, \\ \omega(*) & \text{otherwise.} \end{cases}$$

Clearly, ω' is an operation translation from $\mathcal{L}_1$ to $\mathcal{L}_2$. Note that p depends on m, which, in turn, depends on the proof $\mathcal{P}$ of A.

We show by induction on $\mathcal{P}$ that $\mathsf{JL}_2 \vdash A\omega'$. Note that $A\omega'$ is an $\mathcal{L}_2$-formula by Fact 2.1.2 (1).

For an instance D of an axiom schema S_i of JL_1, $1 \leq i \leq n$, by Fact 2.1.11 (2), $D\omega'$ is an instance of the $\mathcal{L}_2$-formula schema $S_i\omega'$. The schema $S_i\omega'$ 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, $D\omega'$ is an instance of the provable schema $S_i\omega$ and is itself provable in JL_2 by Fact 2.1.11 (1).

Consider a conclusion $c_{i_k} : \ldots : c_{i_1} : D$ of $\mathsf{iAN}_{\mathsf{CS}}$, where D is an instance of an axiom schema S_i for some $1 \leq i \leq n$. As shown in the previous paragraph, $D\omega'$ is an instance of the formula schema $S_i\omega$. By (2.13),

$$\mathsf{JL}_2 \vdash \underbrace{p : \ldots : p}_{k} : D\omega' \ ,$$

which is the same as $\mathsf{JL}_2 \vdash (c_{i_k} : \ldots : c_{i_1} : D)\omega'$ by definition of ω'.

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

3 Proving Realization Theorems

In this chapter, we develop a constructive method for proving realization theorems, general enough to be applicable to any modal logic captured by a cut-free nested sequent system of context-sharing rules. Recall that nested sequent systems cover more modal logics than ordinary Gentzen systems, e.g., they cover all the logics in the modal cube from Figure 1.1. As a consequence, our realization method is more general than existing methods (see also the introduction of this thesis) and allows for new realization theorems, e.g., for $\mathsf{K5}$.

In Section 3.1, we introduce the basic technical machinery that the method is based on. In addition, we formulate the Merging Theorem by Fitting (Theorem 3.1.6), which plays a major role in our method of realization. In Section 3.2, we introduce the formalism of nested sequents and present our general realization method based on nested sequents. In Section 3.3, we apply this method to the nested sequent systems by Brünnler [Brü09] that capture all the 15 normal modal logics in the modal cube, which gives us a uniform and constructive realization theorem for these logics. In addition, we use the results of Chapter 2 to make this realization theorem modular in the sense that it connects every modal logic to all of its justification counterparts.

Note that, while our method is more powerful than existing methods, it is also more complex. The additional complexity is mainly caused by the more complex structure of nested sequents, which are trees of formulas, as opposed to ordinary Gentzen sequents, which are sequences. Further, because our realization theorem in Section 3.3 covers all the logics in the modal cube, there are many modal rules to realize, essentially one for each axiom schema d, t, b, $\mathsf{4}$, and $\mathsf{5}$.

Recall that modal formulas are given in negation normal form, while justification formulas are given in a traditional format with implication and falsum 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.

3.1 Preliminaries

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. To distinguish between different occurrences of modalities we *annotate* them with different natural numbers, using parity to distinguish between $\Box$'s and $\Diamond$'s, and realize such annotated modal formulas. The annotations are purely syntactic devices and have no semantic meaning. We adopt and adapt Fitting's notation from [Fit09].

Definition 3.1.1 (Annotations)**.** *Annotated modal formulas*, or *annotated formulas* for short, are built according to 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. Given a modal formula A', every annotated formula A is called an *annotated version of* A' provided that A' is obtained from A by dropping all indices on its modalities. An annotated formula is called *properly annotated* if no index occurs twice in it.

We mostly work with properly annotated formulas. Note that, since modal formulas are in negation normal form, every subformula of a (properly) annotated formula is itself (properly) annotated.

Remark 3.1.2 (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 the restriction that the set of even numbers is in one-to-one correspondence with the set of variables. This restriction, the

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

Figure 3.1: Realization of annotated formulas

normality condition, is standard. It 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 3.1.3 (Realization function). A *prerealization function* r (in a justification language $\mathcal{L}$) is a partial mapping from positive natural numbers to $\mathcal{L}$-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 3.1. 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.

When working with realization functions, we usually don't mention the underlying justification language explicitly because it is either irrelevant or given by the context, i.e., by the justification logic we work with.

Definition 3.1.4. Let A be an annotated formula and r be a prerealization function. We define

$$\begin{aligned} \mathrm{vars}_{\Diamond}(A) &:= \{x_k \mid \Diamond_{2k} \text{ occurs in } A\} \ , \\ r \restriction A &:= r \restriction \{i \mid i \text{ occurs in } A \text{ as an index}\} \ , \end{aligned}$$

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

Substitutions (recall Definition 1.5.7) play an important role in our realization procedure. As with (pre)realization functions, we usually do not mention the underlying justification language of a substitution. The definition of domain for substitutions differs from the standard one for ordinary functions such as prerealization functions. The *domain of a substitution* σ is

$$\mathrm{dom}(\sigma) \;:=\; \{x_i \mid \sigma(x_i) \neq x_i\} \;\; .$$

Note that the definition of the domain of a substitution differs from the common definition of the domain of a function (e.g., a realization function). The *variable range of* σ, denoted by $\mathrm{vrange}(\sigma)$, is the set of variables that occur in terms from the set $\{\sigma(x_i) \mid x_i \in \mathrm{dom}(\sigma)\}$.

Composition of substitutions is defined as

$$(\sigma_2 \circ \sigma_1)(x_i) \;:=\; \sigma_1(x_i)\sigma_2$$

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:

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

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 (to be used in the proof of Theorem 3.2.11) shows that, although the process of realizing a modal formula starts with annotating it—which can be done in many different ways—the realizability of the formula does not depend on the chosen annotation.

Lemma 3.1.5 (Renaming of Annotations)**.** *Let* JL *be a justification logic and let* CS *be a schematic constant specification for* JL*. Let* A_1 *and* A_2 *be properly annotated versions of the same modal formula* A' *and let* r_1 *be a realization function on* A_1 *with* $\mathsf{JL}_{\mathsf{CS}} \vdash (A_1)^{r_1}$*. Then there exists a realization function* r_2 *on* A_2 *such that* $\mathsf{JL}_{\mathsf{CS}} \vdash (A_2)^{r_2}$*.*

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

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

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 1.5.9 that $(A_2)^{r_2}$ is provable in $\mathsf{JL}_{\mathsf{CS}}$. 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, we further have $(B_1)^{r_1}\sigma = (B_2)^{r_2}$. 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, $(\Diamond_{2m} B_1)^{r_1}\sigma = \neg x_m\sigma : \neg(B_1)^{r_1}\sigma = \neg x_n : \neg(B_2)^{r_2} = (\Diamond_{2n} B_2)^{r_2}$. $\Box$

Our method for proving realization theorems is by induction on the height of a derivation in a nested sequent system for a modal logic. Since in different premises of branching rules, realizations of side formulas may not coincide, these realizations need to be reconciled, which is done using Fitting's merging technique. The merging theorem below is an instance of Theorem 8.2 from [Fit09]. There it is formulated and proved for $\mathsf{JT4}$, but the proof makes use only of the operations $+$ and $\cdot$, of the Internalization Property, and the Substitution Lemma. Hence, the theorem also holds for all the justification logics we consider, provided the constant specifications used are axiomatically appropriate and schematic (i.e., they enforce the Internalization Property and the Substitution Lemma).

Theorem 3.1.6 (Realization Merging). *Let* JL *be a justification logic and let* CS *be an axiomatically appropriate and schematic constant specification for* JL. *Let* A *be a properly annotated formula, and* $r_1, \ldots, r_n$ *be realization functions on* A. *Then there exists a realization function* r *on* A *and a substitution* σ *that lives on* A *such that*

$$\mathsf{JL}_{\mathsf{CS}} \vdash A^{r_i}\sigma \to A^r$$

for $i = 1, \ldots, n$. *(Note that it is not assumed that the* A^{r_i}*'s are provable.)*

The following properties are used in many of the proofs that follow.

Fact 3.1.7 (Combinations of Substitutions and Realization Functions). *Let* A *be an annotated formula,* σ_1 *and* σ_2 *be substitutions, and* r *be a prerealization function.*

(1) $\mathrm{vars}(t\sigma_1) \subseteq \mathrm{vars}(t) \cup \mathrm{vrange}(\sigma_1)$ *for any term* t*;*

(2) $\sigma_2 \circ \sigma_1$ *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,* $A(\sigma_2 \circ \sigma_1) = (A\sigma_1)\sigma_2$*;*

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

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

(5) $\sigma_1 \circ r$ *is a prerealization function with* $\mathrm{dom}(\sigma_1 \circ r) = \mathrm{dom}(r)$*;*

(6) in particular, if r *is a prerealization function on* A, *then so is* $\sigma_1 \circ r$ *and* $A^{\sigma_1 \circ r} = A^r\sigma_1$*;*

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

Whenever r_1 *and* r_2 *are realization functions,*

(8) 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;*

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

(10) $\sigma_1 \circ r_1$ *is a realization function iff* $x_n \notin \mathrm{dom}(\sigma_1)$ *whenever* $r_1(2n)$ *is defined.*

Proof. (1): We prove the statement by induction on the structure of t. If t is a constant c^i_j, $\mathrm{vars}(c^i_j\sigma_1) = \mathrm{vars}(c^i_j) = \varnothing$. If t is a variable

$x_i \in \text{dom}(\sigma_1)$, then $\text{vars}(x_i\sigma_1) = \text{vars}(\sigma_1(x_i)) \subseteq \text{vrange}(\sigma_1)$. If t is a variable $x_i \notin \text{dom}(\sigma_1)$, then $\text{vars}(x_i\sigma_1) = \{x_i\} \subseteq \text{vars}(t)$.

If t is $*s$ for some unary operation $*$, then

$$\text{vars}((*s)\sigma_1) = \text{vars}(*(s\sigma_1)) = \text{vars}(s\sigma_1) \ ,$$

which, by induction hypothesis, is contained in

$$\text{vars}(s) \cup \text{vrange}(\sigma_1) = \text{vars}(*s) \cup \text{vrange}(\sigma_1) \ .$$

If t is $(t_1 * t_2)$, for some binary operation $*$, then $\text{vars}((t_1 * t_2)\sigma_1) = \text{vars}(t_1\sigma_1 * t_2\sigma_1) = \text{vars}(t_1\sigma_1) \cup \text{vars}(t_2\sigma_1)$, which, by induction hypothesis, is contained in

$$\text{vars}(t_1) \cup \text{vrange}(\sigma_1) \cup \text{vars}(t_2) \cup \text{vrange}(\sigma_1) = \\ \text{vars}(t_1 * t_2) \cup \text{vrange}(\sigma_1) \ .$$

(2): For any variable x_i, the composition $(\sigma_2 \circ \sigma_1)(x_i)$ is defined as $\sigma_1(x_i)\sigma_2$. Hence, it is a total mapping from variables to terms and, thus, a substitution. That $\text{dom}(\sigma_2 \circ \sigma_1) \subseteq \text{dom}(\sigma_1) \cup \text{dom}(\sigma_2)$ is obvious.

To show that $\text{vrange}(\sigma_2 \circ \sigma_1) \subseteq \text{vrange}(\sigma_1) \cup \text{vrange}(\sigma_2)$ we show that

$$\text{vars}((\sigma_2 \circ \sigma_1)x_i) \subseteq \text{vrange}(\sigma_1) \cup \text{vrange}(\sigma_2)$$

for any $x_i \in \text{dom}(\sigma_2 \circ \sigma_1)$. Indeed, $\text{vars}((\sigma_2 \circ \sigma_1)x_i) = \text{vars}(\sigma_1(x_i)\sigma_2)$. If $x_i \in \text{dom}(\sigma_1)$, then, by 3.1.7 (1),

$$\text{vars}(\sigma_1(x_i)\sigma_2) \subseteq \text{vars}(\sigma_1(x_i)) \cup \text{vrange}(\sigma_2) \subseteq \text{vrange}(\sigma_1) \cup \text{vrange}(\sigma_2) \ .$$

If $x_i \notin \text{dom}(\sigma_1)$, $x_i \in \text{dom}(\sigma_2)$ and $\text{vars}(\sigma_1(x_i)\sigma_2) = \text{vars}(\sigma_2(x_i)) \subseteq \text{vrange}(\sigma_2)$.

To show $A(\sigma_2 \circ \sigma_1) = A\sigma_1\sigma_2$, it is enough to show $t(\sigma_2 \circ \sigma_1) = t\sigma_1\sigma_2$ by induction on the structure of an arbitrary term t. If t is a constant, there is nothing to prove.

If t is a variable x_i, then, by definition, $x_i(\sigma_2 \circ \sigma_1) = (\sigma_2 \circ \sigma_1)(x_i) = \sigma_1(x_i)\sigma_2 = x_i\sigma_1\sigma_2$.

If t is $*s$, for some unary operation $*$, then, by definition, $(*s)\sigma_1\sigma_2 = (*(s\sigma_1))\sigma_2 = *(s\sigma_1\sigma_2)$, which, by induction hypothesis, is the same as $*(s(\sigma_2 \circ \sigma_1)) = (*s)(\sigma_2 \circ \sigma_1)$.

If t is $(t_1 * t_2)$, for some binary operation $*$, then, by definition,

$$(t_1 * t_2)\sigma_1\sigma_2 = (t_1\sigma_1 * t_2\sigma_1)\sigma_2 = (t_1\sigma_1\sigma_2 * t_2\sigma_1\sigma_2) \ ,$$

which, by induction hypothesis, is the same as $(t_1(\sigma_2 \circ \sigma_1) * t_2(\sigma_2 \circ \sigma_1)) = (t_1 * t_2)(\sigma_2 \circ \sigma_1)$.

(3) is obvious from the definition on page 60.

(4): For $x_i \in \mathrm{dom}(\sigma_1)$ we have $(\sigma_1 \cup \sigma_2)(x_i) = \sigma_1(x_i) = \sigma_1(x_i)\sigma_2$ because no variable from vrange(σ_1) is affected by σ_2.

For $x_i \in \mathrm{dom}(\sigma_2)$ we have $(\sigma_1 \cup \sigma_2)(x_i) = \sigma_2(x_i) = x_i\sigma_2 = \sigma_1(x_i)\sigma_2$ because $\sigma_1(x_i) = x_i$ for a variable from $\mathrm{dom}(\sigma_2)$.

For $x_i \notin \mathrm{dom}(\sigma_1) \cup \mathrm{dom}(\sigma_2)$ we have $(\sigma_1 \cup \sigma_2)(x_i) = x_i = x_i\sigma_1\sigma_2$.

(5): The composition $(\sigma_1 \circ r)(n)$ is defined as $r(n)\sigma_1$, for any n that r is defined on. So it is obviously a prerealization function with the same domain as r.

(6): Since r and $\sigma_1 \circ r$ have the same domain by 3.1.7 (5), if r is a prerealization function on A, then so is $\sigma_1 \circ r$.

We show $A^{\sigma_1 \circ r} = A^r\sigma_1$ by induction on the structure of A. The atomic and propositional cases are trivial.

Let $A = \Box_k B$ for an odd index k. By definition,

$$(\Box_k B)^{\sigma_1 \circ r} = (\sigma_1 \circ r)(k) : B^{\sigma_1 \circ r} = r(k)\sigma_1 : B^{\sigma_1 \circ r} \ ,$$

which, by induction hypothesis, is the same as $r(k)\sigma_1 : (B^r\sigma_1)$. On the other hand, $(\Box_k B)^r\sigma_1 = (r(k) : B^r)\sigma_1 = r(k)\sigma_1 : (B^r\sigma_1)$.

Let $A = \Diamond_{2k} B$. By definition,

$$(\Diamond_{2k} B)^{\sigma_1 \circ r} = \neg(\sigma_1 \circ r)(2k) : \neg B^{\sigma_1 \circ r} = \neg(r(2k)\sigma_1) : \neg B^{\sigma_1 \circ r} \ ,$$

which, by induction hypothesis, is the same as $\neg(r(2k)\sigma_1) : \neg(B^r\sigma_1)$. On the other hand, $(\Diamond_{2k} B)^r\sigma_1 = (\neg r(2k) : \neg B^r)\sigma_1 = \neg(r(2k)\sigma_1) : \neg(B^r\sigma_1)$.

(7) is obvious.

(8) follows from the fact that all realization functions agree on even numbers, i.e., the function value of $2k$ is always the variable x_k.

(9) is obvious.

(10): Assume that $\sigma_1 \circ r_1$ is a realization function and that $r_1(2n)$ is defined. Then $x_n = (\sigma_1 \circ r_1)(2n) = r_1(2n)\sigma_1 = \sigma_1(x_n)$. Therefore $x_n \notin \mathrm{dom}(\sigma_1)$.

Now assume that $x_n \notin \mathrm{dom}(\sigma_1)$ whenever $r_1(2n)$ is defined. That $\sigma_1 \circ r_1$ is a prerealization function follows from 3.1.7 (5). To show that it is a realization function, assume that $(\sigma_1 \circ r_1)(2n)$ is defined for some arbitrary number $2n$. I.e., $r_1(2n)$ is defined and $r_1(2n) = x_n$. From $x_n \notin \mathrm{dom}(\sigma_1)$ it follows that $\sigma_1(x_n) = x_n$ and $(\sigma_1 \circ r_1)(2n) = x_n$. □

Corollary 3.1.8. *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.*

Proof. Assume that r is a realization function on A and σ lives away from A, i.e., $\mathrm{dom}(\sigma) \cap \mathrm{vars}_{\Diamond}(A) = \varnothing$. Then $r \restriction A$ is a realization function on A by 3.1.7 (7) and $\sigma \circ (r \restriction A)$ is a prerealization function on A by 3.1.7 (6). Let $r \restriction A$ be defined on a number $2n$, i.e., $\Diamond_{2n}$ occur in A. Since σ lives away from A, we have $x_n \notin \mathrm{dom}(\sigma)$. It thus follows from 3.1.7 (10) that $\sigma \circ (r \restriction A)$ is a realization function. □

3.2 A General Realization Method

We now present our general realization procedure based on nested sequent calculi. The essence of this procedure is that realizing arbitrary nested sequent rules can be reduced to realizing their non-nested (or shallow) versions (cf. Lemma 3.2.10), which is even simpler than realizing rules of an ordinary sequent calculus. As a consequence, in order to realize a modal logic presented via a nested sequent system, it is sufficient to realize the shallow versions of all the rules of the system (cf. Theorem 3.2.11). Realization of various (shallow) nested sequent rules and proofs of actual realization theorems are postponed until Section 3.3.

Nested sequents. *Nested sequents*, or *sequents* for short, 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 by

$$\underline{\varnothing} := \bot, \qquad \underline{\Sigma, A} := \begin{cases} (\underline{\Sigma} \vee A) & \text{if } \Sigma \neq \varnothing, \\ A & \text{otherwise,} \end{cases}$$

$$\underline{\Sigma, [\Delta]} := \begin{cases} (\underline{\Sigma} \vee \Box\underline{\Delta}) & \text{if } \Sigma \neq \varnothing, \\ \Box\underline{\Delta} & \text{otherwise.} \end{cases} \tag{3.1}$$

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

Sequent contexts. A *sequent context*, or *context* for short—denoted by $\Gamma\{\ \}$, $\Delta\{\ \}$, etc.—is inductively defined as follows:

- $\{\ \}$ is a context (the symbol $\{\ \}$ is called *hole*);
- 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:

$$\mathsf{exch}\ \frac{\Gamma\{\Delta, \Sigma\}}{\Gamma\{\Sigma, \Delta\}} \tag{3.2}$$

One of the instances of (3.2) is

$$\mathsf{exch}\ \frac{\big[P_2 \wedge \Diamond P_3, [P_1], P_1\big], \big[P_1, \neg P_1\big]}{\big[[P_1], P_1, P_2 \wedge \Diamond P_3\big], \big[P_1, \neg P_1\big]}$$

where the context $\Gamma\{\ \} = [\{\ \}], [P_1, \neg P_1]$ and sequents $\Delta = P_2 \wedge \Diamond P_3$ and $\Sigma = [P_1], P_1$. In Section 3.3 we provide systems of such rules for all logics in the modal cube and use these systems to prove actual realization theorems for these logics. In this section, however, we treat arbitrary *context-sharing* nested rules, i.e., rules of the form

$$\frac{\Gamma\{S_1\} \quad \dots \quad \Gamma\{S_n\}}{\Gamma\{S\}}$$

where n is a nonnegative 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. A sequent schema differs from a sequent in the sense that, instead of modal formulas, it contains variables over sequents and variables over modal formulas. For example, $[\Delta, A]$ is a sequent schema, where Δ is a variable over sequents and A is a variable over modal formulas. Each context-sharing 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 (3.2) is

$$\mathsf{sh\text{-}exch}\ \frac{\Delta, \Sigma}{\Sigma, \Delta}$$

From now on, by a *nested rule* we mean a context-sharing nested rule.

In addition, contexts provide for an especially simple definition of subsequent:

Definition 3.2.1 (Subsequent). A *subsequent* of a given sequent Γ is any sequent Δ such that $\Gamma = \Sigma\{\Delta\}$ for some context $\Sigma\{\ \}$.

Definition 3.2.2 (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 Γ is an annotated formula defined as in (3.1), except that the third case is replaced by

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

Remark 3.2.3. 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 $\underline{\Gamma}$, in particular, $\Gamma^r := (\underline{\Gamma})^r$, $\mathrm{vars}_\Diamond(\Gamma) := \mathrm{vars}_\Diamond(\underline{\Gamma})$, 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 3.2.4 (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.*

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

Just like 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 $\underline{\Delta}$ is not in general a subformula of $\underline{\Gamma\{\Delta\}}$—e.g., for $\Gamma\{\ \} = P_1, \{\ \}$ and $\Delta = P_2, P_3$, the formula $\underline{\Delta} = P_2 \vee P_3$ is not a subformula of $\underline{\Gamma\{\Delta\}} = \underline{P_1, P_2, P_3} = (P_1 \vee P_2) \vee P_3$. The following fact is used as a matter of course without explicit mention.

