

Chapman University

Chapman University Digital Commons

Mathematics, Physics, and Computer Science
Faculty Articles and Research

Science and Technology Faculty Articles and
Research

10-22-2021

Algorithmic Correspondence for Relevance Logics, Bunched Implication Logics, and Relation Algebras via an Implementation of the Algorithm PEARL

Willem Conradie

Valentin Goranko

Peter Jipsen

Follow this and additional works at: https://digitalcommons.chapman.edu/scs_articles



Part of the [Algebra Commons](#), and the [Other Mathematics Commons](#)

Algorithmic Correspondence for Relevance Logics, Bunched Implication Logics, and Relation Algebras via an Implementation of the Algorithm PEARL

Comments

This is a pre-copy-editing, author-produced PDF of an article accepted for publication in Relational and Algebraic Methods in Computer Science. RAMiCS 2021. Lecture Notes in Computer Science, volume 13027, in 2021. The final publication may differ and is available at Springer via https://doi.org/10.1007/978-3-030-88701-8_8.

Copyright

Springer

Algorithmic correspondence for relevance logics, bunched implication logics, and relation algebras via an implementation of the algorithm PEARL

Willem Conradie¹, Valentin Goranko^{2,1}, and Peter Jipsen³

¹ School of Mathematics, University of the Witwatersrand, South Africa
willem.conradie@wits.ac.za

² Department of Philosophy, Stockholm University, Sweden
valentin.goranko@philosophy.su.se

³ Department of Mathematics, Chapman University, USA jipsen@chapman.edu

Abstract. The non-deterministic algorithmic procedure PEARL (acronym for ‘Propositional variables Elimination Algorithm for Relevance Logic’) has been recently developed for computing first-order equivalents of formulas of the language of relevance logics \mathcal{L}_R in terms of the standard Routley-Meyer relational semantics. It succeeds on a large class of axioms of relevance logics, including all so called inductive formulas. In the present work we re-interpret PEARL from an algebraic perspective, with its rewrite rules seen as manipulating quasi-inequalities interpreted over Urquhart’s relevant algebras, and report on its recent Python implementation. We also show that all formulae on which PEARL succeeds are canonical, i.e., preserved under canonical extensions of relevant algebras. This generalizes the “canonicity via correspondence” result in [37]. We also indicate that with minor modifications PEARL can also be applied to bunched implication algebras and relation algebras.

1 Introduction

This work relates two important areas of development in non-classical logics, viz. *relevance logics* and *algorithmic correspondence theory*, by applying the latter to the possible worlds semantics for relevance logic based on Routley-Meyer frames [31], by means of an implementation of the recently developed in [7] algorithm PEARL. That semantics is, in turn, duality-theoretically related to the algebraic semantics for relevance logic based on Urquhart’s relevant algebras [37]. Routley-Meyer frames also capture the semantics of (positive) relation algebras [24], [14], and of bunched implication algebras [30], hence the algorithm PEARL implemented here is also applicable to arrow logic [3], [15] and bunched implication logics [30].

Modal Correspondence theory. The Sahlqvist-van Benthem theorem [32], [2], proved in the mid 1970s, is a fundamental result in the model theory of modal logic. It gives a syntactic characterization of a class of modal formulas which define first-order conditions on Kripke frames and which are canonical, hence, when added to the basic normal modal logic K, they axiomatize logics which are

strongly complete with respect to elementary frame classes. The Sahlqvist-van Benthem theorem sets the stage for the emergence and development of the so called *correspondence theory in modal logic*, cf. [4]. The literature on the topic contains many analogues of the Sahlqvist-van Benthem theorem for a wide range of non-classical logics. Various illuminating alternative proofs have appeared, including Jonsson’s purely algebraic proof of the canonicity part [25], and the ‘canonicity-via-correspondence’ approach pioneered by Sambin and Vaccaro [33].

The Sahlqvist-van Benthem class of formulas has been significantly extended to the class of so called *inductive formulas* [21,22,23] which cover frame classes not definable by a Sahlqvist-van Benthem formula while enjoying the same properties of elementarity and canonicity. At about the same time, a new line of research known as *algorithmic correspondence theory* emerged. It involves the use of algorithms like SCAN and DLS to try and compute first-order frame correspondence for modal formulas by eliminating the second-order quantifiers from their standard second-order frame correspondents. In particular, the algorithm SQEMA [9] was developed for algorithmic correspondence in modal logic. It manipulates formulas in an extended hybrid language to eliminate propositional variables and thereby produces pure hybrid formulas which translate into first-order logic via the standard translation, and simultaneously proves their canonicity via an argument in the style of Sambin and Vaccaro. This approach was extended to logics algebraically captured by normal (distributive) lattice expansions [10,11] in a line of research known as *unified correspondence* [6].