Fact 3.2.5 (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.*

Proof. Simply by the fact that every index of Δ is also an index of $\Gamma\{\Delta\}$. □

The following lemma is needed because, in general, the formula $\underline{\Gamma, \Sigma}$ does not coincide with $\underline{\Gamma} \vee \underline{\Sigma}$.

Lemma 3.2.6 (Associativity of Disjunction)**.** *For any justification logic* JL *and constant specification* CS *for* JL*,*

$$\mathsf{JL}_{\mathsf{CS}} \vdash \underline{(\Sigma, \Gamma)}^r\sigma \leftrightarrow \underline{\Sigma}^r\sigma \vee \underline{\Gamma}^r\sigma \ ,$$

where Γ and Σ are arbitrary annotated sequents, r is any realization function on Σ, Γ, and σ is any substitution.

Proof. By induction on the structure of Γ.

If $\Gamma = \varnothing$, then $\Sigma, \Gamma = \Sigma$ and $\Gamma^r\sigma = \bot^r\sigma = \bot$. By propositional reasoning, $\mathsf{JL}_{\mathsf{CS}} \vdash \Sigma^r\sigma \leftrightarrow \Sigma^r\sigma \vee \bot$.

Let $\Gamma = \Delta, A$. By induction hypothesis,

$$\mathsf{JL}_{\mathsf{CS}} \vdash (\Sigma, \Delta)^r\sigma \leftrightarrow \Sigma^r\sigma \vee \Delta^r\sigma \ .$$

By propositional reasoning,

$$\mathsf{JL}_{\mathsf{CS}} \vdash (\Sigma, \Delta)^r\sigma \vee A^r\sigma \leftrightarrow \Sigma^r\sigma \vee (\Delta^r\sigma \vee A^r\sigma) \ ,$$

which can be rewritten as

$$\mathsf{JL}_{\mathsf{CS}} \vdash (\underline{\Sigma, \Delta} \vee A)^r\sigma \leftrightarrow \Sigma^r\sigma \vee (\underline{\Delta} \vee A)^r\sigma \ ,$$

which is what we need since $\underline{\Sigma, \Delta, A} = \underline{\Sigma, \Delta} \vee A$ and $\underline{\Delta, A} = \underline{\Delta} \vee A$.

Let $\Gamma = \Delta, [\Omega]_k$. By induction hypothesis and propositional reasoning,

$$\mathsf{JL}_{\mathsf{CS}} \vdash (\Sigma, \Delta)^r\sigma \vee (\Box_k\underline{\Omega})^r\sigma \leftrightarrow \Sigma^r\sigma \vee \big(\Delta^r\sigma \vee (\Box_k\underline{\Omega})^r\sigma\big) \ ,$$

which can be rewritten as

$$\mathsf{JL}_{\mathsf{CS}} \vdash (\underline{\Sigma, \Delta} \vee \Box_k\underline{\Omega})^r\sigma \leftrightarrow \Sigma^r\sigma \vee \big(\underline{\Delta} \vee \Box_k\underline{\Omega}\big)^r\sigma \ ,$$

which is what we need since $\underline{\Sigma, \Delta, [\Omega]_k} = \underline{\Sigma, \Delta} \vee \Box_k\underline{\Omega}$ and $\underline{\Delta, [\Omega]_k} = \underline{\Delta} \vee \Box_k\underline{\Omega}$. $\Box$

Definition 3.2.7 (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 $\Gamma\{\}, \Lambda_1, \dots, \Lambda_n, \Lambda$ are annotated versions of $\Gamma'\{\}, \Lambda'_1, \dots, \Lambda'_n, \Lambda'$ respectively, $\Gamma\{\Lambda_1\}, \dots, \Gamma\{\Lambda_n\}$, and $\Gamma\{\Lambda\}$ are properly annotated, and no index occurs in both Λ_i and Λ_j for any $1 \le i < j \le n$. Note that the annotated context $\Gamma\{\ \}$ is the same for every premise and the conclusion.

Definition 3.2.8 (Realizable Rule)**.** An instance $\dfrac{}{\Gamma'\{\Lambda'\}}$ of a zero-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 common context $\Gamma'\{\}$ is called *realizable* in JL if there is an annotated version $\dfrac{\Gamma\{\Lambda_1\} \quad \dots \quad \Gamma\{\Lambda_n\}}{\Gamma\{\Lambda\}}$ of this instance such that for arbitrary 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

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

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

Remark 3.2.9 (Realizability of Cut)**.** Currently it is not known whether the cut rule

$$\frac{\Gamma\{A\} \quad \Gamma\{\neg A\}}{\Gamma\{\varnothing\}}$$

is realizable in J or in some of its extensions. Note that the fact that the cut rule is nested is not a problem (cf. Lemma 3.2.10 below). The problem, however, is that the cut formula A occurs positively in one premise and negatively in the other premise. As an example, consider the following (shallow) instance of the cut rule:

$$\frac{\Box P_1 \vee \Diamond P_2 \quad \Diamond\neg P_1 \wedge \Box\neg P_2}{\varnothing}$$

According to Definition 3.2.8, the above instance is realizable in J if there exists an annotated version

$$\frac{\Box_{2k+1} P_1 \vee \Diamond_{2l} P_2 \quad \Diamond_{2m}\neg P_1 \wedge \Box_{2n+1}\neg P_2}{\varnothing}$$

of it such that for arbitrary realization functions r_1 and r_2 on $\Box_{2k+1}P_1 \vee \Diamond_{2l}P_2$ and $\Diamond_{2m}\neg P_1 \wedge \Box_{2n+1}\neg P_2$, respectively, there exists a substitution σ that lives on both (annotated) premises such that

$$\mathsf{J} \vdash (\Box_{2k+1}P_1 \vee \Diamond_{2l}P_2)^{r_1}\sigma \ \to \ (\Diamond_{2m}\neg P_1 \wedge \Box_{2n+1}\neg P_2)^{r_2}\sigma \ \to \ \bot \ ,$$

which can be rewritten as

$$\mathsf{J} \vdash (r_1(2k+1) : P_1 \vee \neg x_l : \neg P_2)\sigma \;\rightarrow\; (\neg x_m : \neg\neg P_1 \wedge r_2(2n+1) : \neg P_2)\sigma \;\rightarrow\; \bot \;,$$

which is propositionally equivalent to

$$\mathsf{J} \vdash (r_1(2k+1) : P_1 \vee \neg x_l : \neg P_2)\sigma \;\rightarrow\; (x_m : \neg\neg P_1 \vee \neg r_2(2n+1) : \neg P_2)\sigma \;.$$

An obvious solution would be to choose a σ that unifies the terms $r_1(2k+1)$ and x_m and the terms $r_2(2n+1)$ and x_l respectively. But since, in general, $r_1(2k+1)$ contains the variable x_l and $r_2(2n+1)$ contains the variable x_m, this unification problem causes circular dependencies and, hence, has no solution. It is open whether a more sophisticated definition of realizability allows for a realization of the cut rule. Fortunately, all our nested sequent systems are cut-free.

Lemma 3.2.10 (From Shallow to Nested)**.** *Let* JL *be a justification logic and let* CS *be an axiomatically appropriate and schematic constant specification for* JL*. For any nested rule* ρ*, if its shallow version* $\mathsf{sh}\text{-}\rho$ *is realizable in* $\mathsf{JL}_{\mathsf{CS}}$*, then* ρ *itself is realizable in* $\mathsf{JL}_{\mathsf{CS}}$*.*

Proof. We prove the lemma for the harder case where ρ has $n > 0$ premises. The proof for the case when $n = 0$ is much simpler and can be read off from the case $n > 0$.

We consider an arbitrary instance

$$\frac{\Delta'\{\Lambda'_1\} \quad \dots \quad \Delta'\{\Lambda'_n\}}{\Delta'\{\Lambda'\}} \tag{3.3}$$

of ρ and show that it is realizable in $\mathsf{JL}_{\mathsf{CS}}$. By assumption, its shallow version $\dfrac{\Lambda'_1 \quad \dots \quad \Lambda'_n}{\Lambda'}$, which is an instance of $\mathsf{sh}\text{-}\rho$, has an annotated version $\dfrac{\Lambda_1 \quad \dots \quad \Lambda_n}{\Lambda}$ such that for any realization functions $r_1, \dots, r_n$ on $\Lambda_1, \dots, \Lambda_n$ respectively, there exists a realization function r_0 on Λ and a substitution σ_0 that lives on each of Λ_i, $i = 1, \dots, n$, such that $\mathsf{JL}_{\mathsf{CS}} \vdash (\Lambda_1)^{r_1}\sigma_0 \rightarrow \dots \rightarrow (\Lambda_n)^{r_n}\sigma_0 \rightarrow \Lambda^{r_0}$.

We prove a stronger statement, namely that for any annotated context $\Gamma\{\;\}$ such that $\Gamma\{\Lambda_1\}, \dots, \Gamma\{\Lambda_n\}$, and $\Gamma\{\Lambda\}$ are properly annotated and for arbitrary 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

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

It then follows that the above also holds for some particular annotated context $\Gamma\{\ \} = \Delta\{\ \}$ such that $\dfrac{\Delta\{\Lambda_1\} \quad \dots \quad \Delta\{\Lambda_n\}}{\Delta\{\Lambda\}}$ is an annotated version of our arbitrary ρ-instance (3.3). The proof is by induction on the structure of $\Gamma\{\ \}$.

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

Case $\Gamma\{\ \} = [\Sigma\{\ \}]_k$. Let $r_1, \dots, r_n$ be realization functions on $[\Sigma\{\Lambda_1\}]_k, \dots, [\Sigma\{\Lambda_n\}]_k$ respectively. Since $\Sigma\{\Lambda_1\}, \dots, \Sigma\{\Lambda_n\}$, and $\Sigma\{\Lambda\}$ are properly annotated as subsequents of properly annotated sequents $[\Sigma\{\Lambda_1\}]_k, \dots, [\Sigma\{\Lambda_n\}]_k$, and $[\Sigma\{\Lambda\}]_k$ respectively and since $r_1, \dots, r_n$ are also realization functions on $\Sigma\{\Lambda_1\}, \dots, \Sigma\{\Lambda_n\}$ 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

$$\mathsf{JL_{CS}} \vdash \Sigma\{\Lambda_1\}^{r_1}\sigma' \ \rightarrow\ \dots\ \rightarrow\ \Sigma\{\Lambda_n\}^{r_n}\sigma' \ \rightarrow\ \Sigma\{\Lambda\}^{r'} \ . \tag{3.4}$$

By Internalization Property 1.5.4, there exists a term $t(x_1, \dots, x_n)$ such that

$$\begin{aligned}\mathsf{JL_{CS}} \vdash r_1(k)\sigma' : \big(\Sigma\{\Lambda_1\}^{r_1}\sigma'\big) \ \rightarrow\ \dots\ \rightarrow\ r_n(k)\sigma' : \big(\Sigma\{\Lambda_n\}^{r_n}\sigma'\big)&\\ \rightarrow\ t\big(r_1(k)\sigma', \dots, r_n(k)\sigma'\big) : \Sigma\{\Lambda\}^{r'}.&\end{aligned} \tag{3.5}$$

Let

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

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 3.1.7. Further, (3.5) can now be rewritten as

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

For each $i = 1, \dots, n$, 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 $r_1, \dots, r_n$ be realization functions on $\Delta, \Sigma\{\Lambda_1\}, \Pi\ , \dots,\ \Delta, \Sigma\{\Lambda_n\}, \Pi$ 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 (3.4) holds.

By Fact 3.2.5, 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 3.1.8,

$$\sigma' \circ (r_i \restriction \Delta, \Pi)$$

is a realization function on Δ, Π for each $i = 1, \dots, n$.

By Theorem 3.1.6 (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 $i = 1, \dots, n$

$$\mathsf{JL_{CS}} \vdash (\Delta, \Pi)^{\sigma' \circ (r_i \restriction \Delta, \Pi)} \sigma_M \ \rightarrow\ (\Delta, \Pi)^{r_M} \ . \tag{3.6}$$

By Fact 3.1.7 (6), $(\Delta, \Pi)^{\sigma' \circ (r_i \restriction \Delta, \Pi)} \sigma_M = (\Delta, \Pi)^{r_i} \sigma' \sigma_M$. Therefore, (3.6) can be rewritten as

$$\mathsf{JL_{CS}} \vdash (\Delta, \Pi)^{r_i} \sigma' \sigma_M \ \rightarrow\ (\Delta, \Pi)^{r_M} \ . \tag{3.7}$$

From the induction hypothesis (3.4) it follows by the Substitution Lemma 1.5.9 that

$$\mathsf{JL_{CS}} \vdash \Sigma\{\Lambda_1\}^{r_1} \sigma' \sigma_M \ \rightarrow\ \dots\ \rightarrow\ \Sigma\{\Lambda_n\}^{r_n} \sigma' \sigma_M \ \rightarrow\ \Sigma\{\Lambda\}^{r'} \sigma_M \ .$$

From this and (3.7) it follows by propositional reasoning that

$$\begin{aligned} \mathsf{JL_{CS}} \vdash \Sigma\{\Lambda_1\}^{r_1} \sigma' \sigma_M \vee (\Delta, \Pi)^{r_1} \sigma' \sigma_M \ &\rightarrow\ \dots \\ \rightarrow\ \Sigma\{\Lambda_n\}^{r_n} \sigma' \sigma_M &\vee (\Delta, \Pi)^{r_n} \sigma' \sigma_M \\ &\rightarrow\ \Sigma\{\Lambda\}^{r'} \sigma_M \vee (\Delta, \Pi)^{r_M} \ . \end{aligned} \tag{3.8}$$

Since $\Delta, \Sigma\{\Lambda\}, \Pi$ is properly annotated and σ_M lives on Δ, Π, it lives away from $\Sigma\{\Lambda\}$, hence $\sigma_M \circ \big(r' \restriction \Sigma\{\Lambda\}\big)$ is a realization function on $\Sigma\{\Lambda\}$ by Corollary 3.1.8. By Facts 3.1.7 (5), 3.1.7 (8), and 3.1.7 (9), 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 $i = 1, \ldots, n$ by Fact 3.1.7 (2). By Fact 3.1.7 (6),

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

which allows to rewrite (3.8) as

$$\begin{aligned}\mathsf{JL}_{\mathsf{CS}} \vdash \big(\Sigma\{\Lambda_1\} \vee (\Delta, \Pi)\big)^{r_1}\sigma \;\rightarrow\; \ldots \;\rightarrow\; & \big(\Sigma\{\Lambda_n\} \vee (\Delta, \Pi)\big)^{r_n}\sigma \\ & \rightarrow\; \big(\Sigma\{\Lambda\} \vee (\Delta, \Pi)\big)^{r} ,\end{aligned}$$

which, by Lemma 3.2.6, is propositionally equivalent to

$$\begin{aligned}\mathsf{JL}_{\mathsf{CS}} \vdash \big(\Delta, \Sigma\{\Lambda_1\}, \Pi\big)^{r_1}\sigma \;\rightarrow\; \ldots \;\rightarrow\; & \big(\Delta, \Sigma\{\Lambda_n\}, \Pi\big)^{r_n}\sigma \\ & \rightarrow\; \big(\Delta, \Sigma\{\Lambda\}, \Pi\big)^{r} . \square\end{aligned}$$

Recall that the height of a node in a (derivation) tree is the length of the longest downward path to a leaf from that node. The *height* of a (derivation) tree is the height of its root.

Theorem 3.2.11 (Realization of Nested Systems)**.** *Let* JL *be a justification logic and let* CS *be an axiomatically appropriate and schematic constant specification for* JL*. Let* S *be a system of nested rules whose shallow versions are realizable in* $\mathsf{JL}_{\mathsf{CS}}$*. 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* $\mathsf{JL}_{\mathsf{CS}} \vdash \Gamma^r$*.*

Proof. By induction on the height of a derivation of Γ' in the system S. By Lemma 3.2.10, all rules used in this derivation are realizable in $\mathsf{JL}_{\mathsf{CS}}$. 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 $\mathsf{JL}_{\mathsf{CS}}$. Let $\Gamma' = \Delta'\{\Lambda'\}$ be the conclusion of an instance

$$\frac{\Delta'\{\Lambda'_1\} \quad \ldots \quad \Delta'\{\Lambda'_n\}}{\Delta'\{\Lambda'\}} \tag{3.9}$$

of an n-premise rule ρ with common context $\Delta'\{\}$, where $n > 0$. Since ρ is realizable in $\mathsf{JL}_{\mathsf{CS}}$, there exists an annotated version

$$\frac{\Delta\{\Lambda_1\} \quad \ldots \quad \Delta\{\Lambda_n\}}{\Delta\{\Lambda\}}$$

of (3.9) such that for arbitrary 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

$$\mathsf{JL}_{\mathsf{CS}} \vdash \Delta\{\Lambda_1\}^{r_1}\sigma \ \to \ \dots \ \to \ \Delta\{\Lambda_n\}^{r_n}\sigma \ \to \ \Delta\{\Lambda\}^{r} \ . \tag{3.10}$$

By induction hypothesis, for each $i = 1, \dots, n$ 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 $\mathsf{JL}_{\mathsf{CS}} \vdash \Delta_i\{\overline{\Lambda}_i\}^{\overline{r}_i}$. Since $\Delta\{\Lambda_i\}$ is another properly annotated version of the same premise $\Delta'\{\Lambda'_i\}$, by Lemma 3.1.5 there exists a realization function r_i on $\Delta\{\Lambda_i\}$ such that $\mathsf{JL}_{\mathsf{CS}} \vdash \Delta\{\Lambda_i\}^{r_i}$.

Let r and σ be obtained from the realizability of ρ for these functions $r_1, \dots, r_n$. By the Substitution Lemma 1.5.9, $\mathsf{JL}_{\mathsf{CS}} \vdash \Delta\{\Lambda_i\}^{r_i}\sigma$ for each $i = 1, \dots, n$. It now follows from (3.10) by n applications of modus ponens that $\mathsf{JL}_{\mathsf{CS}} \vdash \Delta\{\Lambda\}^{r}$. It remains to note that $\Delta\{\Lambda\}$ is a properly annotated version of the conclusion $\Delta'\{\Lambda'\} = \Gamma'$ of the rule instance (3.9). $\square$

3.3 A Uniform and Modular Realization Theorem

We use Theorem 3.2.11 to prove a uniform realization theorem for all the logics in the modal cube, 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 Lemma 3.3.8 (contraction) is the most interesting one. While there is no principal difference in the treatment of modal rules (Lemmas 3.3.9 and 3.3.11), some of the rules require extra work. In this respect, the rules that are used in modal logics with negative introspection turned out to be the hardest.

Remark 3.3.1. It is interesting to note that while dealing with contraction (Lemma 3.3.8) is one of the main challenges of our method, it did not create any problems for Fitting in [Fit09], where he applies a similar method to sequent calculi. For an advanced reader, the reason for this inequality might be interesting. Merging, which plays a crucial role both in Fitting's and in our method, prohibits repetitions in the

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

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

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

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

Figure 3.2: Rules of the nested sequent calculus

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 3.3: Sequent systems of modal logic

annotation, forcing us to annotate the formulas being contracted differently and prompting the explicit reconciliation of the annotations as detailed in Lemma 3.3.8. However, Fitting was able to sidestep this by merging things on a formula level and, thus, being able to use the same annotation for the formulas being contracted. The richer structure of nested sequents, with its structural modalities that also require merging, prevents the same trick from being used in our case.

Remark 3.3.2. Note also that while dealing with the shallow versions of all the logical propositional rules is equally trivial, the case of conjunction is significantly more complicated in the nested case. This is due to the fact that conjunction is the only multi-premise rule, by virtue of which its nested version requires the real use of merging in Lemma 3.2.10.

Consider the inference rules in Figure 3.2. The *sequent system* SK consists of the rules id, $\vee$, $\wedge$, ctr, exch, $\Box$, and k. It corresponds to the axiom system K. *Extensions of system* SK are obtained by adding further rules from Figure 3.2 according to Figure 3.3, where 5 means that all three rules 5a, 5b, and 5c are added. Note that a name in the first row of Figure 3.3 now denotes 1) a logic, 2) a (Hilbert-style) axiom system, and 3) a sequent system at the same time.

These sequent systems are essentially the same as the ones in [Brü09], where their completeness is proved, so we have the following theorem.

Theorem 3.3.3 (Completeness). *System* SK *and its extensions are sound and complete with respect to their corresponding modal logics.*

Lemma 3.3.4 (id-rule). *Let* $\mathsf{JL} \supseteq \mathsf{J}$ *and let* CS *be an arbitrary constant specification for* JL*. The shallow version of the* id*-rule is realizable in* $\mathsf{JL_{CS}}$*.*

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

Lemma 3.3.5 ($\vee$- and exch-rule). *Let* $\mathsf{JL} \supseteq \mathsf{J}$ *and let* CS *be an arbitrary constant specification for* JL*. The shallow versions of the rules* $\vee$ *and* exch *are realizable in* $\mathsf{JL_{CS}}$*.*

Proof. For an arbitrary instance $\dfrac{A', B'}{A' \vee B'}$ of $\mathsf{sh}\text{-}\vee$, 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, $\big(\underline{A, B}\big)^{r_1} \sigma \to (\underline{A \vee B})^r$ is a propositional tautology and, thus, is provable in $\mathsf{JL_{CS}}$.

For an arbitrary instance $\dfrac{\Delta', \Sigma'}{\Sigma', \Delta'}$ of $\mathsf{sh}\text{-}\mathsf{exch}$, 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 $\mathsf{JL_{CS}} \vdash (\Delta, \Sigma)^{r_1} \sigma \to (\Sigma, \Delta)^r$ follows from Lemma 3.2.6. □

The realizability for the $\Box$-rule is trivial:

Lemma 3.3.6 ($\Box$-rule)**.** *Let* $\mathsf{JL} \supseteq \mathsf{J}$ *and let* CS *be an arbitrary constant specification for* JL*. The shallow version of the* $\Box$*-rule is realizable in* $\mathsf{JL}_{\mathsf{CS}}$*.*

Lemma 3.3.7 ($\wedge$-rule)**.** *Let* $\mathsf{JL} \supseteq \mathsf{J}$ *and let* CS *be an arbitrary constant specification for* JL*. The shallow version of the* $\wedge$*-rule is realizable in* $\mathsf{JL}_{\mathsf{CS}}$*.*

Proof. For an arbitrary instance $\dfrac{A' \quad B'}{A' \wedge B'}$ of $\mathsf{sh}\text{-}\wedge$, 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

$$r := (r_1 \upharpoonright A) \cup (r_2 \upharpoonright B)$$

and σ be the identity substitution. The above is a realization function on $A \wedge B$ by Facts 3.1.7 (7) and 3.1.7 (9). Finally, $A^{r_1}\sigma \to B^{r_2}\sigma \to (A \wedge B)^r$ is a propositional tautology and, thus, provable in $\mathsf{JL}_{\mathsf{CS}}$, since $(A \wedge B)^r = A^{r_1}\sigma \wedge B^{r_2}\sigma$. $\Box$

Lemma 3.3.8 (ctr-rule)**.** *Let* $\mathsf{JL} \supseteq \mathsf{J}$ *and let* CS *be an axiomatically appropriate and schematic constant specification for* JL*. The shallow version of the* ctr*-rule is realizable in* $\mathsf{JL}_{\mathsf{CS}}$*.*

Proof. For an arbitrary instance $\dfrac{A', A'}{A'}$ of $\mathsf{sh}\text{-}\mathsf{ctr}$, 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. Given an arbitrary realization function r_1 on A_1, A_2, for each subformula occurrence B_3 of A_3 we construct a realization function r on B_3 and a substitution σ that lives on $B_1 \vee B_2$ and, additionally, with $\mathrm{vrange}(\sigma) \subseteq \mathrm{vars}_{\Diamond}(B_3)$ such that

$$(B_1 \vee B_2)^{r_1}\sigma \to (B_3)^r \tag{3.11}$$

is provable in $\mathsf{JL}_{\mathsf{CS}}$ by induction on the structure of B_3, where B_1 and B_2 denote the subformula occurrences in A_1 and A_2 respectively that correspond to B_3 in A_3 (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 $B_1 \vee B_2$.

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, (3.11) can be rewritten as $B_3 \vee B_3 \to B_3$, a propositional tautology provable in $\mathsf{JL}_{\mathsf{CS}}$. Hence, one can take σ to be the identity substitution and $r := \varnothing$.

To prove the **inductive 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 that live on $D_1 \vee D_2$ and $C_1 \vee C_2$ respectively with $\mathrm{vrange}(\sigma'_D) \subseteq \mathrm{vars}_\Diamond(D_3)$ and $\mathrm{vrange}(\sigma'_C) \subseteq \mathrm{vars}_\Diamond(C_3)$ such that

$$\mathsf{JL}_{\mathsf{CS}} \vdash (D_1 \vee D_2)^{r_1}\sigma'_D \to (D_3)^{r'_D} \quad \text{and}$$
$$\mathsf{JL}_{\mathsf{CS}} \vdash (C_1 \vee C_2)^{r_1}\sigma'_C \to (C_3)^{r'_C} \ .$$

By the Substitution Lemma 1.5.9,

$$\mathsf{JL}_{\mathsf{CS}} \vdash (D_1 \vee D_2)^{r_1}\sigma'_D\sigma'_C \to (D_3)^{r'_D}\sigma'_C \quad \text{and} \tag{3.12}$$
$$\mathsf{JL}_{\mathsf{CS}} \vdash (C_1 \vee C_2)^{r_1}\sigma'_C\sigma'_D \to (C_3)^{r'_C}\sigma'_D \ . \tag{3.13}$$

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, $\mathrm{dom}(\sigma'_C) \subseteq \mathrm{vars}_\Diamond(C_1 \vee C_2)$ 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 3.1.7 (4) 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 3.1.7 (3). 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 3.1.7 (2), $\mathrm{vrange}(\sigma) \subseteq \mathrm{vars}_\Diamond(D_3) \cup \mathrm{vars}_\Diamond(C_3) = \mathrm{vars}_\Diamond(B_3)$. So σ is a suitable substitution and (3.12) and (3.13) can be rewritten as

$$\mathsf{JL}_{\mathsf{CS}} \vdash (D_1 \vee D_2)^{r_1}\sigma \to (D_3)^{r'_D}\sigma'_C \quad \text{and} \tag{3.14}$$
$$\mathsf{JL}_{\mathsf{CS}} \vdash (C_1 \vee C_2)^{r_1}\sigma \to (C_3)^{r'_C}\sigma'_D \ . \tag{3.15}$$

Since σ'_C and σ'_D live away from D_3 and C_3 respectively, by Corollary 3.1.8 both

$$r_D := \sigma'_C \circ (r'_D \restriction D_3) \quad \text{and} \quad r_C := \sigma'_D \circ (r'_C \restriction C_3)$$

are realization functions on D_3 and C_3 respectively. Further, by Fact 3.1.7 (6) we have $(D_3)^{r'_D}\sigma'_C = (D_3)^{r_D}$ and $(C_3)^{r'_C}\sigma'_D = (C_3)^{r_C}$. Now (3.14) and (3.15) can be rewritten as

$$\begin{aligned} &\mathsf{JL}_{\mathsf{CS}} \vdash (D_1 \vee D_2)^{r_1}\sigma \to (D_3)^{r_D} \quad \text{and} \\ &\mathsf{JL}_{\mathsf{CS}} \vdash (C_1 \vee C_2)^{r_1}\sigma \to (C_3)^{r_C} \ . \end{aligned}$$

Finally, by propositional reasoning, it is provable in $\mathsf{JL}_{\mathsf{CS}}$ that

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

which is exactly (3.11) for

$$r := r_D \cup r_C \ .$$

It is easy to see, using Fact 3.1.7, 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' that lives on $C_1 \vee C_2$ with $\mathrm{vrange}(\sigma') \subseteq \mathrm{vars}_\Diamond(C_3)$ such that $\mathsf{JL}_{\mathsf{CS}} \vdash (C_1 \vee C_2)^{r_1}\sigma' \to (C_3)^{r'}$. By propositional reasoning,

$$\mathsf{JL}_{\mathsf{CS}} \vdash \neg(C_3)^{r'} \ \to \ \neg(C_1)^{r_1}\sigma' \quad \text{and} \quad \mathsf{JL}_{\mathsf{CS}} \vdash \neg(C_3)^{r'} \ \to \ \neg(C_2)^{r_1}\sigma' \ .$$

By Internalization Property 1.5.4, there exist terms $t_1(x_1)$ and $t_2(x_1)$ such that

$$\begin{aligned} &\mathsf{JL}_{\mathsf{CS}} \vdash x_n : \neg(C_3)^{r'} \ \to \ t_1(x_n) : \big(\neg(C_1)^{r_1}\sigma'\big) \quad \text{and} \\ &\mathsf{JL}_{\mathsf{CS}} \vdash x_n : \neg(C_3)^{r'} \ \to \ t_2(x_n) : \big(\neg(C_2)^{r_1}\sigma'\big) \ . \end{aligned}$$

It then follows by propositional reasoning that

$$\begin{aligned} \mathsf{JL}_{\mathsf{CS}} \vdash \neg t_1(x_n) : \big(\neg(C_1)^{r_1}\sigma'\big) \ \vee \ \neg t_2(x_n) : \big(\neg(C_2)^{r_1}\sigma'\big) & \\ \to \ \neg x_n : \neg(C_3)^{r'} & \ . \end{aligned} \tag{3.16}$$

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, (3.16) can be rewritten as

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

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 1.5.9 and since $x_n \notin \{x_k, x_m\}$,

$$\mathsf{JL}_{\mathsf{CS}} \vdash \big(\neg t_1(x_n) : \neg(C_1)^{r_1} \ \vee\ \neg t_2(x_n) : \neg(C_2)^{r_1}\big)\sigma'\sigma'' \\ \rightarrow\ \neg x_n : \neg\big((C_3)^{r'}\sigma''\big) \ . \quad (3.17)$$

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 3.1.8 that $\sigma'' \circ (r' \restriction C_3)$ is a realization function on C_3. In addition, $(C_3)^{r'}\sigma'' = {C_3}^{\sigma'' \circ (r' \restriction C_3)}$. Therefore, (3.17) can be rewritten as

$$\mathsf{JL}_{\mathsf{CS}} \vdash \big(\neg t_1(x_n) : \neg(C_1)^{r_1} \ \vee\ \neg t_2(x_n) : \neg(C_2)^{r_1}\big)\sigma'\sigma'' \\ \rightarrow\ \neg x_n : \neg(C_3)^{\sigma'' \circ (r' \restriction C_3)} . \quad (3.18)$$

Let

$$\sigma := \sigma'' \circ \sigma' \qquad \text{and} \qquad r := \big(\sigma'' \circ (r' \restriction C_3)\big) \cup \{2n \mapsto x_n\} \ .$$

Clearly, r is a realization function on B_3. Since σ' affects none of x_k, x_m, $t_1(x_n)$, or $t_2(x_n)$, (3.18) can be rewritten as the provability in $\mathsf{JL}_{\mathsf{CS}}$ of $(\Diamond_{2k}C_1 \vee \Diamond_{2m}C_2)^{r_1}\sigma \ \rightarrow\ (\Diamond_{2n}C_3)^r$, which is exactly (3.11). It remains to note that, by Fact 3.1.7 (2),

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

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' that lives on $C_1 \vee C_2$ with $\mathrm{vrange}(\sigma') \subseteq \mathrm{vars}_\Diamond(C_3)$ such that

$\mathsf{JL}_{\mathsf{CS}} \vdash (C_1 \vee C_2)^{r_1}\sigma' \rightarrow (C_3)^{r'}$. By propositional reasoning and Internalization Property 1.5.4, there exist terms $t_1(x_1)$ and $t_2(x_1)$ such that

$$\begin{aligned}\mathsf{JL}_{\mathsf{CS}} &\vdash r_1(k)\sigma' : \big((C_1)^{r_1}\sigma'\big) \rightarrow t_1\big(r_1(k)\sigma'\big) : (C_3)^{r'} \quad \text{and}\\ \mathsf{JL}_{\mathsf{CS}} &\vdash r_1(l)\sigma' : \big((C_2)^{r_1}\sigma'\big) \rightarrow t_2\big(r_1(l)\sigma'\big) : (C_3)^{r'} \ .\end{aligned}$$

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

$$\begin{aligned}\mathsf{JL}_{\mathsf{CS}} &\vdash r_1(k)\sigma' : \big((C_1)^{r_1}\sigma'\big) \rightarrow s : (C_3)^{r'} \quad \text{and}\\ \mathsf{JL}_{\mathsf{CS}} &\vdash r_1(l)\sigma' : \big((C_2)^{r_1}\sigma'\big) \rightarrow s : (C_3)^{r'} \ .\end{aligned}$$

Thus, by propositional reasoning,

$$\mathsf{JL}_{\mathsf{CS}} \vdash \big(r_1(k) : (C_1)^{r_1} \vee r_1(l) : (C_2)^{r_1}\big)\sigma' \rightarrow s : (C_3)^{r'} \ . \tag{3.19}$$

Let

$$\sigma := \sigma' \quad \text{and} \quad r := (r' \restriction C_3) \cup \{m \mapsto s\} \ .$$

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 (3.19) can be rewritten to state the provability in $\mathsf{JL}_{\mathsf{CS}}$ of $(\Box_k C_1 \vee \Box_l C_2)^{r_1}\sigma \rightarrow (\Box_m C_3)^r$, which is exactly (3.11).

It remains to note that (3.11) for $B_3 = A_3$ and for thus constructed r and σ is just $\big(\underline{A_1, A_2}\big)^{r_1}\sigma \rightarrow \big(\underline{A_3}\big)^{r}$. $\Box$

Lemma 3.3.9 (k-rule)**.** *Let* $\mathsf{JL} \supseteq \mathsf{J}$ *and let* CS *be an axiomatically appropriate constant specification for* JL*. The shallow version of the* k*-rule is realizable in* $\mathsf{JL}_{\mathsf{CS}}$*.*

Proof. For an arbitrary instance $\dfrac{[A', \Delta']}{\Diamond A', [\Delta']}$ of $\mathsf{sh\text{-}k}$, 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 $(A, \Delta)^{r_1} \rightarrow \neg A^{r_1} \rightarrow \Delta^{r_1}$. By Internalization Property 1.5.4 there is a term $t(x_1, x_2)$ such that

$$\mathsf{JL}_{\mathsf{CS}} \vdash r_1(k) : (A, \Delta)^{r_1} \rightarrow x_m : \neg A^{r_1} \rightarrow t\big(r_1(k), x_m\big) : \Delta^{r_1} \ .$$

By propositional reasoning,

$$\mathsf{JL}_{\mathsf{CS}} \vdash r_1(k) : (A, \Delta)^{r_1} \;\rightarrow\; \neg x_m : \neg A^{r_1} \vee t\big(r_1(k), x_m\big) : \Delta^{r_1} \;\; . \tag{3.20}$$

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

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

is a realization function on $\Diamond_{2m}A, [\Delta]_i$. For the identity substitution σ and this r, (3.20) can be rewritten as

$$\mathsf{JL}_{\mathsf{CS}} \vdash \big([A, \Delta]_k\big)^{r_1}\sigma \;\rightarrow\; \big(\Diamond_{2m}A, [\Delta]_i\big)^{r} \;\; . \qquad \square$$

Lemma 3.3.11 covers the remaining rules. The part of Lemma 3.3.11 that concerns the rules 5a, 5b, and 5c uses auxiliary lemmas from Section 2.2 as well as the following corollary:

Corollary 3.3.10 (Internalized Inverse Positive Introspection)**.** *Let* $\mathsf{JL} \supseteq \mathsf{J5}$ *and let* CS *be an axiomatically appropriate and schematic constant specification for* JL*. There exists a term* $\mathrm{invposint}(x_1)$ *such that for the term* $\mathrm{t}_!(x_1)$ *constructed in Lemma 2.2.5,*

$$\mathsf{JL}_{\mathsf{CS}} \vdash \mathrm{invposint}(s) : \big(\neg\, \mathrm{t}_!(s) : s : A \rightarrow \neg s : A\big)$$

for any term s *and any formula* A*.*

Proof. By Lemma 2.2.5, for the terms $\mathrm{posint}(x_1)$ and $\mathrm{t}_!(x_1)$ constructed there, we have $\mathsf{JL}_{\mathsf{CS}} \vdash \mathrm{posint}(x_1) : \big(x_1 : P_1 \rightarrow \mathrm{t}_!(x_1) : x_1 : P_1\big)$. By propositional reasoning and Internalization Property 1.5.4 there exists a ground term p such that

$$\mathsf{JL}_{\mathsf{CS}} \vdash p : \Big(\big(x_1 : P_1 \rightarrow \mathrm{t}_!(x_1) : x_1 : P_1\big) \rightarrow \neg\, \mathrm{t}_!(x_1) : x_1 : P_1 \rightarrow \neg x_1 : P_1\Big) \;\; .$$

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

$$\mathsf{JL}_{\mathsf{CS}} \vdash \mathrm{invposint}(x_1) : \big(\neg\, \mathrm{t}_!(x_1) : x_1 : P_1 \rightarrow \neg x_1 : P_1\big) \;\; .$$

The desired result now follows from the Substitution Lemma 1.5.9. Note that $\mathrm{invposint}(x_1)$ depends on neither s nor A. $\square$

Lemma 3.3.11 (Modal Rules)**.** *Let* $\rho \in \{\mathsf{d}, \mathsf{t}, \mathsf{b}, \mathsf{4}, \mathsf{5a}, \mathsf{5b}, \mathsf{5c}\}$ *and let* $\mathsf{JL} \supseteq \mathsf{J}\rho$*, where by* Jd *we mean* JD*, and so on, except for* $\rho \in \{\mathsf{5a}, \mathsf{5b}, \mathsf{5c}\}$ *where we mean* $\mathsf{J5}$*. Let* CS *be an axiomatically appropriate and schematic constant specification for* JL*. The shallow version of* ρ *is realizable in* $\mathsf{JL}_{\mathsf{CS}}$*.*

Proof. We consider an arbitrary instance of sh-ρ for each rule ρ in turn.

Case ρ = d. For an arbitrary instance $\dfrac{[A']}{\Diamond A'}$ of sh-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 an app-instance $x_m : (A^{r_1} \to \bot) \to r_1(k) : A^{r_1} \to \bigl(x_m \cdot r_1(k)\bigr) : \bot$ it follows by propositional reasoning that

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

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

$$\mathsf{JL_{CS}} \vdash r_1(k) : A^{r_1} \ \to\ x_m : (A^{r_1} \to \bot) \to \bot \ ,$$

which is identical to $\mathsf{JL_{CS}} \vdash r_1(k) : A^{r_1} \to \neg x_m : \neg A^{r_1}$. Since $2m$ is even,

$$r := r_1 \cup \{2m \mapsto x_m\}$$

is a realization function on $\Diamond_{2m}A$ by Facts 3.1.7 (8) and 3.1.7 (9). Thus for the identity substitution σ and this r, we have

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

Case ρ = t. For an arbitrary instance $\dfrac{A'}{\Diamond A'}$ of sh-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 the contrapositive of a jt-instance $x_m : \neg A^{r_1} \to \neg A^{r_1}$ we have $\mathsf{JL_{CS}} \vdash A^{r_1} \to \neg x_m : \neg A^{r_1}$. Again, 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, $\mathsf{JL_{CS}} \vdash A^{r_1}\sigma \to (\Diamond_{2m}A)^r$.

Case ρ = b. For an arbitrary instance $\dfrac{[\Delta'], A'}{[\Delta', \Diamond A']}$ of sh-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 an instance of a propositional tautology, by Internalization Property 1.5.4 there exists a term $t_1(x_1)$ such that

$$\mathsf{JL_{CS}} \vdash r_1(k) : \Delta^{r_1} \;\to\; t_1\big(r_1(k)\big) : (\Delta^{r_1} \vee \neg x_m : \neg A^{r_1}) \ . \tag{3.21}$$

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

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

From this and (3.21) it follows by axiom schema sum and propositional reasoning that

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

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

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

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

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

is a realization function on $[\Delta, \Diamond_{2m} A]_i$. Thus for the identity substitution σ and this r, we have $\mathsf{JL_{CS}} \vdash \big([\Delta]_k, A\big)^{r_1}\sigma \to \big([\Delta, \Diamond_{2m} A]_i\big)^r$.

Case $\rho = 4$. 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 an instance of a propositional tautology, it follows from Internalization Property 1.5.4 that there is a term $s(x_1)$ such that

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

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

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

By axiom schema app and propositional reasoning,

$$\mathsf{JL}_{\mathsf{CS}} \vdash x_m : \neg A^{r_1} \;\rightarrow\; r_1(k) : (\neg x_m : \neg A^{r_1} \vee \Delta^{r_1}) \;\rightarrow\; \big(s(!\, x_m) \cdot r_1(k)\big) : \Delta^{r_1} \;\;,$$

which is propositionally equivalent to

$$\mathsf{JL}_{\mathsf{CS}} \vdash r_1(k) : (\neg x_m : \neg A^{r_1} \vee \Delta^{r_1}) \;\rightarrow\; \neg x_m : \neg A^{r_1} \vee \big(s(!\, x_m) \cdot r_1(k)\big) : \Delta^{r_1} \;\;.$$

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

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

is a realization function on $\Diamond_{2m} A, [\Delta]_i$. Thus for the identity substitution σ and this r, we have $\mathsf{JL}_{\mathsf{CS}} \vdash \big([\Diamond_{2m} A, \Delta]_k\big)^{r_1}\sigma \rightarrow \big(\Diamond_{2m} A, [\Delta]_i\big)^{r}$.