Correspondence theory for Relevance logic. Much work has been done over the years on computing first-order equivalents and proving completeness of a range of specific axioms for relevance logics with respect to the *Routley-Meyer relational semantics* (cf. [31]). Routley-Meyer frames involve not a binary, but a ternary relation, with several conditions imposed on it, needed to ensure upward closedness of the valuations of all formulas. That makes the possible worlds semantics for relevance logic based on such frames technically more complex and proving correspondence results for it “by hand” can be significantly more elaborate than those for modal logics with their standard Kripke semantics, which calls for a systematic development of respective correspondence theory for relevance logics. Until recently, that problem remained little explored, with just a few works, incl. those of Seki [34] and Badia [1], defining some classes of Sahlqvist-van Benthem type formulas for relevance logics and proving correspondence results for them. Likewise, Suzuki [35,36], has established correspondence for the full Lambek calculus with respect to the so-called bi-approximation semantics, obtained via canonical extensions in the style of [16]. For closely related distributive substructural logics, such as bunched implication logics, an elegant categorical approach to canonicity and correspondence is based on duality theory and coalgebras [12]. A general algorithmic correspondence theory of relevance logics has recently been developed in [7], on which the presently reported work is based.

The algorithm PEARL and its implementation. A non-deterministic algorithmic procedure PEARL (acronym for **P**ropositional variables **E**limination **A**lgorithm for **R**elevance **L**ogic) for computing first-order equivalents in terms of frame validity

of formulas of the language \mathcal{L}_R for relevance logics is developed in [7]. PEARL is an adaptation of the above mentioned procedures SQEMA [9] (for normal modal logics) and ALBA [10,11] (for distributive and non-distributive modal logics). Furthermore, a large syntactically defined class of *inductive relevance formulas* in \mathcal{L}_R is defined in [7], based on specific order-theoretic properties of the algebraic interpretations of the connectives, following the general methodology of [11]. It is shown in [7] that PEARL succeeds for all such formulas and correctly computes for them equivalent with respect to frame validity first-order definable conditions on Routley-Meyer frames. This gives a general basis for comparing inductive and Sahlqvist formulas across different logics and for different relational semantics for the same logic. Thus, [11, Example 3.14] has shown that Suzuki’s Sahlqvist class is properly included in the respective class of inductive formulas. Likewise, for the case of \mathcal{L}_R , it is shown in [7] that the class of inductive formulas properly extends the classes of Sahlqvist formulas of Seki [34] and Badia [1].

In the present work we re-interpret the algorithm PEARL from an algebraic perspective with its rewrite rules seen as manipulating quasi-inequalities interpreted over Urquhart’s relevant algebras [37]. This enables us to complete the part of the Sahlqvist-van Benthem theorem still outstanding from the previous work, namely the fact that all inductive \mathcal{L}_R -formulas are canonical, i.e., are preserved under canonical extensions of relevant algebras. Via the discrete duality between perfect relevant algebras and Routley-Meyer frames, this establishes the fact that all inductive \mathcal{L}_R -formulas axiomatise logics which are complete with respect to first-order definable classes of Routley-Meyer frames. This generalizes the “canonicity via correspondence” result in [37] for (what we can now recognise as) a certain special subclass of Sahlqvist-van Benthem formulas in the “groupoid” sublanguage of \mathcal{L}_R where fusion is the only connective. We then present an optimised and deterministic version of PEARL, which we have recently implemented in Python and applied to verify the first-order equivalents of a number of important axioms for relevance logics known from the literature, as well as on several new types of formulas. In this paper we report on the implementation and on some testing results.

Relevance logics and relation algebras. Even though developed with different motivations, these two areas are technically closely related, as noted and explored in several papers besides [37], incl. [5], [28], [24], [26], [15]. We note that, by extending \mathcal{L}_R with a Heyting implication (which is a residual of the meet operation), removing relevant negation, and adding commutativity and associativity as axioms of fusion, our results can also be applied to bunched implication algebras. Alternatively one can extend \mathcal{L}_R with classical implication and apply the same algorithm to relation algebras. In this case the Routley-Meyer frames have the order of an antichain and are the same as atom structures of relation algebras. Further details are discussed at the end of Section 7.

Structure of the paper. In Section 2 we provide the necessary background on the syntax, algebraic and relational semantics of relevance logic, define relevant algebras and then extend their language by adding adjoints and residuals of the standard operators of relevance logic. Then, in Section 3 we establish du-

ality between perfect relevant algebras and complex algebras of Routley-Meyer frames. Section 4 presents the rules of the calculus on which PEARL is based, and Section 5 contains a concise description of the main phases of the algorithm itself. In Section 6 we give a brief description of the implementation of PEARL, and in Section 7 we state some results. We then conclude with Section 8. After the references we have included a short appendix containing some additional technicalities and some examples of the output of PEARL.

2 Preliminaries

In this section we provide background on the syntax and algebraic and relational semantics of relevance logic. For further details we refer the reader to [31], [17] and (for relevance logics) to [37] and [7].