Case $\rho = \mathsf{5a}$**.** Consider an arbitrary instance $\dfrac{[\Delta'], \Diamond A'}{[\Delta', \Diamond A']}$ of sh-5a and 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 an instance of a propositional tautology $\Delta^{r_1} \rightarrow \Delta^{r_1} \vee \neg x_m : \neg A^{r_1}$ and Internalization Property 1.5.4 there exists a term $t_1(x_1)$ such that

$$\mathsf{JL}_{\mathsf{CS}} \vdash r_1(k) : \Delta^{r_1} \;\rightarrow\; t_1\big(r_1(k)\big) : (\Delta^{r_1} \vee \neg x_m : \neg A^{r_1}) \;\;. \qquad (3.22)$$

Similarly, for the instance $\neg x_m : \neg A^{r_1} \rightarrow \Delta^{r_1} \vee \neg x_m : \neg A^{r_1}$ of a propositional tautology there is a term $t_2(x_1)$ such that

$$\mathsf{JL}_{\mathsf{CS}} \vdash ?\, x_m : \neg x_m : \neg A^{r_1} \;\rightarrow\; t_2(?\, x_m) : (\Delta^{r_1} \vee \neg x_m : \neg A^{r_1}) \;\;.$$

From the instance $\neg x_m : \neg A^{r_1} \rightarrow ?\, x_m : \neg x_m : \neg A^{r_1}$ of j5, by propositional reasoning,

$$\mathsf{JL}_{\mathsf{CS}} \vdash \neg x_m : \neg A^{r_1} \;\rightarrow\; t_2(?\, x_m) : (\Delta^{r_1} \vee \neg x_m : \neg A^{r_1}) \;\;. \qquad (3.23)$$

It follows from (3.22) and (3.23) by axiom schema sum and propositional reasoning that

$$\mathsf{JL}_{\mathsf{CS}} \vdash r_1(k) : \Delta^{r_1} \vee \neg x_m : \neg A^{r_1} \;\rightarrow\; t : (\Delta^{r_1} \vee \neg x_m : \neg A^{r_1})$$

for $t := t_1\big(r_1(k)\big) + t_2(?\, x_m)$. The index i does not occur in either Δ or $\Diamond_{2m} A$ because $[\Delta, \Diamond_{2m} A]_i$ is properly annotated. Hence,

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

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

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

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 3.3.10, for the term $\mathrm{invposint}(x_1)$ constructed there and the term $\mathrm{t}_!(x_1)$ from Lemma 2.2.5,

$$\mathsf{JL_{CS}} \vdash \mathrm{invposint}(x_m) : \big(\neg\, \mathrm{t}_!(x_m) : x_m : \neg A^{r_1} \ \to\ \neg x_m : \neg A^{r_1}\big) \ .$$

Thus, by app and MP,

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

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

$$\mathsf{JL_{CS}} \vdash \neg\, \mathrm{t}_!(x_m) : x_m : \neg A^{r_1} \ \to \\ \big(\mathrm{invposint}(x_m) \cdot {?}\,\mathrm{t}_!(x_m)\big) : \neg x_m : \neg A^{r_1} \ . \tag{3.24}$$

By a propositional tautology and Internalization Property 1.5.4, for some ground term p_1

$$\mathsf{JL_{CS}} \vdash p_1 : \big(x_m : \neg A^{r_1} \ \to\ \Pi^{r_1} \vee \neg x_m : \neg A^{r_1} \ \to\ \Pi^{r_1}\big) \ .$$

Thus, by app,

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

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

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

which is propositionally equivalent to

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

From this and (3.24), by propositional reasoning,

$$\mathsf{JL_{CS}} \vdash r_1(i) : (\Pi^{r_1} \vee \neg x_m : \neg A^{r_1}) \;\rightarrow\; \big(\mathrm{invposint}(x_m) \cdot ?\,\mathrm{t}_!(x_m)\big) : \neg x_m : \neg A^{r_1} \;\vee\; s : \Pi^{r_1} \quad . \tag{3.25}$$

By the propositional tautology $\neg x_m{:}\neg A^{r_1} \rightarrow \Delta^{r_1} \vee \neg x_m{:}\neg A^{r_1}$ and Internalization Property 1.5.4 there is a term $t_3(x_1)$ such that from (3.25) and propositional reasoning,

$$\mathsf{JL_{CS}} \vdash r_1(i) : (\Pi^{r_1} \vee \neg x_m : \neg A^{r_1}) \;\rightarrow\; t_3\big(\mathrm{invposint}(x_m) \cdot ?\,\mathrm{t}_!(x_m)\big) : (\Delta^{r_1} \vee \neg x_m : \neg A^{r_1}) \;\vee\; s : \Pi^{r_1} \quad . \tag{3.26}$$

By the propositional tautology $\Delta^{r_1} \rightarrow \Delta^{r_1} \vee \neg x_m{:}\neg A^{r_1}$ and Internalization Property 1.5.4 there is a term $t_4(x_1)$ such that $\mathsf{JL_{CS}} \vdash r_1(k){:}\Delta^{r_1} \rightarrow t_4\big(r_1(k)\big) : (\Delta^{r_1} \vee \neg x_m : \neg A^{r_1})$. Therefore, by axiom schema sum,

$$\mathsf{JL_{CS}} \vdash r_1(k) : \Delta^{r_1} \;\rightarrow\; t : (\Delta^{r_1} \vee \neg x_m : \neg A^{r_1}) \tag{3.27}$$

for $t := t_3\big(\mathrm{invposint}(x_m) \cdot ?\,\mathrm{t}_!(x_m)\big) + t_4\big(r_1(k)\big)$. Similarly, by (3.26) and sum,

$$\mathsf{JL_{CS}} \vdash r_1(i) : (\Pi^{r_1} \vee \neg x_m : \neg A^{r_1}) \;\rightarrow\; t : (\Delta^{r_1} \vee \neg x_m : \neg A^{r_1}) \;\vee\; s : \Pi^{r_1} \quad .$$

Finally, by propositional reasoning with (3.27),

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

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

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

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

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

Case ρ = $\mathsf{5c}$. 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 subcase $\rho = \mathsf{5b}$ (cf. (3.25)) we find

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

Thus, by propositional reasoning,

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

By Internalization Property 1.5.4 there exists a term $s_1(x_1)$ such that

$$\mathsf{JL_{CS}} \vdash r_1(k) : \big(\Delta^{r_1} \;\vee\; r_1(i) : (\Pi^{r_1} \vee \neg x_m : \neg A^{r_1})\big) \;\to\; s_1\big(r_1(k)\big) : \big(\Delta^{r_1} \;\vee\; t_3 : \neg x_m : \neg A^{r_1} \;\vee\; s : \Pi^{r_1}\big) \;\; , \tag{3.28}$$

where $t_3 := \mathrm{invposint}(x_m) \cdot ?\, \mathrm{t}_!(x_m)$. By Lemma 2.2.3, there exists a term $\mathrm{fact}(x_1)$ such that

$$\mathsf{JL_{CS}} \vdash \mathrm{fact}(t_3) : (t_3 : \neg x_m : \neg A^{r_1} \;\to\; \neg x_m : \neg A^{r_1}) \;\; . \tag{3.29}$$

By propositional reasoning and Internalization Property 1.5.4, for some ground term p_2

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

From this and (3.29) by app and MP,

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

It follows by app and MP that

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

for $t_4 := p_2 \cdot \mathrm{fact}(t_3) \cdot s_1\big(r_1(k)\big)$. By propositional reasoning with (3.28),

$$\mathsf{JL}_{\mathsf{CS}} \vdash r_1(k) : \big(\Delta^{r_1} \vee r_1(i) : (\Pi^{r_1} \vee \neg x_m : \neg A^{r_1})\big) \rightarrow \\ t_4 : (\Delta^{r_1} \vee \neg x_m : \neg A^{r_1} \vee s : \Pi^{r_1}) \enspace .$$

The indices l and j do not occur in any of 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, we have

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

Theorem 3.3.12 (Realization). *Let a modal logic* ML *and a justification logic* JL *be chosen respectively from the 1st and the 2nd row of the same column of the following table:*

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

Note that the first row contains all 15 modal logics from the modal cube. Let CS *be an axiomatically appropriate and schematic constant specification for* JL*. Then* $(\mathsf{JL}_{\mathsf{CS}})^\circ = \mathsf{ML}$*. 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* $\mathsf{JL}_{\mathsf{CS}} \vdash A^r$*.*

Proof. The inclusion $(\mathsf{JL}_{\mathsf{CS}})^\circ \subseteq \mathsf{ML}$ follows from Lemma 1.4.3.

So we 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 3.3) 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 3.3.3. By Lemmas 3.3.4–3.3.9 the rules sh-id, sh-$\vee$, sh-$\wedge$, sh-ctr, sh-exch, sh-$\Box$, and sh-k, i.e., the shallow versions of all the rules of the sequent system SK for the modal logic K, are realizable in $\mathsf{JL}_{\mathsf{CS}}$. 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. By Lemma 3.3.11, sh-ρ is realizable in $\mathsf{JL}_{\mathsf{CS}}$. Again, the shallow versions of all the rules of ML are realizable in $\mathsf{JL}_{\mathsf{CS}}$. 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 3.2.11, for some properly annotated version A of A' there exists a realization function r on A such that $\mathsf{JL}_{\mathsf{CS}} \vdash A^r$. Clearly, $(A^r)^\circ = A'$. Hence, $A' \in (\mathsf{JL}_{\mathsf{CS}})^\circ$. $\square$

Theorem 3.3.12 involves only 15 of our 24 justification logics. Based on the fact that all justification counterparts of a modal logic are pairwise equivalent by Theorem 2.2.9, we now 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 our 24 justification logics.

We first need an auxiliary lemma: the requirement that different occurrences of $\Diamond$ be realized by distinct variables can be preserved under embeddings.

Lemma 3.3.13 (Embedding and Realization). *Let JL_1 and JL_2 be arbitrary justification logics (not necessarily extensions of J) over languages $\mathcal{L}_1$ and $\mathcal{L}_2$ respectively. Assume $\mathsf{JL}_1 \mathrel{\tilde{\subseteq}} \mathsf{JL}_2$ and $\mathsf{JL}_1 \vdash A^{r_1}$ for some properly annotated formula A and a realization function r_1 on A. Then there exists a 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 2.1.2 (1), $r_1(i)\omega$ is an $\mathcal{L}_2$-term whenever $r_1(i)$ is defined, r_2 is a prerealization function in language $\mathcal{L}_2$ on A. Whenever $r_2(2k)$ is defined, $r_1(2k) = x_k$ because 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 trivial to check by induction on the structure of A that $A^{r_1}\omega = A^{r_2}$. $\square$

Theorem 3.3.14 (Modular Realization). *Let ML be a modal logic and let JL be one of its justification counterparts. Let CS be an axiomatically appropriate and schematic constant specification for JL. Then $(\mathsf{JL}_{\mathsf{CS}})^\circ = \mathsf{ML}$. 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 $\mathsf{JL}_{\mathsf{CS}} \vdash A^r$.*

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 3.3.12.

Let $\mathsf{S5} \vdash A'$. By Theorem 3.3.12, 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 $\mathsf{S5}$ and let CS be an axiomatically appropriate and schematic constant specification for JL. By Theorem 2.2.9, $\mathsf{JL}_{\mathsf{CS}} \equiv \mathsf{JT45}$. Therefore, by Lemma 3.3.13, there exists a realization function r_2 on A such that $\mathsf{JL}_{\mathsf{CS}} \vdash A^{r_2}$. Clearly, $(A^{r_2})^\circ = A'$. Hence, $A' \in (\mathsf{JL}_{\mathsf{CS}})^\circ$. The inclusion $(\mathsf{JL}_{\mathsf{CS}})^\circ \subseteq \mathsf{S5}$ follows from Lemma 1.4.3.

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

By Theorems 3.3.14 and 2.2.9, two justification logics are equivalent if and only if they realize the same modal logic:

Corollary 3.3.15. *For two justification logics* JL_1 *and* JL_2 *and axiomatically appropriate and schematic constant specifications* CS_1 *and* CS_2 *for* JL_1 *and* JL_2 *respectively,*

$$\mathsf{JL}_{1\mathsf{CS}_1} \equiv \mathsf{JL}_{2\mathsf{CS}_2} \quad \Longleftrightarrow \quad (\mathsf{JL}_{1\mathsf{CS}_1})^\circ = (\mathsf{JL}_{2\mathsf{CS}_2})^\circ \enspace .$$

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 3.3.16. Alternatively, Theorem 3.3.14 could be proved using the fact that, by Corollary 3.3.17 below, each rule of the sequent system $\mathsf{S5}$ ($\mathsf{KB5}$) is realizable in every justification counterpart of the logic $\mathsf{S5}$ ($\mathsf{KB5}$). Theorem 3.3.14 could thus be proved similarly to Theorem 3.3.12, using Theorem 3.2.11.

Corollary 3.3.17 (Realizability of Modal Rules)**.** *The following rules are realizable:*

(1) the b*-rule in* $\mathsf{JL}_{\mathsf{CS}}$*, where* $\mathsf{JL} \supseteq \mathsf{JT5}$ *and* CS *an axiomatically appropriate and schematic constant specification for* JL*;*

(2) the $\mathsf{4}$*-rule in* $\mathsf{JL}_{\mathsf{CS}}$*, where either* $\mathsf{JL} \supseteq \mathsf{JT5}$ *or* $\mathsf{JL} \supseteq \mathsf{JB5}$ *and* CS *an axiomatically appropriate and schematic constant specification for* JL*;*

(3) the $\mathsf{5}$*-rules in* $\mathsf{JL}_{\mathsf{CS}}$*, where* $\mathsf{JL} \supseteq \mathsf{JB4}$ *and* CS *an axiomatically appropriate and schematic constant specification for* JL*;*

(4) the t*-rule in* $\mathsf{JL}_{\mathsf{CS}}$*, where either* $\mathsf{JL} \supseteq \mathsf{JDB4}$ *or* $\mathsf{JL} \supseteq \mathsf{JDB5}$ *and* CS *an axiomatically appropriate and schematic constant specification for* JL*.*

Proof. Using Lemma 2.2.6, we can prove realizability of the shallow rules in the respective justification logics by repeating the proof of Lemma 3.3.11, replacing each use of the axiom schema jb with $A \to ?\, s : \neg s : \neg A$, of the axiom schema $\mathsf{j4}$ for $\mathsf{JL} \supseteq \mathsf{JT5}$ with $s : A \to \mathrm{t}_!(s) : s : A$, of the axiom schema $\mathsf{j4}$ for $\mathsf{JL} \supseteq \mathsf{JB5}$ with $s : A \to \mathrm{t}'_!(s) : s : A$, and of the axiom schema $\mathsf{j5}$ with $\neg s : A \to \mathrm{t}_?(s) : \neg s : A$. Note also that the provability of axiom schema jt for $\mathsf{JL} \supseteq \mathsf{JDB4}$ and $\mathsf{JL} \supseteq \mathsf{JDB5}$ follows from Lemma 2.2.6. The realizability of the nested rules then follows from Lemma 3.2.10. $\square$

4 Gentzen Systems for Logics of Belief and Inversed Internalization

We first introduce Gentzen systems for logics of belief (i.e., for J, J4, J5, J45, JB, JB4, JB5, and JB45) and prove their completeness via syntactic cut elimination. We then use these systems to prove—for J and J4—a property that we call *inversed internalization*: for arbitrary t and A,

$$\vdash t : A \quad \text{implies} \quad \vdash A \ .$$

We also show that this property fails for all other logics of belief. To our knowledge, inversed internalization has not yet been studied in the literature. However, Kuznets proved a similar, but more restricted statement (cf. Lemma 3.4.10 in [Kuz08]). Note that proving inversed internalization is not straight-forward: if a formula $t : A$ is obtained from an MP-instance

$$\frac{B \quad B \to t : A}{t : A} \ ,$$

then it is not obvious—in the absence of the axiom schema jt—that formula A is provable.

Gentzen systems for logics of belief have not yet been present in the literature, except for [Mil12], where systems for J and J4 were introduced without proving their completeness. Artemov [Art01] presented a cut-free Gentzen system for JT4. However, instead of via syntactic cut elimination, he proved completeness via arithmetical semantics for JT4. Later, Artemov [Art02] presented a Gentzen system for the intuitionistic version of JT4 and proved completeness via a rather involved syntactic cut elimination procedure. A similar proof would probably also work for the classical version of JT4 (and for other extensions of JT). Here, however, we present a cut elimination proof for logics of belief that is—thanks to the absence of the axiom schema jt—significantly simpler and more standard than the one from [Art02].

Unfortunately, neither the cut elimination proof presented here nor Artemov's proof apply to JD and extensions. For them, finding cut-free Gentzen systems remains an open problem.

Beyond traditional Gentzen systems for justification logics, a number of developments have been going on recently: Renne [Ren06][1] presented tableaux systems for JT4 and S4LP (a hybrid logic that combines JT4 and S4). Further, tableaux and hypersequent systems for S4LP and S4LPN (a variant of S4LP that additionally contains negative introspection) have been introduced by Kurokawa [Kur09, Kur11]. Based on Kurokawa's work, Fitting [Fit12] presented nested sequent systems for the hybrid logics K+J (modal logic K combined with J) and S4LP; he also showed that the nested sequent systems and the tableaux systems are notational variants of each other. Poggiolesi [Pog10] introduced a hypersequent-like system for intuitionistic JT4.

Common to all the systems mentioned (including the systems presented here) is the fact that they—although cut-free—are not analytic, i.e., they do not enjoy the subformula property. See [Pog10] for a discussion on this matter.

4.1 Gentzen Systems for Logics of Belief

We start by defining Gentzen systems for logics of belief. Then we prove some structural properties of these systems such as invertibility of the implication rules and (partial) admissibility of contraction. Finally, we establish cut elimination and completeness.

Note that—in contrast to the situation in modal logics, where the connectives $\Diamond$ and $\Box$ are dual—for justification logics, there is no dual to $t : A$ and, as a consequence, justification formulas cannot be transformed into negation normal form. Therefore, the sequents we are using are two-sided.

A *sequent* is a pair $\Gamma \Rightarrow \Delta$, where Γ and Δ are multisets of justification formulas (in the language of the logic the system is formulated for). In the following, we use Γ and Δ (with or without sub- or superscripts) to denote multisets of formulas. We adopt common notation and write, e.g., Γ, Δ for $\Gamma \cup \Delta$, where $\cup$ denotes multiset union. We

[1] The article contains a mistake, a correction of which is under way but not yet published.

also write, e.g., Γ, A instead of $\Gamma, \{A\}$. The *corresponding formula* of a sequent $\Gamma \Rightarrow \Delta$ is $\bigwedge\Gamma \to \bigvee\Delta$, where $\bigwedge\Gamma$ ($\bigvee\Delta$) denotes the conjunction (disjunction) of all the formulas in Γ (Δ) in some standard order. See [TS96] for a general introduction on Gentzen systems.

For $\mathsf{JL} \in \{\mathsf{J}, \mathsf{J4}, \mathsf{J5}, \mathsf{J45}, \mathsf{JB}, \mathsf{JB4}, \mathsf{JB5}, \mathsf{JB45}\}$ and a constant specification CS for JL, we define the system $\mathsf{GJL_{CS}}$ (or GJL if CS is the total constant specification for JL). The axiom schemas of $\mathsf{GJL_{CS}}$ are:

$$(\mathsf{Ax})\ \frac{}{P, \Gamma \Rightarrow \Delta, P} \qquad (\bot)\ \frac{}{\bot, \Gamma \Rightarrow \Delta} \qquad (\mathsf{CS})\ \frac{c^n_{i_n} : \ldots : c^1_{i_1} : A \in \mathsf{CS}}{\Gamma \Rightarrow \Delta, c^n_{i_n} : \ldots : c^1_{i_1} : A}$$

The meta-variable P in (Ax) denotes either a proposition or a formula of the form $t : A$. The level n in (CS) is greater than zero.

The propositional rules of $\mathsf{GJL_{CS}}$ are:

$$(\mathsf{L}{\to})\ \frac{\Gamma \Rightarrow \Delta, A \quad B, \Gamma \Rightarrow \Delta}{A \to B, \Gamma \Rightarrow \Delta} \qquad (\mathsf{R}{\to})\ \frac{A, \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \to B}$$

The only structural rule of $\mathsf{GJL_{CS}}$ is a restricted version of right-contraction:

$$(\mathsf{RC})\ \frac{\Gamma \Rightarrow \Delta, t : A, t : A}{\Gamma \Rightarrow \Delta, t : A}$$

$\mathsf{GJL_{CS}}$ contains rules corresponding to the axiom schemas sum and app:

$$(+)\ \frac{\Gamma \Rightarrow \Delta, t : A}{\Gamma \Rightarrow \Delta, (s + t) : A} \qquad (+)\ \frac{\Gamma \Rightarrow \Delta, t : A}{\Gamma \Rightarrow \Delta, (t + s) : A}$$

$$(\cdot)\ \frac{\Gamma \Rightarrow \Delta, s : (A \to B) \quad \Gamma \Rightarrow \Delta, t : A}{\Gamma \Rightarrow \Delta, (s \cdot t) : B}$$

$\mathsf{GJL_{CS}}$ contains the following rule iff $\mathsf{j4}$ is an axiom schema of JL:

$$(!)\ \frac{\Gamma \Rightarrow \Delta, t : A}{\Gamma \Rightarrow \Delta, !t : t : A}$$

$\mathsf{GJL_{CS}}$ contains the following rule iff $\mathsf{j5}$ is an axiom schema of JL:

$$(?)\ \frac{t : A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, ?t : \neg t : A}$$

$\mathsf{GJL_{CS}}$ contains the following rule iff jb is an axiom schema of JL:

$$(\bar{?})\,\frac{\Gamma \Rightarrow \Delta, A}{\Gamma \Rightarrow \Delta, \bar{?}\,t : \neg t : \neg A}$$

Note that the language of the formulas in the above rules depends on the logic JL. Unless stated otherwise, from this point on, by $\mathsf{GJL_{CS}}$ we denote any Gentzen system for $\mathsf{JL} \in \{\mathsf{J}, \mathsf{J4}, \mathsf{J5}, \mathsf{J45}, \mathsf{JB}, \mathsf{JB4}, \mathsf{JB5}, \mathsf{JB45}\}$ and a constant specification CS for JL. Further, when we say that a formula A is provable in $\mathsf{GJL_{CS}}$, we mean that the sequent $\Rightarrow A$ is provable in $\mathsf{GJL_{CS}}$.

Remark 4.1.1 (Differences Against Other Systems)**.** The systems presented here have some structural differences against Artemov's original system for $\mathsf{JT4}$. Our systems do not have a left-contraction rule and only a restricted version of right-contraction. This facilitates the proof of cut elimination. Further, (Ax) is restricted to propositions and formulas of the form $t : A$. The reason for this choice is that we need the implication rules to be height-preserving invertible. Finally, instead of (CS), which also takes into account constant specifications, Artemov's system for $\mathsf{JT4}$ contains a rule

$$\frac{\Gamma \Rightarrow \Delta, A}{\Gamma \Rightarrow \Delta, c : A}$$

with A being an axiom.

The rule $(\cdot)$ is the one that prevents the above systems from being analytic: a formula A in the premise may not occur as a subformula in the conclusion. Another downside of the systems is that the rules $(+)$, $(\cdot)$, $(!)$, $(?)$, and $(\bar{?})$ are not invertible. For example, $(s+t) : A \Rightarrow (s+t) : A$ is provable (as an axiom), but $(s+t) : A \Rightarrow t : A$ may not be. As a consequence, the rule (RC) is not admissible, i.e., it cannot be absorbed into the other rules.

Despite these drawbacks, our Gentzen systems still have some advantages over Hilbert proof systems. For example, sequent derivations satisfy a kind of subterm property. The following lemma is the key ingredient to prove the partial conservativity results in Section 5.2. The lemma states that operations introduced by rules $(!)$, $(?)$, and $(\bar{?})$ cannot be eliminated and, hence, occur in the endsequent of a derivation. This was first observed by Milnikel [Mil12] for a Gentzen system for $\mathsf{J4}$.

Lemma 4.1.2 (Partial Subterm Property).
Let $\mathsf{JL} \in \{\mathsf{J}, \mathsf{JB}, \mathsf{J4}, \mathsf{J5}, \mathsf{J45}\}$ *and let* CS *be a constant specification for* JL*. If a* $\mathsf{GJL}_{\mathsf{CS}}$*-derivation* $\mathcal{D}$ *makes use of a rule* $(*)$*, where* $* \in \{!, ?, \bar{?}\}$*, then the endsequent of* $\mathcal{D}$ *contains the operation* $*$*.*

Proof. We say that an operation $*$ occurs at depth 0 in a formula A iff either A is of the form $t : B$ and $*$ occurs in t, or A is of the form $B \to C$ and $*$ occurs at depth 0 in B or in C. For example, $?$ occurs at depth 0 in $\neg x_1 : P_1 \to ?\, x_1 : \neg x_1 : P_1$. We say $*$ occurs at depth 0 in a sequent iff $*$ occurs at depth 0 in the corresponding formula of this sequent.

Since $* \in \{!, ?, \bar{?}\}$ occurs at depth 0 in the conclusion of $(*)$, the lemma follows immediately from the following **claim:** if an operation $* \in \{!, ?, \bar{?}\}$ occurs at depth 0 in a premise of a rule, then it occurs at depth 0 in the conclusion of that rule.

The claim is obvious for all the rules except for $(\bar{?})$, which is only present if JL is JB. In this case, $*$ must be $\bar{?}$. But since $\bar{?}$ occurs at depth 0 in the conclusion of $(\bar{?})$, the claim holds trivially. $\square$

Lemma 4.1.2 fails for proper extensions of JB. The following derivation serves as a counterexample for $\mathsf{GJB4}$ and $\mathsf{GJB45}$ (it makes use of the rule (!) but the endsequent does not contain !):

$$
(\cdot)\ \dfrac{\begin{array}{c}\mathcal{D}\\ \vdots \\ \Rightarrow p : (\neg x_1 : \neg ! c_1^1 : c_1^1 : A \to \top)\end{array} \qquad (\bar{?})\ \dfrac{(!)\ \dfrac{(\mathsf{CS})\ \dfrac{}{\Rightarrow c_1^1 : A}}{\Rightarrow ! c_1^1 : c_1^1 : A}}{\Rightarrow \bar{?}\, x_1 : \neg x_1 : \neg ! c_1^1 : c_1^1 : A}}{\Rightarrow c_2^1 \cdot (\bar{?}\, x_1) : \top}
$$

where $\mathcal{D}$ is a derivation of the indicated sequent for some ground term p, which exists by Internalization Property 1.5.4. A similar counterexample can easily be found for $\mathsf{GJB5}$.

We now prove some auxiliary lemmas that are needed to prove cut elimination. In the following, we write $\vdash_h \Gamma \Rightarrow \Delta$ if the sequent $\Gamma \Rightarrow \Delta$ has a $\mathsf{GJL}_{\mathsf{CS}}$-derivation of height at most $h \geq 0$.

Lemma 4.1.3 (Height-Preserving Invertibility of $(\mathsf{L}\to)$ and $(\mathsf{R}\to)$).
For every Gentzen system $\mathsf{GJL}_{\mathsf{CS}}$*,*

1. *if* $\vdash_h A \to B, \Gamma \Rightarrow \Delta$*, then* $\vdash_h \Gamma \Rightarrow \Delta, A$ *and* $\vdash_h B, \Gamma \Rightarrow \Delta$*;*

2. *if* $\vdash_h \Gamma \Rightarrow \Delta, A \to B$, *then* $\vdash_h A, \Gamma \Rightarrow \Delta, B$.

Proof. We first prove invertibility of $(\mathsf{L}\to)$ by induction on h. The **base case** $h = 0$ is trivial: since $A \to B$ cannot be the principal formula of an axiom, if $A \to B, \Gamma \Rightarrow \Delta$ is an axiom, so are $\Gamma \Rightarrow \Delta, A$ and $B, \Gamma \Rightarrow \Delta$.

For the **inductive step** we look at the last rule R of a derivation of $A \to B, \Gamma \Rightarrow \Delta$. Note that the only possibility for $A \to B$ being the principal formula of R is when R is $(\mathsf{L}\to)$. In this case, we can simply take the premises of R and are done. In every other case—including the case where R is $(\mathsf{L}\to)$ and $A \to B$ is not principal—we apply the induction hypothesis to the premise(s) of R and then apply R (twice). We exemplarily show the case $\mathsf{R} = (\cdot)$:

$$\frac{A \to B, \Gamma \Rightarrow \Delta', s : (C \to D) \quad A \to B, \Gamma \Rightarrow \Delta', t : C}{A \to B, \Gamma \Rightarrow \Delta', (s \cdot t) : D}$$

By induction hypothesis,

$$\begin{array}{lrcl}
\vdash_{h-1} & \Gamma & \Rightarrow & \Delta', s : (C \to D), A \ ; \\
\vdash_{h-1} & B, \Gamma & \Rightarrow & \Delta', s : (C \to D) \ ; \\
\vdash_{h-1} & \Gamma & \Rightarrow & \Delta', t : C, A \ ; \\
\vdash_{h-1} & B, \Gamma & \Rightarrow & \Delta', t : C \ .
\end{array}$$

From the first and third assertions, by applying $(\cdot)$, $\vdash_h \Gamma \Rightarrow \Delta', (s \cdot t) : D, A$. Similarly, from the second and fourth assertions, $\vdash_h B, \Gamma \Rightarrow \Delta', (s \cdot t) : D$.

Proving invertibility of $(\mathsf{R}\to)$ is dual to the case of $(\mathsf{L}\to)$. In the **base case**, if $\Gamma \Rightarrow \Delta, A \to B$ is an axiom, so is $A, \Gamma \Rightarrow \Delta, B$. For the **inductive step** we look at the last rule R of a derivation of $\Gamma \Rightarrow \Delta, A \to B$. If R is $(\mathsf{R}\to)$ with $A \to B$ as its principal formula, we take the premise of R and are done. In every other case, we apply the induction hypothesis to the premise(s) of R and then apply R. □

Lemma 4.1.4 (Height-Preserving Admissibility of Weakening)**.** *For every Gentzen system* $\mathsf{GJL}_{\mathsf{CS}}$*, if* $\vdash_h \Gamma \Rightarrow \Delta$ *, then* $\vdash_h \Gamma, \Gamma' \Rightarrow \Delta, \Delta'$.

Proof. By a straight-forward induction on h: if $\Gamma \Rightarrow \Delta$ is an axiom, then so is $\Gamma, \Gamma' \Rightarrow \Delta, \Delta'$; and if $\Gamma \Rightarrow \Delta$ was derived by a rule, we apply the induction hypothesis to its premise(s) and apply the rule again. □

Lemma 4.1.5 (Admissibility of Contraction). *For every Gentzen system* $\mathsf{GJL_{CS}}$*,*

1. *if* $\vdash A, A, \Gamma \Rightarrow \Delta$*, then* $\vdash A, \Gamma \Rightarrow \Delta$ *;*
2. *if* $\vdash \Gamma \Rightarrow \Delta, A, A$*, then* $\vdash \Gamma \Rightarrow \Delta, A$ *.*

Proof. For all h, A, Γ, and Δ we prove the statements

$$\begin{array}{lll} \vdash_h A, A, \Gamma \Rightarrow \Delta & \text{implies} & \vdash A, \Gamma \Rightarrow \Delta \quad \text{and} \\ \vdash_h \Gamma \Rightarrow \Delta, A, A & \text{implies} & \vdash \Gamma \Rightarrow \Delta, A \end{array}$$

by a simultaneous induction on the structure of A, with a subinduction on h.

Base case. Assume that A is either a proposition P_i or $\bot$. For the *first assertion*, if $A, A, \Gamma \Rightarrow \Delta$ is an axiom, so is $A, \Gamma \Rightarrow \Delta$. If $A, A, \Gamma \Rightarrow \Delta$ was derived by a rule R, then A cannot be principle in R. We consider the case where R is a single-premise rule (the two-premise case is similar):

$$\mathsf{R}\,\frac{A, A, \Gamma' \Rightarrow \Delta'}{A, A, \Gamma \Rightarrow \Delta}$$

By subinduction hypothesis, $\vdash A, \Gamma' \Rightarrow \Delta'$ and by an application of R, $\vdash A, \Gamma \Rightarrow \Delta$.

The *second assertion* is proved similarly: for A a proposition P_i or $\bot$, if $\Gamma \Rightarrow \Delta, A, A$ is an axiom, so is $\Gamma \Rightarrow \Delta, A$; if $\Gamma \Rightarrow \Delta, A, A$ was derived by a rule R, apply the subinduction hypothesis to the premise(s) of R and then apply R.

Inductive step. We first prove the *first assertion*. Assume A is of the form $t:B$. If $t:B, t:B, \Gamma \Rightarrow \Delta$ is an axiom, so is $t:B, \Gamma \Rightarrow \Delta$. If $t:B, t:B, \Gamma \Rightarrow \Delta$ was derived by a rule R, then—since $t:B$ cannot be principle—we apply the subinduction hypothesis to the premise(s) of R and then apply R.

Assume A is of the form $B \to C$. If $B \to C, B \to C, \Gamma \Rightarrow \Delta$ is an axiom or was derived by a rule in which $B \to C$ is not principal, we proceed as in the case $A = t:B$. So assume $B \to C, B \to C, \Gamma \Rightarrow \Delta$ was derived by $(\mathsf{L}\to)$ with principal formula $B \to C$:

$$(\mathsf{L}\to)\,\frac{B \to C, \Gamma \Rightarrow \Delta, B \qquad C, B \to C, \Gamma \Rightarrow \Delta}{B \to C, B \to C, \Gamma \Rightarrow \Delta}$$

By height-preserving invertibility of $(\mathsf{L}\!\rightarrow)$ (Lemma 4.1.3),

$$\vdash_{h-1} \Gamma \Rightarrow \Delta, B, B \qquad \text{and} \qquad \vdash_{h-1} C, C, \Gamma \Rightarrow \Delta \ .$$

By induction hypothesis, $\vdash \Gamma \Rightarrow \Delta, B$ and $\vdash C, \Gamma \Rightarrow \Delta$. By an application of $(\mathsf{L}\!\rightarrow)$, $\vdash B \rightarrow C, \Gamma \Rightarrow \Delta$.

Let's turn to the *second assertion.* If A is of the form $t : B$ and $\Gamma \Rightarrow \Delta, t\!:\!B, t\!:\!B$ is either an axiom or was derived by a rule in which $t : B$ is not principle, we proceed as in the case $A = t : B$ of the first assertion. If A is of the form $t\!:\!B$ and $\Gamma \Rightarrow \Delta, t\!:\!B, t\!:\!B$ was derived by a rule in which $t\!:\!B$ is principal, then we apply (RC)[2]. Assume A is of the form $B \rightarrow C$. If $\Gamma \Rightarrow \Delta, B \rightarrow C, B \rightarrow C$ is an axiom or was derived by a rule in which $B \rightarrow C$ is not principal, we proceed as in the case $A = t : B$ of the first assertion. So assume $\Gamma \Rightarrow \Delta, B \rightarrow C, B \rightarrow C$ was derived by $(\mathsf{R}\!\rightarrow)$ with principal formula $B \rightarrow C$:

$$(\mathsf{R}\!\rightarrow)\,\frac{B, \Gamma \Rightarrow \Delta, B \rightarrow C, C}{\Gamma \Rightarrow \Delta, B \rightarrow C, B \rightarrow C}$$

By height-preserving invertibility of $(\mathsf{R}\!\rightarrow)$ (Lemma 4.1.3),

$$\vdash^{h-1} B, B, \Gamma \Rightarrow \Delta, C, C \ .$$

By applying the induction hypothesis twice, $\vdash B, \Gamma \Rightarrow \Delta, C$. By an application of $(\mathsf{R}\!\rightarrow)$, $\vdash \Gamma \Rightarrow \Delta, B \rightarrow C$. $\square$

For any system $\mathsf{GJL_{CS}}$ the system $\mathsf{GJL^+_{CS}}$ is obtained from $\mathsf{GJL_{CS}}$ by adding the cut rule:

$$(\mathsf{cut})\,\frac{\Gamma \Rightarrow \Delta, A \quad A, \Gamma' \Rightarrow \Delta'}{\Gamma, \Gamma' \Rightarrow \Delta, \Delta'}$$

The formula A is called the *cut formula.*

Definition 4.1.6 (Cut Level and Rank)**.** The *rank* of a cut-instance with cut formula A is the number of implications in A. The *level* of a cut-instance (in a given derivation) is the sum of the heights of the derivations of its premises.

Lemma 4.1.7 (Reduction Lemma)**.** *Let $\mathcal{D}$ be a $\mathsf{GJL^+_{CS}}$-derivation with final inference a cut such that $\mathcal{D}$ contains no other cuts. Then there exists a (cut-free) $\mathsf{GJL_{CS}}$-derivation $\mathcal{D}'$ with the same endsequent.*

[2]Note that this increases the height of the derivation.

Proof. We proceed by induction on the rank of the final cut

$$(\mathsf{cut})\ \dfrac{\overset{\mathcal{D}_1}{\Gamma \Rightarrow \Delta, A} \qquad \overset{\mathcal{D}_2}{A, \Gamma' \Rightarrow \Delta'}}{\Gamma, \Gamma' \Rightarrow \Delta, \Delta'}$$

of $\mathcal{D}$, with a subinduction on its level. Note that $\mathcal{D}_1$ and $\mathcal{D}_2$ are cut-free. We follow [TS96] by distinguishing three cases:

1. Either $\mathcal{D}_1$ or $\mathcal{D}_2$ is an axiom.
2. $\mathcal{D}_1$ and $\mathcal{D}_2$ are not axioms and the cut formula A is not principal in at least one of the premises.
3. The cut formula A is principal in both premises.

Note that the base cases of the induction and the subinduction, as well as the inductive steps are proved implicitly in the case distinctions.

Case 1. We consider the following subcases:

1a. $\mathcal{D}_1$ is an instance of (Ax) or $(\bot)$ and the cut formula A is not principal in this instance. We consider the case (Ax) (the case $(\bot)$ is similar):

$$(\mathsf{cut})\ \dfrac{P, \Gamma'' \Rightarrow \Delta'', P, A \qquad \overset{\mathcal{D}_2}{A, \Gamma''' \Rightarrow \Delta'''}}{P, \Gamma'', \Gamma''' \Rightarrow \Delta'', P, \Delta'''}$$

The conclusion is itself an axiom and can be taken as $\mathcal{D}'$.

1b. $\mathcal{D}_1$ is an instance of (Ax) and the cut formula $A = P$ is principal in this instance:

$$(\mathsf{cut})\ \dfrac{P, \Gamma'' \Rightarrow \Delta, P \qquad \overset{\mathcal{D}_2}{P, \Gamma' \Rightarrow \Delta'}}{P, \Gamma'', \Gamma' \Rightarrow \Delta, \Delta'}$$

By Lemma 4.1.4, we can replace the cut by an application of weakening to the endsequent of $\mathcal{D}_2$.

1c. $\mathcal{D}_1$ is an instance of (CS) and the cut formula A is not principal in this instance. In this case, the conclusion of the cut is itself an instance of (CS) and can be taken as $\mathcal{D}'$.

1d. $\mathcal{D}_1$ is an instance of (CS) and the cut formula A is principal in this instance:

$$\mathsf{(cut)}\ \dfrac{\Gamma \Rightarrow \Delta, c^n_{i_n} : \ldots : c^1_{i_1} : B \qquad \begin{array}{c}\mathcal{D}_2\\ \vdots\\ c^n_{i_n} : \ldots : c^1_{i_1} : B, \Gamma' \Rightarrow \Delta'\end{array}}{\Gamma, \Gamma' \Rightarrow \Delta, \Delta'}$$

If $\mathcal{D}_2$ is also an axiom, then so is the conclusion of the cut (note that if $\mathcal{D}_2$ is (Ax), then the conclusion of the cut is either (Ax) or an instance of (CS)). If $\mathcal{D}_2$ is not an axiom, then the cut formula $A = c^n_{i_n} : \ldots : c^1_{i_1} : B$ cannot be principal in the last rule R of $\mathcal{D}_2$. We consider the case where R is a two-premise rule (the one-premise case is similar):

$$\mathsf{(cut)}\ \dfrac{\Gamma, \Rightarrow \Delta, A \qquad \mathsf{R}\ \dfrac{\begin{array}{c}\mathcal{D}_2'\\ \vdots\\ A, \Gamma'' \Rightarrow \Delta''\end{array} \quad \begin{array}{c}\mathcal{D}_2''\\ \vdots\\ A, \Gamma''' \Rightarrow \Delta'''\end{array}}{A, \Gamma' \Rightarrow \Delta'}}{\Gamma, \Gamma' \Rightarrow \Delta, \Delta'}$$

We can replace this cut by two cuts of equal rank but smaller level:

$$\mathsf{R}\ \dfrac{\mathsf{(cut)}\ \dfrac{\Gamma, \Rightarrow \Delta, A \quad \begin{array}{c}\mathcal{D}_2'\\ \vdots\\ A, \Gamma'' \Rightarrow \Delta''\end{array}}{\Gamma, \Gamma'' \Rightarrow \Delta, \Delta''} \qquad \mathsf{(cut)}\ \dfrac{\Gamma, \Rightarrow \Delta, A \quad \begin{array}{c}\mathcal{D}_2''\\ \vdots\\ A, \Gamma''' \Rightarrow \Delta'''\end{array}}{\Gamma, \Gamma''' \Rightarrow \Delta, \Delta'''}}{\Gamma, \Gamma' \Rightarrow \Delta, \Delta'}$$

Then, by subinduction hypothesis, the conclusions of the new cuts have cut-free derivations and, hence, so has $\Gamma, \Gamma' \Rightarrow \Delta, \Delta'$.

1e. $\mathcal{D}_2$ is an instance of (Ax) or $(\bot)$ and the cut formula is not principal in this instance. This case is similar to subcase 1a.

1f. $\mathcal{D}_2$ is an instance of (Ax) and the cut formula $A = P$ is principal in this instance:

$$\mathsf{(cut)}\ \dfrac{\begin{array}{c}\mathcal{D}_1\\ \vdots\\ \Gamma \Rightarrow \Delta, P\end{array} \quad P, \Gamma' \Rightarrow \Delta', P}{\Gamma, \Gamma' \Rightarrow \Delta, \Delta', P}$$

By Lemma 4.1.4, we can replace the cut by an application of weakening to the endsequent of $\mathcal{D}_1$.

1g. $\mathcal{D}_2$ is an instance of $(\bot)$ and the cut formula $A = \bot$ is principal in this instance:

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ \vdots\\ \Gamma \Rightarrow \Delta, \bot\end{array} \quad \bot, \Gamma' \Rightarrow \Delta'}{\Gamma, \Gamma' \Rightarrow \Delta, \Delta'}\ \textsf{(cut)}$$

If $\mathcal{D}_1$ is an axiom, then so is the conclusion of the cut. If $\mathcal{D}_1$ is not an axiom, then $\Gamma \Rightarrow \Delta, \bot$ was derived by a rule R in which the cut formula $A = \bot$ is not principal. We proceed similarly as in subcase 1d (i.e., replace the cut by two cuts of equal rank but smaller level).

1h. $\mathcal{D}_2$ is an instance of (CS). In this case, the conclusion of the cut it itself an instance of (CS).

Case 2. $\mathcal{D}_1$ and $\mathcal{D}_2$ are not axioms and the cut formula is not principal in at least one of the premises of the cut. Assume that the cut formula A is not principal in the last rule of $\mathcal{D}_1$ (the case of $\mathcal{D}_2$ is similar), and that $\mathcal{D}_1$ ends with a two-premise rule R (again, the one-premise case is similar):

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}_1'\\ \vdots\\ \Gamma'' \Rightarrow \Delta'', A\end{array} \quad \begin{array}{c}\mathcal{D}_1''\\ \vdots\\ \Gamma''' \Rightarrow \Delta''', A\end{array}}{\Gamma \Rightarrow \Delta, A}\ \mathsf{R} \qquad \begin{array}{c}\mathcal{D}_2\\ \vdots\\ A, \Gamma' \Rightarrow \Delta'\end{array}}{\Gamma, \Gamma' \Rightarrow \Delta, \Delta'}\ \textsf{(cut)}$$

We can replace this cut by two cuts of equal rank but smaller level:

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}_1'\\ \vdots\\ \Gamma'' \Rightarrow \Delta'', A\end{array} \quad \begin{array}{c}\mathcal{D}_2\\ \vdots\\ A, \Gamma' \Rightarrow \Delta'\end{array}}{\Gamma'', \Gamma' \Rightarrow \Delta'', \Delta'}\ \textsf{(cut)} \qquad \dfrac{\begin{array}{c}\mathcal{D}_1''\\ \vdots\\ \Gamma''' \Rightarrow \Delta''', A\end{array} \quad \begin{array}{c}\mathcal{D}_2\\ \vdots\\ A, \Gamma' \Rightarrow \Delta'\end{array}}{\Gamma''', \Gamma' \Rightarrow \Delta''', \Delta'}\ \textsf{(cut)}}{\Gamma, \Gamma' \Rightarrow \Delta, \Delta'}\ \mathsf{R}$$

Then, by subinduction hypothesis, the conclusions of the new cuts have cut-free derivations and, hence, so has $\Gamma, \Gamma' \Rightarrow \Delta, \Delta'$.