2.1 Relevance logic and its algebraic semantics

The language of propositional relevance logic \mathcal{L}_R over a fixed set of propositional variables VAR is given by

$$A = p \mid \perp \mid \top \mid \mathbf{t} \mid \sim A \mid (A \wedge A) \mid (A \vee A) \mid (A \circ A) \mid (A \rightarrow A)$$

for $p \in \text{VAR}$. The relevant connectives \circ , \sim and \rightarrow are called **fusion**, **(relevant) negation** and **(relevant) implication**, respectively. The constant \mathbf{t} is referred to as **(relevant) truth**. We also add the constants \top and \perp for convenience. Equations and inequalities of \mathcal{L}_R -formulas can be algebraically interpreted in relevant algebras as defined by Urquhart in [37].

Definition 1 ([37]). *A structure $\mathbb{A} = \langle A, \wedge, \vee, \circ, \rightarrow, \sim, \mathbf{t}, \top, \perp \rangle$ is called a **relevant algebra** if it satisfies the following conditions:*

1. $\langle A, \wedge, \vee, \top, \perp \rangle$
is a bounded distributive lattice,
2. $a \circ (b \vee c) = (a \circ b) \vee (a \circ c)$,
3. $(b \vee c) \circ a = (b \circ a) \vee (c \circ a)$,
4. $\sim(a \vee b) = \sim a \wedge \sim b$,
5. $\sim(a \wedge b) = \sim a \vee \sim b$,
6. $\sim\top = \perp$ and $\sim\perp = \top$,
7. $a \circ \perp = \perp \circ a = \perp$,
8. $\mathbf{t} \circ a = a$, and
9. $a \circ b \leq c$ iff $a \leq b \rightarrow c$.

An \mathcal{L}_R -formula ϕ is **valid on a relevant algebra** \mathbb{A} if the inequality $\mathbf{t} \leq \phi$ (implicitly universally quantified over all propositional variables) is valid on \mathbb{A} and **valid on a class of relevant algebras** if it is valid on each member of that class. We also refer the reader to [37] for axiomatizations of the logic of the class of all relevant algebras.

2.2 Relational semantics

Relevance logic can be given relational semantics based on structures called ‘Routley-Meyer frames’, which we will now define. A **relevance frame** is a tuple $\mathcal{F} = \langle W, O, R, * \rangle$, where:

- W is a non-empty set of states (possible worlds);
- $O \subseteq W$ is the subset of **normal** states;

- $R \subseteq W^3$ is a **relevant accessibility relation**;
- $*$: $W \rightarrow W$ is a function, called the **Routley star**.

The binary relation \preceq is defined in every relevance frame by specifying that $u \preceq v$ iff $\exists o(o \in O \wedge Rouv)$. A **Routley-Meyer frame**⁴ (for short, **RM-frame**) is a relevance frame satisfying the following conditions for all $u, v, w, x, y, z \in W$:

1. $x \preceq x$
2. If $x \preceq y$ and $Ryuv$ then $Rxuv$.
3. If $x \preceq y$ and $Ruyv$ then $Ruxv$.
4. If $x \preceq y$ and $Ruvx$ then $Ruvy$.
5. If $x \preceq y$ then $y^* \preceq x^*$.
6. O is upward closed w.r.t. \preceq ,
i.e. if $o \in O$ and $o \preceq o'$ then $o' \in O$.

These properties ensure that \preceq is reflexive and transitive, hence a preorder, and that the semantics of the logical connectives has the upward monotonicity property stated below.

A **Routley-Meyer model (RM-model)** is a tuple $\mathcal{M} = \langle W, O, R, *, V \rangle$, where $\langle W, O, R, * \rangle$ is a Routley-Meyer frame and $V : \text{VAR} \rightarrow \mathcal{P}(W)$ is a mapping, called a **relevant valuation**, assigning to every atomic proposition $p \in \text{VAR}$ a set $V(p)$ of states which is *upward closed* w.r.t. \preceq .

Truth of a formula A in an RM-model $\mathcal{M} = \langle W, O, R, *, V \rangle$ at a state $u \in W$, denoted $\mathcal{M}, u \Vdash A$, is defined as follows:

- $\mathcal{M}, u \Vdash p$ iff $u \in V(p)$;
- $\mathcal{M}, u \Vdash \mathbf{t}$ iff $u \in O$;
- $\mathcal{M}, u \Vdash \sim A$ iff $\mathcal{M}, u^* \not\Vdash A$;
- $\mathcal{M}, u \Vdash A \wedge B$ iff $\mathcal{M}, u \Vdash A$ and $\mathcal{M}, u \Vdash B$;
- $\mathcal{M}, u \Vdash A \vee B$ iff $\mathcal{M}, u \Vdash A$ or $\mathcal{M}, u \Vdash B$;
- $\mathcal{M}, u \Vdash A \rightarrow B$ iff for every v, w , if $Ruvw$ and $\mathcal{M}, v \Vdash A$ then $\mathcal{M}, w \Vdash B$.
- $\mathcal{M}, u \Vdash A \circ B$ iff there exist v, w such that $Rvuw$, $\mathcal{M}, v \Vdash A$ and $\mathcal{M}, w \Vdash B$.