Case 3. The cut formula A is principal in both premises. Then we have the following situation:

$$(\mathsf{cut})\ \dfrac{(\mathsf{L}\!\to)\ \dfrac{\overset{\mathcal{D}_1'}{A,\Gamma \Rightarrow \Delta, B}}{\Gamma \Rightarrow \Delta, A \to B} \qquad (\mathsf{R}\!\to)\ \dfrac{\overset{\mathcal{D}_2'}{\Gamma' \Rightarrow \Delta', A} \quad \overset{\mathcal{D}_2''}{B,\Gamma' \Rightarrow \Delta'}}{A \to B, \Gamma' \Rightarrow \Delta'}}{\Gamma,\Gamma' \Rightarrow \Delta,\Delta'}$$

We replace this cut by two cuts of smaller rank:

$$(\mathsf{cut})\ \dfrac{\overset{\mathcal{D}_2'}{\Gamma' \Rightarrow \Delta', A} \qquad (\mathsf{cut})\ \dfrac{\overset{\mathcal{D}_1'}{A,\Gamma \Rightarrow \Delta, B} \quad \overset{\mathcal{D}_2''}{B,\Gamma' \Rightarrow \Delta'}}{A,\Gamma,\Gamma' \Rightarrow \Delta,\Delta'}}{\Gamma,\Gamma',\Gamma' \Rightarrow \Delta,\Delta',\Delta'}$$

By induction hypothesis, $A, \Gamma, \Gamma' \Rightarrow \Delta, \Delta'$ has a cut-free derivation and therefore, again by induction hypothesis, so does $\Gamma, \Gamma', \Gamma' \Rightarrow \Delta, \Delta', \Delta'$. By a number of left- and right-contractions, which are admissible by Lemma 4.1.5, we obtain a cut-free derivation of $\Gamma, \Gamma' \Rightarrow \Delta, \Delta'$. □

Using the above lemma, every $\mathsf{GJL}^+_{\mathsf{CS}}$-derivation can be transformed into a (cut-free) $\mathsf{GJL}_{\mathsf{CS}}$-derivation:

Theorem 4.1.8 (Cut Elimination)**.** *For every Gentzen system* $\mathsf{GJL}_{\mathsf{CS}}$*, if a sequent has a* $\mathsf{GJL}^+_{\mathsf{CS}}$*-derivation, then it has a (cut-free)* $\mathsf{GJL}_{\mathsf{CS}}$*-derivation.*

Proof. By induction on the height h of an arbitrary $\mathsf{GJL}^+_{\mathsf{CS}}$-derivation. If $h = 0$, then the derived sequent is an axiom and the derivation is trivially cut-free. If $h > 0$ and the derivation ends with a rule R other than cut, then, by induction hypothesis, the premise(s) of R have cut-free derivations and, hence, so has the conclusion of R. So assume $h > 0$ and the derivation ends with a cut with conclusion $\Gamma \Rightarrow \Delta$. Then, by induction hypothesis, the premises of this cut have cut-free derivations. Therefore, there exists a derivation of $\Gamma \Rightarrow \Delta$ that ends with a cut and contains no other cuts. By the Reduction Lemma 4.1.7, $\Gamma \Rightarrow \Delta$ has a (cut-free) $\mathsf{GJL}_{\mathsf{CS}}$-derivation. □

Theorem 4.1.9 (Completeness of $\mathsf{GJL_{CS}}$)**.** *Let*

$$\mathsf{JL} \in \{\mathsf{J}, \mathsf{J4}, \mathsf{J5}, \mathsf{J45}, \mathsf{JB}, \mathsf{JB4}, \mathsf{JB5}, \mathsf{JB45}\}$$

and let CS *be a constant specification for* JL. *We have*

$$\mathsf{GJL_{CS}} \vdash \ \Gamma \Rightarrow \Delta \qquad \Longleftrightarrow \qquad \mathsf{JL_{CS}} \vdash \bigwedge \Gamma \to \bigvee \Delta \ .$$

Proof. The soundness direction follows by a straight-forward induction on the height of a $\mathsf{GJL_{CS}}$-derivation.

The completeness direction is proved by induction on $\mathsf{JL_{CS}}$-proofs. In the **base case**, we show that every axiom of $\mathsf{JL_{CS}}$ is derivable in $\mathsf{GJL_{CS}}$.

Since weakening and contraction are admissible by Lemmas 4.1.4 and 4.1.5, the system $\mathsf{GJL_{CS}}$ is propositionally complete and, hence, every propositional axiom is derivable in it. See, e.g., the system **G2c** in [TS96] for details.

The following is a derivation of an an arbitrary instance of one of the sum-axiom schemas (the other one is similar):

$$\dfrac{\dfrac{t : B \Rightarrow t : B}{t : B \Rightarrow (s + t) : B}\,(+)}{\Rightarrow t : B \to (s + t) : B}\,(\mathsf{R}\!\to)$$

The following is a derivation of an arbitrary app-instance:

$$\dfrac{\dfrac{\dfrac{s : (B \to C), t : B \Rightarrow s : (B \to C) \quad s : (B \to C), t : B \Rightarrow t : B}{s : (B \to C), t : B \Rightarrow (s \cdot t) : C}\,(\cdot)}{s : (B \to C) \Rightarrow t : B \to (s \cdot t) : C}\,(\mathsf{R}\!\to)}{\Rightarrow s : (B \to C) \to (t : B \to (s \cdot t) : C)}\,(\mathsf{R}\!\to)$$

The following is a derivation of an arbitrary $\mathsf{j4}$-instance:

$$\dfrac{\dfrac{t : B \Rightarrow t : B}{t : B \Rightarrow\, !t : t : B}\,(!)}{\Rightarrow t : B \to\, !t : t : B}\,(R\!\to)$$

The following is a derivation of an arbitrary $\mathsf{j5}$-instance (recall that

$\neg A = A \to \bot$):

$$
\dfrac{\dfrac{\dfrac{t : B \Rightarrow t : B}{\Rightarrow t : B, ?\,t : \neg t : B}\,(?) \qquad \bot \Rightarrow ?\,t : \neg t : B}{\neg t : B \Rightarrow ?\,t : \neg t : B}\,(\mathsf{L}\!\to)}{\Rightarrow \neg t : B \to ?\,t : \neg t : B}\,(\mathsf{R}\!\to)
$$

Given a derivation $\mathcal{D}$ of an arbitrary sequent $B \Rightarrow B$, the following is a derivation of an arbitrary jb-instance:

$$
\dfrac{\dfrac{\begin{array}{c}\mathcal{D}\\ \vdots \\ B \Rightarrow B\end{array}}{B \Rightarrow \bar{?} : \neg t : \neg B}\,(\bar{?})}{\Rightarrow B \to \bar{?} : \neg t : \neg B}\,(\mathsf{R}\!\to)
$$

If $c^n_{i_n} : \ldots : c^1_{i_1} : B$ is the conclusion of the $\mathsf{iAN_{CS}}$-rule, then the sequent $\Rightarrow c^n_{i_n} : \ldots : c^1_{i_1} : B$ is derivable by the (CS)-rule.

Inductive step. If a formula A is the conclusion of MP with premises B and $B \to A$, by induction hypothesis, there exist $\mathsf{GJL_{CS}}$-derivations of $\Rightarrow B$ and $\Rightarrow B \to A$. By invertibility of $(\mathsf{R}\!\to)$ (Lemma 4.1.3), the sequent $B \Rightarrow A$ is derivable in $\mathsf{GJL_{CS}}$. Therefore, by admissibility of cut (Theorem 4.1.8), there exists a $\mathsf{GJL_{CS}}$-derivation of $\Rightarrow A$. □

4.2 Inversed Internalization

We now use the Gentzen systems of the previous section to prove that J and $\mathsf{J4}$ satisfy the property of *inversed internalization*: if $t : A$ is provable, then so is A. Note that inversed internalization trivially holds for logics that prove the axiom schema jt. However, it fails for some extensions of $\mathsf{J5}$ and JB, as we show in Lemma 4.2.3.

Inversed internalization for J and $\mathsf{J4}$ is a consequence of the following lemma. The name *stripping lemma* is due to Artemov [Art02]; however, he uses it to denote a different statement. Note that in Lemma 4.2.1 we are only working with sequents of the form $\Rightarrow \Delta$ because, in the context of inversed internalization, we are interested in formulas/sequents where (sub)formulas of type $t : A$ occur only positively.

Lemma 4.2.1 (Stripping Lemma)**.** *Let* $\mathsf{JL} \in \{\mathsf{J}, \mathsf{J4}\}$*, let* CS *be a downward closed constant specification for* JL*, let* Δ *be a (possibly empty) multiset of formulas of the form* $t : A$ *(in the language of* JL*) and let* $n > 0$*. If a sequent*

$$\Rightarrow \Delta, s_1 : B_1, \ldots, s_n : B_n \tag{4.1}$$

is derivable in $\mathsf{GJL_{CS}}$*, then so is*

$$\Rightarrow \Delta, B_1, \ldots, B_n \enspace .$$

Proof. By induction on the height of a $\mathsf{GJL_{CS}}$-derivation of (4.1).

In the **base case**, $\Rightarrow \Delta, s_1{:}B_1, \ldots, s_n{:}B_n$ must be the conclusion of the rule (CS). If the principal formula is one of Δ, then $\Rightarrow \Delta, B_1, \ldots, B_n$ is another instance of (CS). So assume—without loss of generality—that $s_1 : B_1 = c^l_{i_l} : \ldots : c^1_{i_1} : C_1 \in \mathsf{CS}$ is the principal formula:

$$(\mathsf{CS})\,\frac{}{\Rightarrow \Delta, \underbrace{c^l_{i_l} : \ldots : c^1_{i_1} : C_1}_{s_1 : B_1},\ \ldots\ , s_n : B_n}$$

If $l > 1$, then, because CS is downward closed,

$$\Rightarrow \Delta, \underbrace{c^{l-1}_{i_{l-1}} : \ldots : c^1_{i_1} : C_1}_{B_1}, \ldots, B_n$$

is also an instance of (CS). If $l = 1$, by completeness of $\mathsf{GJL_{CS}}$ (Theorem 4.1.9), the JL-axiom $C_1 = B_1$ is provable in $\mathsf{GJL_{CS}}$ and, by admissibility of weakening (Lemma 4.1.4), so is the sequent $\Rightarrow \Delta, B_1, \ldots, B_n$.

For the **inductive step** we look at the last rule R of a derivation of (4.1). Note that R cannot be ($\mathsf{L}{\rightarrow}$) or ($\mathsf{R}{\rightarrow}$).

We first assume that none of $s_1 : B_1, \ldots, s_n : B_n$ is principle in R and that R has one premise (the two-premise case being analogous):

$$\mathsf{R}\,\frac{\Rightarrow \Delta', s_1 : B_1, \ldots, s_n : B_n}{\Rightarrow \Delta, s_1 : B_1, \ldots, s_n : B_n}$$

By induction hypothesis, $\Rightarrow \Delta', B_1, \ldots, B_n$ is provable in $\mathsf{GJL_{CS}}$ and, by an application of R, so is $\Rightarrow \Delta, B_1, \ldots, B_n$.

In the following, we assume—without loss of generality—that $s_1 : B_1$ is principal in R.

Assume that R is (RC):

$$(\mathsf{RC})\ \frac{\Rightarrow \Delta, s_1 : B_1, s_1 : B_1, \ldots, s_n : B_n}{\Rightarrow \Delta, s_1 : B_1, \ldots, s_n : B_n}$$

By induction hypothesis, $\vdash \Rightarrow \Delta, B_1, B_1, \ldots, B_n$ and by admissibility of right-contraction (Lemma 4.1.5), $\vdash \Rightarrow \Delta, B_1, \ldots, B_n$.

Assume that R is one of the $(+)$-rules, e.g.:

$$(+)\ \frac{\Rightarrow \Delta, s'_1 : B_1, \ldots, s_n : B_n}{\Rightarrow \Delta, (s'_1 + s''_2) : B_1, \ldots, s_n : B_n}$$

The statement of the lemma follows by induction hypothesis.

Assume that R is $(\cdot)$:

$$(\cdot)\ \frac{\Rightarrow \Delta, s'_1 : (C_1 \to B_1), \ldots, s_n : B_n \quad \Rightarrow \Delta, s''_1 : C_1, \ldots, s_n : B_n}{\Rightarrow \Delta, (s'_1 \cdot s''_1) : B_1, \ldots, s_n : B_n}$$

By applying the induction hypothesis to the left premise and by invertibility of the rule $(\mathsf{R}\!\to)$ (Lemma 4.1.3),

$$\vdash\ C_1 \Rightarrow \Delta, B_1, \ldots, B_n\ . \tag{4.2}$$

By applying the induction hypothesis to the right premise,

$$\vdash \Rightarrow \Delta, C_1, \ldots, B_n\ . \tag{4.3}$$

By admissibility of cut (Theorem 4.1.8) on (4.2) and (4.3),

$$\vdash \Rightarrow \Delta, B_2, \ldots, B_n, \Delta, B_1, \ldots, B_n\ .$$

Since right-contraction is admissible by Lemma 4.1.5,

$$\vdash \Rightarrow \Delta, B_1, \ldots, B_n\ .$$

Finally, provided $\mathsf{JL} = \mathsf{J4}$, assume that R is $(!)$:

$$(!)\ \frac{\Rightarrow \Delta, s'_1 : C_1, s_2 : B_2 \ldots, s_n : B_n}{\Rightarrow \Delta, !\, s'_1 : s'_1 : C_1, s_2 : B_2, \ldots, s_n : B_n}$$

Note that $\Delta, s'_1{:}C_1$ is a multiset of formulas of the form $t{:}A$. Therefore, by induction hypothesis, $\vdash \Rightarrow \Delta, s'_1 : C_1, B_2, \ldots, B_n$, which is what we want. Note that, in this case, it is crucial that the induction hypothesis allows not to strip s'_1 from $s'_1 : C_1$. $\square$

Inversed internalization for J and $\mathsf{J4}$ now follows from Lemma 4.2.1 and Theorem 4.1.9.

Corollary 4.2.2 (Inversed Internalization)**.** *For* $\mathsf{JL} \in \{\mathsf{J}, \mathsf{J4}\}$*, a downward closed constant specification* CS *for* JL*, and for arbitrary terms and formulas* t *and* A *respectively,*

$$\mathsf{JL}_{\mathsf{CS}} \vdash t : A \quad \Longrightarrow \quad \mathsf{JL}_{\mathsf{CS}} \vdash A \ .$$

Lemma 4.2.3 (Inversed Internalization Fails)**.**
Let $\mathsf{JL} \in \{\mathsf{J5}, \mathsf{J45}, \mathsf{JB5}, \mathsf{JB45}, \mathsf{JD5}, \mathsf{JD45}, \mathsf{JB}, \mathsf{JB4}\}$*. We have*

$$\mathsf{JL} \vdash t : A \quad \not\Longrightarrow \quad \mathsf{JL} \vdash A$$

for some term t *and formula* A*.*

Proof. We give counterexamples for all the logics.

By Lemma 2.2.3 there exists a term $\mathrm{fact}(x_1)$ such that for any formula A

$$\mathsf{J5} \vdash \mathrm{fact}(x_1) : (x_1 : A \to A) \ .$$

Assume to the contrary that $x_1 : A \to A$ is provable in $\mathsf{J5}$. Then, by Lemma 1.4.3, $\mathsf{K5} \vdash \Box A^\circ \to A^\circ$, which is well-known to be false. The same argument works for $\mathsf{J45}$, $\mathsf{JB5}$, $\mathsf{JB45}$, $\mathsf{JD5}$, and $\mathsf{JD45}$.

We now turn to JB and $\mathsf{JB4}$. Take a propositional tautology B. By axiom schema jb and MP, $\bar{?}:\neg t : \neg B$ is provable in JB for any term t. Assume to the contrary that $\neg t : \neg B$ is provable in JB. Then, by Lemma 1.4.3, $\mathsf{B} \vdash \Diamond B^\circ$. But any non-reflexive one-world Kripke model is a (symmetric) B-countermodel for $\Diamond B^\circ$. So we have a contradiction. The same argument works for $\mathsf{JB4}$ since any non-reflexive one-world Kripke model is also transitive and, hence, a $\mathsf{B4}$-countermodel for $\Diamond B^\circ$. $\square$

Note that the above argument for JB and $\mathsf{JB4}$ does not work for JDB because $\Diamond B^\circ$ *is* provable in modal logic KDB.

Figure 4.1 summarizes the above results on inversed internalization. A question mark means that it is not known whether the property holds for the given logic. Note that logics that prove jt are not listed because, for them, inversed internalization holds trivially.

Since the Internalization Property 1.5.2 also holds with hypotheses, it is a natural question whether inversed internalization holds with hypotheses, i.e., whether $s_1 : B_1, \ldots, s_m : B_m \vdash t : A$ implies $B_1, \ldots, B_m \vdash A$. The following lemma shows that this is the case for J.

J	JB	J5	J4	J45	JB4	JB5	JB45	JD	JDB	JD4	JD5	JD45
$\surd$	–	–	$\surd$	–	–	–	–	?	?	?	–	–

Figure 4.1: Summary of Inversed Internalization

Lemma 4.2.4 (Inversed Internalization with Hypotheses)**.** *Let* CS *be a downward closed constant specification for* J*. Then for every* $m \geq 0$ *and for arbitrary terms* $s_1, \dots, s_m, t$ *and formulas* $B_1, \dots, B_m, A$ *respectively,*

$$s_1 : B_1, \dots, s_m : B_m \vdash_{\mathsf{J}_{\mathsf{CS}}} t : A \quad \Longrightarrow \quad B_1, \dots, B_m \vdash_{\mathsf{J}_{\mathsf{CS}}} A \ .$$

Proof. We show that in $\mathsf{GJ}_{\mathsf{CS}}$, for all numbers $m \geq 0$ and $n > 0$,

$$\vdash s_1 : B_1, \dots, s_m : B_m \Rightarrow t_1 : A_1, \dots, t_n : A_n \tag{4.4}$$

implies

$$\vdash B_1, \dots, B_m \Rightarrow A_1, \dots, A_n \ . \tag{4.5}$$

The statement of the lemma then follows from completeness of $\mathsf{GJ}_{\mathsf{CS}}$ (Theorem 4.1.9) and the Deduction Theorem (Lemma 1.5.3). We proceed by induction on the height of a $\mathsf{GJ}_{\mathsf{CS}}$-derivation (4.4).

Base case. First assume that (4.4) is an instance of (CS) and—without loss of generality—that $t_1 : A_1$ is the principal formula. Then either A_1 is an axiom and thus (4.5) is provable by completeness (Theorem 4.1.9) and admissibility of weakening (Lemma 4.1.4); or, since CS is downward closed, $A_1 \in \mathsf{CS}$ and, hence, (4.5) is an instance of (CS). If (4.4) is an instance of (Ax), then the sequent in (4.5) has the form $B_1, \dots, B_m \Rightarrow \dots, B_i, \dots$ for some $1 \leq i \leq m$, and thus is provable by completeness (Theorem 4.1.9) and admissibility of weakening (Lemma 4.1.4).

For the **inductive step** we look at the last rule R of a derivation (4.4). Note that R cannot be ($\mathsf{L}\!\rightarrow$) or ($\mathsf{R}\!\rightarrow$). In the following, we assume—without loss of generality—that $t_1 : A_1$ is principal in R.

Assume that R is (RC):

$$(\mathsf{RC})\ \frac{s_1 : B_1, \dots, s_m : B_m \Rightarrow t_1 : A_1, t_1 : A_1, \dots, t_n : A_n}{s_1 : B_1, \dots, s_m : B_m \Rightarrow t_1 : A_1, \dots, t_n : A_n}$$

By induction hypothesis, $\vdash\ B_1, \dots, B_m \Rightarrow A_1, A_1, \dots, A_n$ and by admissibility of right-contraction (Lemma 4.1.5),

$$\vdash\ B_1, \dots, B_m \Rightarrow A_1, \dots, A_n\ .$$

If R is one of the (+)-rules, the statement follows by induction hypothesis.

Assume that R is $(\cdot)$ with premises

$$s_1 : B_1, \dots, s_m : B_m \Rightarrow t_1' : (C_1 \to A_1), \dots, t_n : A_n\ , \tag{4.6}$$
$$s_1 : B_1, \dots, s_m : B_m \Rightarrow t_1'' : C_1, \dots, t_n : A_n \tag{4.7}$$

and conclusion

$$s_1 : B_1, \dots, s_m : B_m \Rightarrow (t_1' \cdot t_1'') : A_1, \dots, t_n : A_n\ .$$

By applying the induction hypothesis to (4.6),

$$\vdash B_1, \dots, B_m \Rightarrow C_1 \to A_1, \dots, A_n$$

and by invertibility of the rule $(\mathsf{R}\!\to)$ (Lemma 4.1.3),

$$\vdash C_1, B_1, \dots, B_m \Rightarrow A_1, \dots, A_n\ . \tag{4.8}$$

By applying the induction hypothesis to (4.7),

$$\vdash B_1, \dots, B_m \Rightarrow C_1, \dots, A_n\ . \tag{4.9}$$

By admissibility of cut (Theorem 4.1.8) on (4.8) and (4.9),

$$\vdash B_1, \dots, B_m, B_1, \dots, B_m \Rightarrow A_2, \dots, A_n, A_1, \dots, A_n$$

and by a number of left- and right-contractions, which are admissible by Lemma 4.1.5,

$$\vdash B_1, \dots, B_m, \Rightarrow A_1, \dots, A_n\ . \qquad \square$$

An easy counterexample shows that Lemma 4.2.4 does not hold for extensions of $\mathsf{J4}$: we have $c_i^1 : P_1 \vdash_{\mathsf{J4}} !\,c_i^1 : c_i^1 : P_1$ by axiom schema $\mathsf{j4}$ and the Deduction Theorem (Lemma 1.5.3), but $P_1 \nvdash_{\mathsf{J4}} c_i^1 : P_1$.

5 Conservativity

In this chapter, we show that among our 24 justification logics some extensions are conservative and others are not—unfortunately, for logics of belief only a partial result is proved. Recall the following definition:

Definition 5.0.5 (Conservative Extension)**.** A logic (i.e., a set of formulas, cf. Definition 1.1.1) L_2 is called an *extension* of a logic L_1 if $\mathsf{L}_1 \subseteq \mathsf{L}_2$. If L_2 is an extension of L_1 and for any formula A in the language of L_1, $\mathsf{L}_2 \vdash A$ implies $\mathsf{L}_1 \vdash A$, then L_2 is called a *conservative extension* of L_1.

The approach in Section 5.1 that applies to logics of knowledge is due to Fitting [Fit08]; we just modify it slightly such that it also covers extensions of JTB. The partial conservativity result for logics of belief in Section 5.2 was originally proved by Milnikel [Mil12] for the extension J4 of J. Using the Gentzen systems from Chapter 4, we generalize his result to cover also JB, J5, and J45. In Section 5.3, we show that extensions of logics of consistent belief, i.e., logics that contain axiom schemas jd but lack jt, are *not* conservative. The first such result, namely that JD4 is not a conservative extensions of JD, was found by Milnikel [Mil12].

5.1 Conservativity for Logics of Knowledge

Until now, the set taut of propositional axiom schemas was not specified, any finite complete set of axiom schemas would work (cf. Figure 1.5). However, Fitting's [Fit08] method for proving conservativity for logics of knowledge relies on a specific choice of propositional axiom schemas. Moreover, it is required that propositional variants of jt are also axiom schemas. We show later that the logics satisfying these additional requirements are equivalent to the original ones.

For $\mathsf{JL} \supseteq \mathsf{JT}$ let $\mathsf{JL}^\star$ denote the logic that, in addition to the axiom schemas of JL, contains the following axiom schemas:

$$A \to A \ , \tag{5.1}$$

$$A \to \neg\neg A \ , \tag{5.2}$$

$$A \to \neg t : \neg A \ , \tag{5.3}$$

$$(A \to B) \to (t : A \to B) \ . \tag{5.4}$$

Note that (5.1) and (5.2) are tautological and, hence, may already be axiom schemas of JL. Axiom schemas (5.3) and (5.4) are consequences of jt (cf. Lemma 5.1.2 for a proof of this).

We adopt the notation from [Fit08], which allows for an elegant formulation of a conservativity statement that covers all the extensions of JT. Since we do not analyze weak logics without $+$ or $\cdot$, our notation is a bit simpler than the one in [Fit08]. For $S \subseteq \{!, ?, \bar{?}\}$, by $K(S)$ (the K stands for *knowledge*) we denote the smallest extension of $\mathsf{JT}^\star$ that contains the defining axiom schema for every operation in S. For example, $K(?, \bar{?})$ denotes $\mathsf{JTB5}^\star$ (for simplicity, we write $K(?, \bar{?})$ instead of $K(\{?, \bar{?}\})$, and so on).

Theorem 5.1.1 generalizes Fitting's conservativity result from [Fit08] in the sense that it includes extensions of $\mathsf{JB}^\star$. It shows, for example, that the following logics are conservative extensions of $\mathsf{JT}^\star$: $\mathsf{JT4}^\star$, $\mathsf{JTB}^\star$, $\mathsf{JT5}^\star$, $\mathsf{JTB5}^\star$, $\mathsf{JT45}^\star$, $\mathsf{JTB45}^\star$, $\mathsf{JTB4}^\star$. Note that the theorem does not cover those justification counterparts of $\mathsf{S5}$ that do not contain jt as an axiom schema because they are not extensions of JT. For example, $\mathsf{JDB4}$ is not an extension of JT because—although provable by Lemma 2.2.6—jt is not an axiom schema of $\mathsf{JDB4}$ and, hence, no formula $c_i^1 : (t : A \to A)$ is a theorem of $\mathsf{JDB4}$.

Also note that Theorem 5.1.1 is formulated for logics with total constant specifications. The proof—in general—would not work with restricted constant specifications because it requires a certain flexibility from the constants. For example, a constant $c_{i_1}^1$ justifying an axiom $t : A \to\, !t : t : A$ in $\mathsf{JT4}^\star$ must also justify the (propositional) axiom $t : A \to t : A$ in $\mathsf{JT}^\star$.

Theorem 5.1.1 (Conservativity for Logics of Knowledge)**.** *Let $S_1 \subsetneq S_2 \subseteq \{!, ?, \bar{?}\}$. Then $K(S_2)$ is a conservative extension of $K(S_1)$.*

Proof. Let $S := S_2 \setminus S_1$. For a formula A, by $A^{\#S}$ we denote the formula obtained from A by dropping all the terms that contain operations from S. Formally,

$$
\begin{aligned}
(P_i)^{\#S} &:= P_i \ ; \\
\bot^{\#S} &:= \bot \ ; \\
(A \to B)^{\#S} &:= A^{\#S} \to B^{\#S} \ ; \\
(t : A)^{\#S} &:= \begin{cases} A^{\#S} & \text{if } t \text{ contains an operation from } S, \\ t : A^{\#S} & \text{otherwise} \ . \end{cases}
\end{aligned}
$$

Note that if A is an $\mathcal{L}(S_2)$-formula, then $A^{\#S}$ is an $\mathcal{L}(S_1)$-formula. We first show that if A is an axiom of $K(S_2)$, then $A^{\#S}$ is an axiom of $K(S_1)$. This requires a simple but tedious case analysis:

If A is an instance of a propositional axiom schema, then so is $A^{\#S}$.

If A is an instance $t : B \to B$ or $(B \to C) \to (t : B \to C)$ of jt or of (5.4) respectively, then there are two possibilities:

- if t contains an operation from S, then $A^{\#S}$ is an instance of (5.1);
- if t does not contain an operation from S, then $A^{\#S}$ is another instance of jt or of (5.4).

If A is an instance $B \to \neg t{:}\neg B$ of (5.3), then there are two possibilities:

- if t contains an operation from S, then $A^{\#S}$ is an instance of (5.2);
- if t does not contain an operation from S, then $A^{\#S}$ is another instance of (5.3).

If A is an instance $t{:}B \to (s+t){:}B$ of sum, we distinguish the following cases:

- if t contains an operation from S, then $A^{\#S}$ is an instance of (5.1);
- if s but not t contains an operation from S, then $A^{\#S} = t{:}B^{\#S} \to B^{\#S}$ is an instance of jt;
- if neither t nor s contain an operation from S, then $A^{\#S} = t : B^{\#S} \to (s + t) : B^{\#S}$ is another instance of sum.

If A is an instance $t{:}(B \to C) \to (s{:}B \to (t{\cdot}s){:}C)$ of app, we distinguish the following cases:

- if both t and s contain an operation from S, then $A^{\#S}$ is an instance of (5.1);

- if s but not t contains an operation from S, then $A^{\#S} = t : (B^{\#S} \to C^{\#S}) \to (B^{\#S} \to C^{\#S})$ is an instance of jt;
- if t but not s contains an operation from S, then $A^{\#S} = (B^{\#S} \to C^{\#S}) \to (s : B^{\#S} \to C^{\#S})$ is an instance of (5.4);
- if neither t nor s contain an operation from S, then $A^{\#S} = t : (B^{\#S} \to C^{\#S}) \to (s : B^{\#S} \to (t \cdot s) : C^{\#S})$ is another instance of app.

If A is an instance $B \to \bar{?}\, t : \neg t : \neg B$ of jb, then there are three cases to consider:

- if t contains an operation from S, then $A^{\#S} = B^{\#S} \to \neg\neg B^{\#S}$ is an instance of (5.2);
- if t does not contain an operation from S and $\bar{?} \notin S$, then $A^{\#S} = B^{\#S} \to \bar{?}\, t : \neg t : \neg B^{\#S}$ is another instance of jb;
- if t does not contain an operation from S and $\bar{?} \in S$, then $A^{\#S} = B^{\#S} \to \neg t : \neg B^{\#S}$ is an instance of (5.3).

If A is an instance $t : B \to\, !\, t : t : B$ of $\mathsf{j4}$, we distinguish the following cases:

- if t contains an operation from S, then $A^{\#S} = B^{\#S} \to B^{\#S}$ is an instance of (5.1);
- if t does not contain an operation from S and $! \notin S$, then $A^{\#S} = t : B^{\#S} \to\, !\, t : t : B^{\#S}$ is another instance of $\mathsf{j4}$;
- if t does not contain an operation from S and $! \in S$, then $A^{\#S} = t : B^{\#S} \to t : B^{\#S}$ is an instance of (5.1).

The case where A is an instance of $\mathsf{j5}$ is similar.

Proving the theorem is now straightforward. Let A be an arbitrary $\mathcal{L}(S_1)$-formula provable in $K(S_2)$. Then $A = A^{\#S}$. Consider a proof of A in $K(S_2)$ and replace each line C of that proof with $C^{\#S}$. As we have shown, every axiom of $K(S_2)$ is turned into an axiom of $K(S_1)$ and, consequently, instances of iAN in $K(S_2)$ are turned into instances of iAN in $K(S_1)$. Because instances of MP are also turned into instances of MP, we have obtained a proof of $A = A^{\#S}$ in $K(S_1)$. $\square$

Lemma 5.1.2. *Let* $\mathsf{JL} \supseteq \mathsf{JT}$. *Then* $\mathsf{JL}^\star$ *and* JL *are equivalent.*

Proof. Since both JL and $\mathsf{JL}^\star$ satisfy conditions (1) and (2) of Theorem 2.1.15, to prove the equivalence of $\mathsf{JL}^\star$ and JL it is enough to show that every axiom schema of $\mathsf{JL}^\star$ is provable in JL and vice versa;

the identity operation translation hence being the witness for both embeddings.

Since $\mathsf{JL}^\star$ is an extension of JL, every axiom schema of JL is obviously provable in $\mathsf{JL}^\star$.

The propositional axiom schemas (5.1) and (5.2) of $\mathsf{JL}^\star$ are tautological and, hence, provable in JL. Axiom schema (5.3) of $\mathsf{JL}^\star$ is provable in JL by taking the contraposition of jt. Axiom schema (5.4) of $\mathsf{JL}^\star$ follows from jt by MP and the tautological formula schema $(t{:}A \to A) \to (A \to B) \to (t : A \to B)$. $\square$

It is an open question whether Theorem 5.1.1 also holds for logics without the additional axiom schemas (5.1)-(5.4).

5.2 Partial Conservativity for Logics of Belief

We now turn to logis of belief, i.e., logics that do not contain axiom schemas jt and jd. We introduce a similar notation as in Section 5.1. For $S \subseteq \{!, ?, \bar{?}\}$ by $B(S)$ (the B stands for *belief*) we denote the smallest extension of J that contains the defining axiom schemas for every operation in S. For example, $B(?, \bar{?})$ denotes $\mathsf{JB5}$ (again, we write $B(?, \bar{?})$ instead of $B(\{?, \bar{?}\})$, and so on).

Definition 5.2.1 ($A^{\ddagger S}$ and $\mathsf{CS}^{\ddagger S}$)**.** Let $S \subseteq \{!, ?, \bar{?}\}$. For any formula A, by $A^{\ddagger S}$ we denote the formula obtained from A by replacing all the terms that contain operations from S with variable x_1. Formally,

$$
\begin{aligned}
(P_i)^{\ddagger S} &\quad := \quad P_i \ ; \\
\bot^{\ddagger S} &\quad := \quad \bot \ ; \\
(A \to B)^{\ddagger S} &\quad := \quad A^{\ddagger S} \to B^{\ddagger S} \ ; \\
(t : A)^{\ddagger S} &\quad := \quad \begin{cases} x_1 : A^{\ddagger S} & \text{if } t \text{ contains an operation from } S, \\ t : A^{\ddagger S} & \text{otherwise} \ . \end{cases}
\end{aligned}
$$

For a Gentzen sequent $\Gamma \Rightarrow \Delta$, the sequent $\Gamma^{\ddagger S} \Rightarrow \Delta^{\ddagger S}$ is obtained from $\Gamma \Rightarrow \Delta$ by replacing every formula A in Γ or Δ with $A^{\ddagger S}$. Further, for a constant specification CS:

$$\mathsf{CS}^{\ddagger S} := \{A^{\ddagger S} \mid A \in \mathsf{CS}\} \ .$$

The following fact is obvious from Definition 5.2.1:

Fact 5.2.2 ($\ddagger S$ and Language)**.** *For $S_1 \subseteq S_2 \subseteq \{!, ?, \bar{?}\}$ and $S := S_2 \setminus S_1$, if A is an $\mathcal{L}(S_2)$-formula, then $A^{\ddagger S}$ is an $\mathcal{L}(S_1)$-formula. For example, if A is an $\mathcal{L}(!, ?)$-formula, then $A^{\ddagger\{!\}}$ is an $\mathcal{L}(?)$-formula.*

Lemma 5.2.3 proves a partial conservativity result for the following logics:

$$\mathsf{J} \subset \mathsf{JB}, \quad \mathsf{J} \subset \mathsf{J4}, \quad \mathsf{J} \subset \mathsf{J5}, \quad \mathsf{J} \subset \mathsf{J45}, \quad \mathsf{J4} \subset \mathsf{J45}, \quad \mathsf{J5} \subset \mathsf{J45} \ .$$

Lemma 5.2.3 (Partial Conservativity for Logics of Belief)**.** *Let S_2 be either $\{\bar{?}\}$ or a nonempty subset of $\{!, ?\}$ and let $S_1 \subsetneq S_2$. Let CS be a constant specification for $B(S_2)$ such that*

$$c^n_{i_n} : \ldots : c^1_{i_1} : C \notin \mathsf{CS} \ ,$$

for any instance C of a defining axiom schema of an operation from $S := S_2 \setminus S_1$ and for arbitrary constants $c^1_{i_1}, \ldots, c^n_{i_n}$ [1]*. Then $\mathsf{CS}^{\ddagger S}$ is a constant specification for $B(S_1)$, and for any $\mathcal{L}(S_1)$-formula A,*

$$B(S_2)_{\mathsf{CS}} \vdash A \quad \Longrightarrow \quad B(S_1)_{\mathsf{CS}^{\ddagger S}} \vdash A \ .$$

Proof. We first verify that $\mathsf{CS}^{\ddagger S}$ is a constant specification for $B(S_1)$. Since $\ddagger S$ does not affect constants, i.e., $(c^n_{i_n} : \ldots : c^1_{i_1} : C)^{\ddagger S} = c^n_{i_n} : \ldots : c^1_{i_1} : C^{\ddagger S}$, and since CS does not contain instances of defining axiom schemas of $S = S_2 \setminus S_1$, it is enough to show that for every instance C of a $B(S_1)$-axiom schema in language $\mathcal{L}(S_2)$, the formula $C^{\ddagger S}$ is an axiom of $B(S_1)$. From Fact 5.2.2 it follows that $C^{\ddagger S}$ is indeed an $\mathcal{L}(S_1)$-formula. It is easy to verify that if C is an instance of taut, app, or sum, then so is $C^{\ddagger S}$. For example,

$$(t : A \to (s + t) : A)^{\ddagger S} \;=\; t^{\ddagger S} : A^{\ddagger S} \to (t^{\ddagger S} + s^{\ddagger S}) : A^{\ddagger S}$$

is another instance (in language $\mathcal{L}(S_1)$) of sum.

It is also easy to verify that if C is an instance of a defining axiom schema of an operation from S_1, then so is $C^{\ddagger S}$. For example, assuming $? \in S_1$ (and, hence, $? \notin S$),

$$(t : A \to ?\, t : t : A)^{\ddagger S} \;=\; t^{\ddagger S} : A^{\ddagger S} \to ?\, t^{\ddagger S} : t^{\ddagger S} : A^{\ddagger S}$$

[1] Informally, CS is a constant specification for $B(S_1)$, but in language $\mathcal{L}(S_2)$. E.g., if $B(S_1)$ is J and $B(S_2)$ is $\mathsf{J5}$, then $c^n_{i_n} : \ldots : c^1_{i_1} : (\neg t : A \to \bar{?}\, t : \neg t : A) \notin \mathsf{CS}$.

is another instance (in language $\mathcal{L}(S_1)$) of $\mathsf{j5}$.

Therefore, $\mathsf{CS}^{\ddagger S}$ is a constant specification for $B(S_1)$.

Let A be an $\mathcal{L}(S_1)$-formula provable in $B(S_2)_{\mathsf{CS}}$. By Theorem 4.1.9, there exists a derivation $\mathcal{D}$ of $\Rightarrow A$ in the Gentzen system $\mathsf{G}B(S_2)_{\mathsf{CS}}$. We show by induction on the height of a sequent $\Gamma \Rightarrow \Delta$ in $\mathcal{D}$ that $\Gamma^{\ddagger S} \Rightarrow \Delta^{\ddagger S}$ is provable in the system $\mathsf{G}B(S_1)_{\mathsf{CS}^{\ddagger S}}$. It then follows from Theorem 4.1.9 that $A = A^{\ddagger S}$ is provable in $B(S_1)_{\mathsf{CS}^{\ddagger S}}$.

Base case. If $\Gamma \Rightarrow \Delta$ is an instance of $(\bot)$ or (Ax), then so is $\Gamma^{\ddagger S} \Rightarrow \Delta^{\ddagger S}$. If $\Gamma \Rightarrow \Delta$ is an instance of (CS), then it is of the form

$$\Gamma \Rightarrow \Delta', c^n_{i_n} : \ldots : c^1_{i_1} : C \ ,$$

where $c^n_{i_n} : \ldots : c^1_{i_1} : C \in \mathsf{CS}$. By definition of $\mathsf{CS}^{\ddagger S}$,

$$(c^n_{i_n} : \ldots : c^1_{i_1} : C)^{\ddagger S} = c^n_{i_n} : \ldots : c^1_{i_1} : C^{\ddagger S} \in \mathsf{CS}^{\ddagger S}$$

and, hence, $\Gamma^{\ddagger S} \Rightarrow \Delta'^{\ddagger S}, c^n_{i_n} : \ldots : c^1_{i_1} : C^{\ddagger S}$ is provable in $\mathsf{G}B(S_1)_{\mathsf{CS}^{\ddagger S}}$ by the rule (CS).

For the **inductive step** we look at the last rule R of a (sub-)derivation of $\Gamma \Rightarrow \Delta$.

By Lemma 4.1.2, if the derivation $\mathcal{D}$ makes use of a rule $(*)$, where $* \in \{!, ?, \bar{?}\}$, then the endsequent $\Rightarrow A$ must contain $*$. Therefore, for $* \in S$, R cannot be the rule $(*)$ because otherwise $*$ would occur in $\Rightarrow A$, which is impossible since A is an $\mathcal{L}(S_1)$-formula.

We first consider the case where R is $(\cdot)$ (the cases $(\mathsf{L}{\rightarrow})$, $(\mathsf{R}{\rightarrow})$, (RC), and $(+)$ are similar):

$$(\cdot)\ \frac{\Gamma \Rightarrow \Delta', s : (B \rightarrow C) \qquad \Gamma \Rightarrow \Delta', t : B}{\Gamma \Rightarrow \Delta', (s \cdot t) : C}$$

By induction hypothesis,

$$\begin{aligned} &\vdash \Gamma^{\ddagger S} \Rightarrow \Delta'^{\ddagger S}, \big(s : (B \rightarrow C)\big)^{\ddagger S} \qquad \text{and} \\ &\vdash \Gamma^{\ddagger S} \Rightarrow \Delta'^{\ddagger S}, (t : B)^{\ddagger S} \ , \end{aligned}$$

which can be rewritten as

$$\begin{aligned} &\vdash \Gamma^{\ddagger S} \Rightarrow \Delta'^{\ddagger S}, s^{\ddagger S} : (B^{\ddagger S} \rightarrow C^{\ddagger S}) \qquad \text{and} \\ &\vdash \Gamma^{\ddagger S} \Rightarrow \Delta'^{\ddagger S}, t^{\ddagger S} : B^{\ddagger S} \ . \end{aligned}$$

By an application of $(\cdot)$ we obtain a derivation of

$$\Gamma^{\ddagger S} \Rightarrow \Delta'^{\ddagger S}, (s^{\ddagger S} \cdot t^{\ddagger S}) : C^{\ddagger S} \quad = \quad \Gamma^{\ddagger S} \Rightarrow \Delta'^{\ddagger S}, \big((s \cdot t) : C\big)^{\ddagger S} \ .$$

The cases where R is a rule $(*)$ for $* \in \{!, ?, \bar{?}\}$ and $* \notin S$ are also straightforward. Exemplarily, we show the case where R is $(?)$ (and, hence, $? \notin S$):

$$(?)\,\frac{t : B, \Gamma \Rightarrow \Delta'}{\Gamma \Rightarrow \Delta', ?\,t : \neg t : B}$$

By induction hypothesis, $t^{\ddagger S} : B^{\ddagger S}, \Gamma^{\ddagger S} \Rightarrow \Delta'^{\ddagger S}$ is derivable. By an application of $(?)$, so is

$$\Gamma^{\ddagger S} \Rightarrow \Delta'^{\ddagger S}, ?\,t^{\ddagger S} : \neg t^{\ddagger S} : B^{\ddagger S} \ ,$$

which, since $?\,t^{\ddagger S} = (?\,t)^{\ddagger S}$, is the same as

$$\Gamma^{\ddagger S} \Rightarrow \Delta'^{\ddagger S}, (?\,t : \neg t : B)^{\ddagger S} \ . \qquad \square$$

5.3 Negative Results for Logics of Consistent Belief

Surprisingly, when it comes to logics of consistent belief, i.e., logics that contain axiom schema jd but lack jt, extensions of logics are *not* conservative. For example, $\mathsf{JD4}$ is not a conservative extension of JD, as was shown by Milnikel in [Mil12]. In this section, we expand this result and show that also $\mathsf{JD5}$ is not conservative over JD, JDB is not conservative over JD, and so on. It remains an open question whether $\mathsf{JDB45}$ is a conservative extension of $\mathsf{JDB4}$ and of $\mathsf{JDB5}$, but we give an informal argument for a negative answer at the end of the section.

Lemma 5.3.1 (Non-Conservativity)**.**
Let $\mathsf{JL}_1 \in \{\mathsf{JD}, \mathsf{JD4}, \mathsf{JD5}, \mathsf{JDB}, \mathsf{JD45}\}$ *and let* $\mathsf{JL}_2 \supsetneq \mathsf{JL}_1$*. Then* JL_2 *is not a conservative extension of* JL_1*.*

Proof. We first consider the case where JL_2 is either an extension of JT or one of the logics $\mathsf{JDB4}, \mathsf{JDB5}, \mathsf{JDB45}$. By Lemma 2.2.6, JL_2 proves jt. The following argument shows that the $\mathcal{L}(\varnothing)$-schema jt is not provable in JL_1 and, hence, JL_2 is not a conservative extension of JL_1: if jt were provable in JL_1, then, by Lemma 1.4.3, every formula

$\Box A \to A$ were provable in the modal logic that JL_1 is a justification counterpart of, i.e., in D, $\mathsf{D4}$, $\mathsf{D5}$, DB, or $\mathsf{D45}$. But it is well-known that none of these logics proves $\Box A \to A$.