Thus, the Routley-Meyer semantics follows a standard pattern for relational semantics of modal operators. In particular, the fusion is a binary ‘diamond’, interpreted with a ternary relation, and negation is both a unary box and diamond, interpreted via a functional binary relation. One can show, by a routine structural induction on formulas, (cf. e.g. [31]) that this semantics satisfies **upward monotonicity**: for every RM-model \mathcal{M} and a formula A of \mathcal{L}_R , the set $\llbracket A \rrbracket_{\mathcal{M}} = \{u \mid \mathcal{M}, u \Vdash A\}$ is upward closed.

A formula A is declared **true in an RM-model** \mathcal{M} , denoted by $\mathcal{M} \Vdash A$, if $\mathcal{M}, o \Vdash A$ for every $o \in O$. It is **valid in an RM-frame** \mathcal{F} , denoted by $\mathcal{F} \Vdash A$, iff it is true in every RM-model over that frame, and A is **RM-valid**, denoted by $\Vdash A$, iff it is true in every RM-model.

⁴ The definition of Routley-Meyer frames takes the relation R and subset O as primary and defines the pre-order \preceq in terms of them. This does not restrict the pre-orders that can occur within Routley-Meyer frames. Indeed, given an upward closed subset $O \subseteq W$ and a pre-order \preceq on W one can define a respective ternary relation $R \subseteq W^3$ by specifying that, for all triples (x, y, z) , $Rxyz$ iff $x \preceq o$ for some $o \in O$ and $x \preceq y$.

All semantic notions of truth and validity defined above can be translated to FOL, resp. universal monadic second order, by means of a **standard translation**, analogous to the one applied to modal logic (cf. [4]). See the details in the full paper [8].

2.3 Perfect relevant algebras and the extended language \mathcal{L}_R^+

Given a Routley-Meyer frame $\mathcal{F} = \langle W, R, *, O \rangle$, its **complex algebra** is the structure

$$\mathcal{F}^+ = \langle \mathcal{P}^\uparrow(W), \cap, \cup, \rightarrow, \circ, \sim, O, W, \emptyset \rangle$$

where $\mathcal{P}^\uparrow(W)$ is the set of all upwards closed subsets (hereafter called **up-sets**) of W , \cap and \cup are set-theoretic intersection and union, and for all $Y, Z \in \mathcal{P}^\uparrow(W)$ the following hold:

$$Y \rightarrow Z = \{x \in W \mid \text{for all } y, z \in W, \text{ if } Rxyz \text{ and } y \in Y, \text{ then } z \in Z\},$$

$$Y \circ Z = \{x \in W \mid \text{there exist } y \in Y \text{ and } z \in Z \text{ such that } Ryzx\},$$

$$\sim Y = \{x \in W \mid x^* \notin Y\}.$$

It is easy to check that \mathcal{F}^+ is a relevant algebra.

An element a of a lattice \mathbb{L} is **completely join-irreducible** (resp., **completely join-prime**) if whenever $a = \bigvee S$ ($a \leq \bigvee S$) for some $S \subseteq L$, then $a = s$ ($a \leq s$) for some $s \in S$. The notions of **meet-irreducibility** and **primality** are defined order-dually. Complete join/meet primality implies complete join/meet irreducibility and for complete distributive lattices the notions coincide.

A relevant algebra $\mathbb{A} = \langle A, \wedge, \vee, \circ, \rightarrow, \sim, \mathbf{t}, \top, \perp \rangle$ is **perfect** if $\langle A, \wedge, \vee, \top, \perp \rangle$ is a complete, completely distributive lattice that is join-generated (resp., meet-generated) by the set of its completely join-irreducible elements $J^\infty(\mathbb{A})$ (resp., the set of its completely meet-irreducible elements $M^\infty(\mathbb{A})$), while $\bigvee S \circ a = \bigvee_{s \in S} (s \circ a)$, $a \circ \bigvee S = \bigvee_{s \in S} (a \circ s)$, $\bigvee S \rightarrow a = \bigwedge_{s \in S} (s \rightarrow a)$, $a \rightarrow \bigwedge S = \bigwedge_{s \in S} (a \rightarrow s)$, $\sim \bigvee S = \bigwedge_{s \in S} \sim s$ and $\sim \bigwedge S = \bigvee_{s \in S} \sim s$ for all $S \subseteq A$ and $a \in A$. Now, in fact, every \mathcal{F}^+ is a *perfect* relevant algebra. Further, every relevant algebra \mathbb{A} can be compactly and densely embedded in a unique perfect relevant algebra, namely in its *canonical extension* (cf. e.g. [16]) which we will denote \mathbb{A}^δ .

For any perfect distributive lattice \mathbb{A} , the map $\kappa : J^\infty(\mathbb{A}) \rightarrow M^\infty(\mathbb{A})$ defined by $j \mapsto \bigvee \{u \in \mathbb{A} \mid j \not\leq u\}$ is an order isomorphism (cf. [19, Sec. 2.3]) when considering $J^\infty(\mathbb{A})$ and $M^\infty(\mathbb{A})$ as subsets of \mathbb{A} . The inverse of κ is $\lambda : M^\infty(\mathbb{A}) \rightarrow J^\infty(\mathbb{A})$, given by the assignment $m \mapsto \bigwedge \{u \in \mathbb{A} \mid u \not\leq m\}$. From these definitions, we immediately have that, for every $u \in \mathbb{A}$, every $j \in J^\infty(\mathbb{A})$ and every $m \in M^\infty(\mathbb{A})$,

$$j \not\leq u \text{ iff } u \leq \kappa(j), \tag{1}$$

$$u \not\leq m \text{ iff } \lambda(m) \leq u. \tag{2}$$

Since in perfect relevant algebras each of $\sim, \vee, \wedge, \circ$ and \rightarrow preserves or reverses arbitrary meets and/or joins in each coordinate, they are residuated in each coordinate (see e.g. [18]). The algebra therefore supports the interpretation of an extended language with connectives for the residuals of these operations.

In particular, we extend the language \mathcal{L}_R to \mathcal{L}_R^+ by adding the **left adjoint** \sim^b and the **right adjoint** \sim^\sharp of \sim , the **intuitionistic (Heyting) implication** \Rightarrow (as right residual of \wedge), the **coimplication** \leftarrow as the left residual of \vee , and the operation \leftrightarrow as the residual of \circ in the second coordinate and of \rightarrow in the first coordinate. Thus, in any perfect relevant algebra \mathbb{A} we have that:

1. $\sim a \leq b$ iff $\sim^b b \leq a$
2. $a \leq \sim b$ iff $b \leq \sim^\sharp a$
3. $a \leq b \vee c$ iff $a \leftarrow b \leq c$
4. $a \wedge b \leq c$ iff $a \leq b \Rightarrow c$
5. $a \circ b \leq c$ iff $a \leq b \rightarrow c$
6. $a \circ b \leq c$ iff $b \leq a \leftrightarrow c$

We also include in \mathcal{L}_R^+ two countably infinite sets of special variables, $\text{NOM} = \{\mathbf{j}_0, \mathbf{j}_1, \mathbf{j}_2, \dots\}$ and $\text{CNOM} = \{\mathbf{m}_0, \mathbf{m}_1, \mathbf{m}_2, \dots\}$. These are respectively called **nominals** and **co-nominals** and will be interpreted as ranging respectively over completely join-irreducibles and completely meet-irreducibles. Informally, we will denote nominals by $\mathbf{i}, \mathbf{j}, \mathbf{k}$, possibly with indices, while co-nominals will be denoted by \mathbf{m}, \mathbf{n} , possibly with indices. To distinguish visually from \mathcal{L}_R , the formulas of the extended language \mathcal{L}_R^+ will be denoted by lowercase greek letters, typically $\alpha, \beta, \gamma, \phi, \psi, \xi$, etc. and are defined by the following grammar:

$$\begin{aligned} \phi = & p \mid \mathbf{i} \mid \mathbf{m} \mid \top \mid \perp \mid \mathbf{t} \mid \sim\phi \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid (\phi \circ \phi) \mid (\phi \rightarrow \phi) \mid \\ & \sim^b\phi \mid \sim^\sharp\phi \mid (\phi \leftarrow \phi) \mid (\phi \Rightarrow \phi) \mid (\phi \leftrightarrow \phi) \end{aligned}$$

where $p \in \text{VAR}$, $\mathbf{i} \in \text{NOM}$ and $\mathbf{m} \in \text{CNOM}$. We denote $\text{ATOMS} := \text{VAR} \cup \text{NOM} \cup \text{CNOM}$. The elements of ATOMS will be called **atoms**. An \mathcal{L}_R^+ -formula is called **pure** if it contains no propositional variables but only, possibly, nominals, co-nominals and constants. To each connective we assign a **polarity type**⁵ indicating whether each coordinate of its interpretation in (perfect) relevant algebras is order-preserving or order-reversing, as follows:

1. $\epsilon_{\sim} = \epsilon_{\sim^b} = \epsilon_{\sim^\sharp} = (-)$
2. $\epsilon_{\wedge} = \epsilon_{\vee} = \epsilon_{\circ} = (+, +)$
3. $\epsilon_{\rightarrow} = \epsilon_{\Rightarrow} = \epsilon_{\leftarrow} = (-, +)$
4. $\epsilon_{\leftarrow} = (+, -)$

We write $\epsilon_h(i)$ for the i -th coordinate of ϵ_h . We now define the notions of **positive** and **negative occurrences** of atoms in \mathcal{L}_R^+ -formulas recursively: an occurrence of an atom a is positive in a ; an occurrence of a which is positive (negative) in ϕ is positive (negative) in $h(\psi_1, \dots, \psi_{i-1}, \phi, \psi_{i+1}, \dots, \psi_n)$ if $\epsilon_h(i) = +$ and negative (positive) in $h(\psi_1, \dots, \psi_{i-1}, \phi, \psi_{i+1}, \dots, \psi_n)$ if $\epsilon_h(i) = -$. We then say that a **formula** $\phi \in \mathcal{L}_R^+$ is **positive (negative) in an atom** a iff all occurrences of a in ϕ are positive (negative). An **inequality** $\phi \leq \psi$ is **positive (negative) in an atom** a if ϕ is negative (positive) in a while ψ is positive (negative) in a .