We now prove the remaining cases.

The following is a proof[2] in both $\mathsf{JD4}$ and $\mathsf{JD45}$:

1. $x_1 : P_1$ (assumption)
2. $x_2 : (x_1 : P_1 \to \bot)$ (assumption)
3. $!\,x_1 : x_1 : P_1$ (by 1. and $\mathsf{j4}$)
4. $x_2 : (x_1 : P_1 \to \bot) \to !\,x : x : P \to x_2 \cdot !\,x_1 : \bot$ (app)
5. $x_2 \cdot !\,x_1 : \bot$ (by 2., 3., 4., and MP)
6. $\bot$ (by 5. and jd)

Hence, by the Deduction Theorem (Lemma 1.5.3),

$$\mathsf{JD4}/\mathsf{JD45} \vdash x_1 : P_1 \to x_2 : (x_1 : P_1 \to \bot) \to \bot \ .$$

Note that this is a formula in the language $\mathcal{L}(\varnothing)$. We show that it is not provable in $\mathsf{JD5}$ (and thus not in $\mathsf{JD} \subset \mathsf{JD5}$) and, hence, $\mathsf{JD4}$ and $\mathsf{JD45}$ are not conservative extensions of JD, and $\mathsf{JD45}$ is not a conservative extension of $\mathsf{JD5}$. Assume to the contrary that the formula is provable in $\mathsf{JD5}$. Then, by Lemma 1.4.3, its forgetul projection

$$\Box P_1 \to \Box(\Box P_1 \to \bot) \to \bot \ = \ \Diamond\neg P_1 \vee \Diamond\Box P_1 \tag{5.5}$$

is provable in $\mathsf{D5}$. We build a $\mathsf{D5}$-countermodel to show that this is a contradiction. Consider the following (serial and euclidean) Kripke frame:

$$\begin{tikzcd} w_1 \arrow[r] & w_2 \arrow[loop above] \arrow[r, bend left] & w_3 \arrow[loop above] \arrow[l, bend left] \end{tikzcd}$$

Now build a Kripke model based on the above frame with P_1 being true at w_2 and false at w_1 and w_3. Then $\Diamond\neg P_1$ and $\Diamond\Box P_1$ are false at w_1 and, hence, we have a $\mathsf{D5}$-countermodel for (5.5).

Now consider the following proof in $\mathsf{JD5}/\mathsf{JD45}$:

1. $\neg x_1 : P_1$ (assumption)
2. $x_2 : (\neg x_1 : P_1 \to \bot)$ (assumption)

[2] It is based on an informal example from [Mil12].

3. $?\,x_1 : \neg x_1 : P_1$ (by 1. and $\mathsf{j5}$)
4. $x_2 : (\neg x_1 : P_1 \to \bot) \to (?\,x_1 : \neg x_1 : P_1 \to x_2 \cdot ?\,x_1 : \bot)$ (app)
5. $x_2 \cdot ?\,x_1 : \bot$ (by 2., 3., 4., and MP)
6. $\bot$ (by 5. and jd)

Hence, by the Deduction Theorem (Lemma 1.5.3),

$$\mathsf{JD5}/\mathsf{JD45} \vdash \neg x_1 : P_1 \to x_2 : (\neg x_1 : P_1 \to \bot) \to \bot \ .$$

We show that this $\mathcal{L}(\varnothing)$-formula is not provable in $\mathsf{JD4}$ (and thus not in $\mathsf{JD} \subset \mathsf{JD4}$) and, hence, $\mathsf{JD45}$ is not a conservative extension of $\mathsf{JD4}$, and $\mathsf{JD5}$ is not a conservative extension of JD. Assume to the contrary that the formula is provable in $\mathsf{JD4}$. Then, by Lemma 1.4.3, its forgetul projection

$$\neg\Box P_1 \to \Box(\neg\Box P_1 \to \bot) \to \bot \ = \ \Box P_1 \vee \Diamond\Diamond\neg P_1 \tag{5.6}$$

is provable in $\mathsf{D4}$. We build a $\mathsf{D4}$-countermodel to show that this is a contradiction. Consider the following (serial and transitive) Kripke frame:

$$\begin{tikzcd} w_1 \arrow[r] \arrow[dr] & w_2 \arrow[d] \\ & w_3 \arrow[loop right] \end{tikzcd}$$

Now build a Kripke model based on this frame with P_1 being false at w_2 and true at w_3. Then $\Box P_1$ and $\Diamond\Diamond\neg P_1$ are false at w_1 and, hence, we have a $\mathsf{D4}$-countermodel for (5.6).

It remains to show that JDB is not a conservative extension of JD. Consider the following proof in JDB:

1. P_1 (assumption)
2. $x_2 : (\neg x_1 : \neg P_1 \to \bot)$ (assumption)
3. $\bar{?}\,x_1 : \neg x_1 : \neg P_1$ (by 1. and jb)
4. $x_2 : (\neg x_1 : \neg P_1 \to \bot) \to (\bar{?}\,x_1 : \neg x_1 : \neg P_1 \to x_2 \cdot \bar{?}\,x_1 : \bot)$ (app)
5. $x_2 \cdot \bar{?}\,x_1 : \bot$ (by 2., 3., 4., and MP)
6. $\bot$ (by 5. and jd)

Hence, by the Deduction Theorem (Lemma 1.5.3),

$$\mathsf{JD5} \vdash P_1 \to x_2 : (\neg x_1 : \neg P_1 \to \bot) \to \bot \ .$$

We show that this $\mathcal{L}(\varnothing)$-formula is not provable in JD and, hence, JDB is not a conservative extension of JD. Assume to the contrary that the formula is provable in JD. Then, by Lemma 1.4.3, its forgetul projection

$$P_1 \to \Box(\neg\Box\neg P_1 \to \bot) \to \bot \;=\; \neg P_1 \vee \Diamond\Diamond P_1 \tag{5.7}$$

is provable in D. We build a D-countermodel to show that this is a contradiction. Consider the following (serial) Kripke frame:

$$\begin{tikzcd} w_1 \arrow[r] & w_2 \arrow[loop above] \end{tikzcd}$$

Now build a Kripke model based on the above frame with P_1 being true at w_1 but false at w_2. Then $\neg P_1$ and $\Diamond\Diamond P_1$ are false at w_1 and, hence, we have a D-countermodel for (5.7). $\Box$

Non-conservativity of $\mathsf{JDB45}$ over $\mathsf{JDB4}$ and $\mathsf{JDB5}$ cannot be proved in the same way as Lemma 5.3.1 because, by Theorem 3.3.14, these three logics all have the same forgetul projection, i.e., $\mathsf{S5}$. We leave it open whether $\mathsf{JDB45}$ is conservative over $\mathsf{JDB4}$ and $\mathsf{JDB5}$. But the conjecture is that it is not. We conclude this section by giving an argument for this conjecture.

We internalize, in $\mathsf{JDB45}$, the second counterexample from the proof of Lemma 5.3.1 and argue that the so obtained formula is not provable in $\mathsf{JDB4}$. A similar argument (internalize the first counterexample from the proof of Lemma 5.3.1 instead of the second one) supports the conjecture that $\mathsf{JDB45}$ is not a conservative extension of $\mathsf{JDB5}$.

By rule (CS), applied to instances of $\mathsf{j5}$, app, and jd respectively,

$$\vdash c_1^1 : (\underbrace{\neg x_1 : P_1}_{A} \to \underbrace{?\,x_1 : \neg x_1 : P_1}_{B}) \;, \tag{5.8}$$

$$\vdash c_2^1 : \big(\underbrace{x_2 : (\neg x_1 : P_1 \to \bot)}_{C} \to (\underbrace{?\,x_1 : \neg x_1 : P_1}_{B} \to \underbrace{x_2 \cdot ?\,x_1 : \bot}_{D})\big) \;, \tag{5.9}$$

$$\vdash c_3^1 : (\underbrace{x_2 \cdot ?\,x_1 : \bot}_{D} \to \bot) \;. \tag{5.10}$$

By a propositional tautology and Internalization Property 1.5.4,

$$\vdash p : \Big((A \to B) \;\;\to\;\; \big(C \to (B \to D)\big) \;\;\to\;\; (D \to \bot) \;\;\to\;\; A \to C \to \bot\Big)$$

for some ground term p. It is easy to see from the proof of the Internalization Property 1.5.4 that p can be an $\mathcal{L}(\varnothing)$-term. By the above, an instance of app, (5.8), and MP,

$$\vdash p \cdot c_1^1 : \Big((C \to (B \to D)) \;\; \to \;\; (D \to \bot) \;\; \to \;\; A \to C \to \bot \Big) \ .$$

Similarly, by the above and (5.9),

$$\vdash (p \cdot c_1^1) \cdot c_2^1 : \big((D \to \bot) \;\; \to \;\; A \to C \to \bot \big) \ .$$

Finally, by the above and (5.10),

$$\vdash ((p \cdot c_1^1) \cdot c_2^1) \cdot c_3^1 : \big(A \to C \to \bot \big) \ ,$$

which is the same as

$$\vdash ((p \cdot c_1^1) \cdot c_2^1) \cdot c_3^1 : \big(\neg x_1 : P_1 \;\; \to \;\; x_2 : (\neg x_1 : P_1 \to \bot) \;\; \to \;\; \bot \big) \ . \tag{5.11}$$

Note that we have derived an $\mathcal{L}(\varnothing)$-formula (which is also an $\mathcal{L}(!, \bar{?})$-formula) in $\mathsf{JDB45}$. However, the described proof is clearly not a $\mathsf{JDB4}$-proof because, in $\mathsf{JDB4}$, $\mathsf{j5}$ is not an axiom schema and, hence, (5.8) is not provable in $\mathsf{JDB4}$. Note, however, that negative introspection is present in $\mathsf{JDB4}$ by Lemma 2.2.6. As a consequence, when translating the above proof into a $\mathsf{JDB4}$-proof, we are forced to use

$$\vdash q : (\neg x_1 : P_1 \to \mathrm{t}_?(x_1) : \neg x_1 : P_1)$$

instead of (5.8), where q is some ground term more complex than c_1^1 and $\mathrm{t}_?(x_1)$ is the term constructed in Lemma 2.2.6. Therefore, instead of (5.11), we would obtain (5.11) with q substituted for c_1^1. It seems that negative introspection being axiomatic is essential for proving (5.11) and, therefore, (5.11) is not provable in $\mathsf{JDB4}$. But a formal proof of this is left to others.

Conclusions

We have studied several aspects of all the 24 justification logics formed by the axiom schemas jd, jt, jb, j4, and j5. We have therefore provided a survey of the justification counterparts of all the modal logics in the modal cube.

We have classified these 24 logics by introducing an equivalence relation on justification logics that is based on translations of justification operations. We have proved that the justification counterparts of a modal logic are all pairwise equivalent, and have thereby shown that the equivalence relation is natural in that justification logics are equivalent if and only if they realize the same modal logic.

We have developed 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 in the modal cube. We have thereby reproved in a uniform way several known realization theorems and have realized modal logics that did not have justification counterparts prior to the publication of [BGK10]. Using the notion of equivalence of justification logics, we have demonstrated that the realization theorem for these 15 modal logics can be made modular, independent of whether the modal sequent systems are modular (i.e., whether any combination of modal rules corresponding to axiom schemas d, t, b, 4, and 5 yields a complete system). Our realization theorem is modular in the sense that we produce a justification counterpart for each axiomatization of a modal logic.

Our method is easily applicable to the modular nested sequent systems from [BS09]. Although these systems have turned out to be incomplete, our method should also be applicable to the corrected versions that Brünnler and Straßburger are working on.

We have introduced cut-free Gentzen systems for justification logics of belief, most of them did not have such systems before. The Gentzen systems for J and J4 enabled us to show that the Internalization Property for these logics has a converse direction: not only can we, for every formula A, build a term p such that $p : A$ is provable if A is, but for any term t, if $t : A$ is provable, then so is A. We have also shown that this property, which we called inversed internalization, does not hold for all logics of belief.

In the last chapter, we have studied which justification logics are conservative extensions of others. We have restated some known results but have also provided new ones.

The main open problem of this thesis is related to conservativity: it is not known whether the partial result on conservativity for logics of belief can be generalized to obtain a full result. That is, whether J4 is a conservative extension of J, J5 of J, and so on. The problem here is that the $(\cdot)$-rule of our Gentzen systems does not satisfy the subformula property and, hence, operations !, ?, and $\overline{?}$ can possibly be eliminated in a derivation. There exists also a semantic approach [Mil12] for proving this partial result. However, it is not much more promising because the semantic closure condition[3] for the operation $\cdot$ violates what could be called a semantic analog of the subformula property.

Apart from conservativity for logics of belief, there are some smaller open problems.

It remains unknown 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 [Fit09]. The main difficulty of obtaining this extra condition via our realization method lies in the contraction rule.

Since we have introduced new justification logics, i.e., the extensions of JB, an obvious next step is to look for appropriate semantics 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.

---

[3] Semantics for justification logics involve a so-called evidence function $\mathcal{E}$ from terms to formulas. The closure condition for $\cdot$ states that if $A \to B \in \mathcal{E}(t)$ and $A \in \mathcal{E}(s)$, then $B \in \mathcal{E}(t \cdot s)$.

It is not known whether cut-free Gentzen systems exist for logics of consistent belief, i.e., for JD, JD4, JD5, etc. It is also unknown whether the property of inversed internalization holds for JD, JDB, and JD4.

A long-term goal 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. [BKS11]).

Bibliography

[AK06] Sergei N. Artemov and Roman Kuznets. Logical omniscience via proof complexity. In Zoltán Ésik, editor, *Computer Science Logic, 20th International Workshop, CSL 2006, 15th Annual Conference of the EACSL, Szeged, Hungary, September 25–29, 2006, Proceedings*, volume 4207 of *Lecture Notes in Computer Science*, pages 135–149. Springer, 2006.

[AK09] Sergei N. Artemov and Roman Kuznets. Logical omniscience as a computational complexity problem. In Aviad Heifetz, editor, *Theoretical Aspects of Rationality and Knowledge, Proceedings of the Twelfth Conference (TARK 2009)*, pages 14–23, Stanford University, California, July 6–8, 2009. ACM.

[AKS99] Sergei N. Artemov, Yevgeny Kazakov, and Dmitry Shapiro. Logic of knowledge with justifications. Technical Report CFIS 99–12, Cornell University, 1999.

[Art95] Sergei N. Artemov. Operational modal logic. Technical Report MSI 95–29, Cornell University, December 1995.

[Art01] Sergei N. Artemov. Explicit provability and constructive semantics. *Bulletin of Symbolic Logic*, 7(1):1–36, March 2001.

[Art02] Sergei N. Artemov. Unified semantics for modality and λ-terms via proof polynomials. In Kees Vermeulen and Ann Copestake, editors, *Algebras, Diagrams and Decisions in Language, Logic and Computation*, volume 144 of *CSLI Lecture Notes*, pages 89–118. CSLI Publications, Stanford, 2002.

[Art08] Sergei N. Artemov. The logic of justification. *The Review of Symbolic Logic*, 1(4):477–513, December 2008.

[BGK10] Kai Brünnler, Remo Goetschi, and Roman Kuznets. A syntactic realization theorem for justification logics. In Lev Beklemishev, Valentin Goranko, and Valentin Shehtman, editors, *Proceedings of Advances in Modal Logic 2010*, volume 8. College Publications, 2010.

[BK12] Samuel R. Buss and Roman Kuznets. Lower complexity bounds in justification logic. *Annals of Pure and Applied Logic*, 163(7):888–905, July 2012. Published online November 2011.

[BKS11] Samuel Bucheli, Roman Kuznets, and Thomas Studer. Justifications for common knowledge. *Journal of Applied Non-classical Logics*, 21:35–60, 2011.

[BN98] Franz Baader and Tobias Nipkow. *Term* Re*writing and* All That. Cambridge University Press, 1998.

[Bre00] Vladimir N. Brezhnev. On explicit counterparts of modal logics. Technical Report CFIS 2000–05, Cornell University, 2000.

[Brü03] Kai Brünnler. *Deep Inference and Symmetry in Classical Proofs*. PhD thesis, Technische Universität Dresden, September 2003.

[Brü09] Kai Brünnler. Deep sequent systems for modal logic. *Archive for Mathematical Logic*, 48(6):551–577, July 2009.

[BS09] Kai Brünnler and Lutz Straßburger. Modular sequent systems for modal logic. In Martin Giese and Arild Waaler, editors, *Proceedings of TABLEAUX 2009*, volume 5607 of *Lecture Notes in Artificial Intelligence*, pages 152–166. Springer, 2009.

[Che80] Brian F. Chellas. *Modal Logic: An Introduction*. Cambridge University Press, 1980.

[Fit05] Melvin Fitting. The logic of proofs, semantically. *Annals of Pure and Applied Logic*, 132(1):1–25, February 2005.

[Fit07] Melvin Fitting. Justification logics and conservative extensions. Technical Report TR–2007015, CUNY Ph.D. Program in Computer Science, July 2007.

[Fit08] Melvin Fitting. Justification logics, logics of knowledge, and conservativity. *Annals of Mathematics and Artificial Intelligence*, 53(1–4):153–167, August 2008.

[Fit09] Melvin Fitting. Realizations and LP. *Annals of Pure and Applied Logic*, 161(3):368–387, December 2009.

[Fit11] Melvin Fitting. The realization theorem for $\mathsf{S5}$: A simple, constructive proof. In Johan van Benthem, Amitabha Gupta, and Eric Pacuit, editors, *Games, Norms and Reasons: Logic at the Crossroads*, volume 353 of *Synthese Library*, chapter 4, pages 61–76. Springer, 2011.

[Fit12] Melvin Fitting. Prefixed tableaus and nested sequents. *Annals of Pure and Applied Logic*, 163(3):291–313, March 2012. Published online October 2011.

[Gar09] James Garson. Modal logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Winter 2009 edition, 2009. `http://plato.stanford.edu/archives/win2009/entries/logic-modal/`.

[Get63] Edmund L. Gettier. Is justified true belief knowledge? *Analysis*, 23(6):121–123, June 1963.

[GK12] Remo Goetschi and Roman Kuznets. Realization for justification logics via nested sequents: Modularity through embedding. *Annals of Pure and Applied Logic*, 2012.

[Gug07] Alessio Guglielmi. A system of interaction and structure. *ACM Transactions on Computational Logic*, 8(1:1), January 2007.

[HC96] George E. Hughes and Max J. Cresswell. *A New Introduction to Modal Logic*. Routledge, 1996.

[Kur09] Hidenori Kurokawa. Tableaux and hypersequents for Justification Logic. In Sergei [N.] Artemov and Anil Nerode, editors, *Logical Foundations of Computer Science, International Symposium, LFCS 2009, Deerfield Beach, FL, USA, January 3–6, 2009, Proceedings*, volume 5407 of *Lecture Notes in Computer Science*, pages 295–308. Springer, 2009.

[Kur11] Hidenori Kurokawa. Tableaux and hypersequents for justification logics. *Annals of Pure and Applied Logic*, Articles In Press, October 2011.

[Kuz05] Roman Kuznets. On decidability of the logic of proofs with arbitrary constant specifications. In *2004 Annual Meeting of the Association for Symbolic Logic, Carnegie Mellon University, Pittsburgh, PA, May 19–23, 2004*, volume 11(1) of *Bulletin of Symbolic Logic*, page 111. Association for Symbolic Logic, March 2005. Abstract.

[Kuz08] Roman Kuznets. *Complexity Issues in Justification Logic*. PhD thesis, CUNY Graduate Center, May 2008.

[Kuz10] Roman Kuznets. Self-referential justifications in epistemic logic. *Theory of Computing Systems*, 46(4):636–661, May 2010.

[Mil07] Robert S. Milnikel. Derivability in certain subsystems of the Logic of Proofs is Π_2^p-complete. *Annals of Pure and Applied Logic*, 145(3):223–239, March 2007.

[Mil09] Robert S. Milnikel. Conservativity for logics of justified belief. In Sergei N. Artemov and Anil Nerode, editors, *Logical Foundations of Computer Science, International Symposium, LFCS 2009, Deerfield Beach, FL, USA, January 3–6, 2009, Proceedings*, volume 5407 of *Lecture Notes in Computer Science*, pages 354–364. Springer, 2009.

[Mil12] Robert S. Milnikel. Conservativity for logics of justified belief: Two approaches. Extended version of [Mil09], 2012.

[Pac05] Eric Pacuit. A note on some explicit modal logics. In *Proceedings of the 5th Panhellenic Logic Symposium*, pages 117–125, Athens, Greece, July 25–28, 2005. University of Athens.

[Pog10] Francesca Poggiolesi. Towards a satisfying proof analysis of the logic of proofs. In Xabier Arrazola and María Ponte, editors, *LogKCA-10, Proceedings of the Second ILCLI International Workshop on Logic and Philosophy of Knowledge, Communication and Action*, pages 371–385. University of the Basque Country Press, 2010.

[Ren06] Bryan Renne. Semantic cut-elimination for two explicit modal logics. In Janneke Huitink and Sophia Katrenko, editors, *Proceedings of the Eleventh ESSLLI Student Session, 18th European Summer School in Logic, Language and Information (ESSLLI'06)*, pages 148–158, Málaga, Spain, July 31–August 11, 2006. FoLLI.

[Rub06] Natalia M. Rubtsova. Evidence reconstruction of epistemic modal logic $\mathsf{S5}$. In Dima Grigoriev, John Harrison, and Edward A. Hirsch, editors, *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*, pages 313–321. Springer, 2006.

[Stu11] Thomas Studer. An application of justification logic to protocol verification. In *Proceedings of Computational Intelligence and Security CIS 2011*, page 779–783. IEEE, IEEE, 2011. URL: `http://www.iam.unibe.ch/ltgpub/2011/stu11b.pdf`.

[TS96] Anne Sjerp Troelstra and Helmut Schwichtenberg. *Basic Proof Theory*. Cambridge University Press, 1996.

Index