3 Duality between perfect relevant algebras and complex algebras of Routley-Meyer frames

As already mentioned, the complex algebra \mathcal{F}^+ of any Routley-Meyer frame $\mathcal{F} = \langle W, R, *, O \rangle$ is a perfect relevant algebra. Moreover, $J^\infty(\mathcal{F}^+) = \{\uparrow x \mid x \in W\}$

⁵ Also called an **order type** (e.g. [19]) or **monotonicity type** (e.g. [20]).

the set of all **principal up-sets** $\uparrow x = \{y \in W \mid y \succeq x\}$ and $M^\infty(\mathcal{F}^+) = \{(\downarrow x)^c \mid x \in W\}$ the set of all set-theoretic complements of principal downwards closed subsets (hereafter called **co-downsets**) $\downarrow x = \{y \in W \mid x \succeq y\}$. Conversely, we will show that every perfect relevant algebra is isomorphic to the complex algebra of a Routley-Meyer frame.

Lemma 2. *In a perfect relevant algebra \mathbb{A} , it is the case that \sim^\sharp maps $J^\infty(\mathbb{A})$ into $M^\infty(\mathbb{A})$ and \sim^\flat maps $M^\infty(\mathbb{A})$ into $J^\infty(\mathbb{A})$.*

Proof. See proof in the full paper [8].

The following definition adapts a well-known method (see [16]) for obtaining dual relational structures from perfect algebras:

Definition 3. *The **prime structure** of a perfect relevant algebra $\mathbb{A} = \langle A, \wedge, \vee, \circ, \rightarrow, \sim, \mathbf{t}, \top, \perp \rangle$ is the structure $\mathbb{A}_\bullet = \langle J^\infty(\mathbb{A}), O_{\mathbf{t}}, R_\circ, * \sim \rangle$ where:*

1. $R_\circ abc$ iff $c \leq a \circ b$
2. $O_{\mathbf{t}} = \{j \in J^\infty(\mathbb{A}) \mid j \leq \mathbf{t}\}$
- and
3. $a * \sim = \lambda(\sim^\sharp a)$

Lemma 4. *\mathbb{A}_\bullet is a Routley-Meyer frame. Moreover the order \preceq on \mathbb{A}_\bullet coincides with the dual lattice order \geq restricted to $J^\infty(\mathbb{A})$.*

Proof. We begin by noting that $b \preceq c$ iff there exists $j_0 \in O_{\mathbf{t}} = \{j \in J^\infty(\mathbb{A}) \mid j \leq \mathbf{t}\}$ such that $R_\circ j_0 bc$. By definition, the latter is equivalent to $c \leq j_0 \circ b$ for some completely join-irreducible $j_0 \leq \mathbf{t}$. By the monotonicity of \circ , this implies that $c \leq \mathbf{t} \circ b$ which is equivalent to $c \leq b$ by the clause 8 of Definition 1. Conversely, if $c \leq b$, then, by the same clause, we have $c \leq \mathbf{t} \circ b = \bigvee \{j \in J^\infty(\mathbb{A}) \mid j \leq \mathbf{t}\} \circ b = \bigvee \{j \circ b \in J^\infty(\mathbb{A}) \mid j \leq \mathbf{t}\}$. Since $c \in J^\infty(\mathbb{A})$, this means there is some $j_0 \in J^\infty(\mathbb{A})$ such that $j_0 \leq \mathbf{t}$ and $c \leq j_0 \circ b$, which implies $b \preceq c$. It is clear from the construction that \mathbb{A}_\bullet is a relevance frame. In particular, the fact that $* \sim$ maps elements of $J^\infty(\mathbb{A})$ into $J^\infty(\mathbb{A})$ follows from the definition of λ and Lemma 2. We verify the six defining properties of Routley-Meyer frames in [8].

Proposition 5. *For any perfect relevant algebra \mathbb{A} it is the case that $\mathbb{A} \simeq (\mathbb{A}_\bullet)^+$.*

Proof. We show that the map $\theta : \mathbb{A} \rightarrow (\mathbb{A}_\bullet)^+$ given by $\theta(a) \mapsto \{j \in J^\infty(\mathbb{A}) \mid j \leq a\}$ is a relevant algebra isomorphism. See details in the full paper [8].

4 The calculus of the algorithm PEARL

In this section we present a calculus of rewrite rules⁶, in the style of the algorithms SQEMA [9] and ALBA [10,11], which is sound and complete for deriving

⁶ These rules can be seen as instantiations of the rules of the general-purpose algorithm ALBA [11] in the context of perfect relevant algebras. However, the fact that the latter are distributive lattice expansions allows us to present simpler formulations of these rules closer to those in [10] and, to some extent, [9]. The approximation rules presented in [11] allow for the extraction of subformulas deep from within the consequents of quasi-inequalities, subject to certain conditions, rather than the connective-by-connective style of our presentation. Although the former style of rule is also sound in the present setting, we opted for the latter as we believe it is simpler to present since the formulation requires significantly fewer auxiliary notions.

first-order frame correspondents and simultaneously proving canonicity for a large class of formulas of \mathcal{L}_R , viz. the class of *inductive (relevance) formulas* (see [7]). The algorithm PEARL and its implementation, described in the next section, are based on this calculus. The algorithm accepts (inequalities of) \mathcal{L}_R^+ formulas as input and, if it succeeds, it produces first-order formulas in the language of RM-frames that is valid in an RM-frame if and only if the original formulas are valid in the complex algebra of this RM-frame.

The rules manipulate **quasi-inequalities**⁷ of \mathcal{L}_R^+ formulas, i.e., expressions of the form $\phi_1 \leq \psi_1, \dots, \phi_n \leq \psi_n \implies \phi \leq \psi$ with $\phi, \psi, \phi_i, \psi_i \in \mathcal{L}_R^+$. In the setting of relevant algebras, quasi-inequalities are considered universally quantified over all propositional variables. Any formula $\phi \in \mathcal{L}_R^+$ can be treated as the inequality $\mathbf{t} \leq \phi$, which is a quasi-inequality with no assumptions. The inequalities not affected by the application of the rule are regarded as a **context**, which will be denoted by Γ . Given a set of inequalities Γ , we say that Γ is positive (negative) in an atom a whenever each member of Γ is positive (negative) in a . We will write $\Gamma(\alpha/p)$ for the set of inequalities obtained by uniformly substituting α for atom p in each member of Γ .

All rules that are indicated below by a double line are invertible, although the algorithm PEARL only applies the approximation rules in the downward direction.

Monotone variable elimination rules:

$$\frac{\Gamma(p) \implies \gamma(p) \leq \beta(p)}{\Gamma(\top/p) \implies \gamma(\perp/p) \leq \beta(\perp/p)} (\perp) \quad \frac{\Delta(p) \implies \beta(p) \leq \gamma(p)}{\Delta(\perp/p) \implies \beta(\top/p) \leq \gamma(\top/p)} (\top)$$

where $\beta(p)$ and Γ are positive in p , while $\gamma(p)$ and $\Delta(p)$ are negative in p .

First approximation rule:

$$\frac{\Gamma \implies \phi \leq \psi}{\mathbf{j} \leq \phi, \psi \leq \mathbf{m}, \Gamma \implies \mathbf{j} \leq \mathbf{m}}$$

where \mathbf{j} is a nominal and \mathbf{m} is a co-nominal not occurring in the premise.

Approximation rules:

$$\begin{array}{cc} \frac{\chi \rightarrow \phi \leq \mathbf{m}, \Gamma \implies \alpha \leq \beta}{\mathbf{j} \leq \chi, \mathbf{j} \rightarrow \phi \leq \mathbf{m}, \Gamma \implies \alpha \leq \beta} (\rightarrow\text{Appr-L}) & \frac{\chi \rightarrow \phi \leq \mathbf{m}, \Gamma \implies \alpha \leq \beta}{\phi \leq \mathbf{n}, \chi \rightarrow \mathbf{n} \leq \mathbf{m}, \Gamma \implies \alpha \leq \beta} (\rightarrow\text{Appr-R}) \\ \frac{\mathbf{i} \leq \chi \circ \phi, \Gamma \implies \alpha \leq \beta}{\mathbf{j} \leq \chi, \mathbf{i} \leq \mathbf{j} \circ \phi, \Gamma \implies \alpha \leq \beta} (\circ\text{Appr-L}) & \frac{\mathbf{i} \leq \chi \circ \phi, \Gamma \implies \alpha \leq \beta}{\mathbf{j} \leq \phi, \mathbf{i} \leq \chi \circ \mathbf{j}, \Gamma \implies \alpha \leq \beta} (\circ\text{Appr-R}) \\ \frac{\sim\phi \leq \mathbf{m}, \Gamma \implies \alpha \leq \beta}{\phi \leq \mathbf{n}, \sim\mathbf{n} \leq \mathbf{m}, \Gamma \implies \alpha \leq \beta} (\sim\text{Appr-L}) & \frac{\mathbf{i} \leq \sim\phi, \Gamma \implies \alpha \leq \beta}{\mathbf{j} \leq \phi, \mathbf{i} \leq \sim\mathbf{j}, \Gamma \implies \alpha \leq \beta} (\sim\text{Appr-R}) \end{array}$$

where \mathbf{j} a nominal and \mathbf{n} is a co-nominal not appearing in the premises.

⁷ In [7] these are treated set-theoretically and are called there ‘quasi-inclusions’.

Residuation rules:

$$\frac{\phi \leq \chi \vee \psi, \Gamma \implies \alpha \leq \beta}{\phi \prec \chi \leq \psi, \Gamma \implies \alpha \leq \beta} (\vee\text{Res}) \quad \frac{\phi \wedge \chi \leq \psi, \Gamma \implies \alpha \leq \beta}{\phi \leq \chi \Rightarrow \psi, \Gamma \implies \alpha \leq \beta} (\wedge\text{Res})$$

$$\frac{\phi \leq \chi \rightarrow \psi, \Gamma \implies \alpha \leq \beta}{\phi \circ \chi \leq \psi, \Gamma \implies \alpha \leq \beta} (\rightarrow\text{Res}) \quad \frac{\psi \leq \phi \hookrightarrow \chi, \Gamma \implies \alpha \leq \beta}{\phi \circ \psi \leq \chi, \Gamma \implies \alpha \leq \beta} (\hookrightarrow\text{Res})$$

Adjunction rules:

$$\frac{\phi \vee \chi \leq \psi, \Gamma \implies \alpha \leq \beta}{\phi \leq \psi, \chi \leq \psi, \Gamma \implies \alpha \leq \beta} (\vee\text{Adj}) \quad \frac{\psi \leq \phi \wedge \chi, \Gamma \implies \alpha \leq \beta}{\psi \leq \phi, \psi \leq \chi, \Gamma \implies \alpha \leq \beta} (\wedge\text{Adj})$$

$$\frac{\sim \phi \leq \psi, \Gamma \implies \alpha \leq \beta}{\sim^b \psi \leq \phi, \Gamma \implies \alpha \leq \beta} (\sim\text{Adj-L}) \quad \frac{\phi \leq \sim \psi, \Gamma \implies \alpha \leq \beta}{\psi \leq \sim^{\#} \phi, \Gamma \implies \alpha \leq \beta} (\sim\text{Adj-R})$$

Not to clutter the procedure with extra rules, we allow commuting the arguments of \wedge and \vee whenever needed before applying the rules ($\wedge\text{Adj}$) and ($\vee\text{Adj}$) above. These rules are applied exhaustively in the downward direction, and produce the same results regardless of how an expression is parenthesized.

Ackermann-rules: The Right Ackermann-rule (RAR) and Left Ackermann-rule (LAR) are subject to the following conditions:

- p does not occur in α ,
- β is positive in p ,
- γ is negative in p ,
- Γ is negative in p ,
- Δ is positive in p ,

$$\frac{\alpha \leq p, \Delta(p) \implies \gamma(p) \leq \beta(p)}{\Delta(\alpha/p) \implies \gamma(\alpha/p) \leq \beta(\alpha/p)} (\text{RAR}) \quad \frac{p \leq \alpha, \Gamma(p) \implies \beta(p) \leq \gamma(p)}{\Gamma(\alpha/p) \implies \beta(\alpha/p) \leq \gamma(\alpha/p)} (\text{LAR})$$

Note that the rules (\perp) and (\top) are, in fact, special cases of the Ackermann-rules (RAR) and (LAR), respectively.

Simplification rules: In the rules below Γ is a possibly empty list of inequalities.

$$\frac{\Gamma, \mathbf{i} \leq \phi \implies \mathbf{i} \leq \psi}{\Gamma \implies \phi \leq \psi} (\text{Simpl-Left}) \quad \frac{\Gamma, \psi \leq \mathbf{m} \implies \phi \leq \mathbf{m}}{\Gamma \implies \phi \leq \psi} (\text{Simpl-Right})$$

In the rule (Simpl-Left) the nominal \mathbf{i} must not occur in ϕ , or ψ , or any inequality in Γ . Likewise, in the rule (Simpl-Right) the co-nominal \mathbf{m} must not occur in ϕ , or ψ , or any inequality in Γ . These rules are usually applied in the post-processing, to eliminate nominals and co-nominals introduced by the approximation rules.

Example 6. We illustrate an application of PEARL on the following formula (known as axiom B2 in [31]): $(p \rightarrow q) \wedge (q \rightarrow r) \rightarrow (p \rightarrow r)$. In the full paper [8] we show that the elimination phase of PEARL succeeds and produces the following pure quasi-inequality:

$$\mathbf{i} \circ (\mathbf{i} \circ \mathbf{j}_1) \leq \mathbf{n}_1, \mathbf{j}_1 \rightarrow \mathbf{n}_1 \leq \mathbf{m} \implies \mathbf{i} \leq \mathbf{m}.$$

5 Algorithmic description of PEARL

5.1 Pre-processing and main phase of PEARL

Here we will present a deterministic algorithmic version of the procedure PEARL, which is used for the implementation.

1. Receive a formula ϕ in input.
2. If ϕ is an implication $\psi \rightarrow \theta$ set $X := \{\psi \leq \theta\}$, otherwise form the initial inequality $\mathbf{t} \leq \phi$ and set $X := \{\mathbf{t} \leq \phi\}$.
3. Now preprocess the set X by iterating steps 3a, 3b until a pass is reached in which none of the steps are applicable.
 - (a) For any $(\theta \leq \chi) \in X$, find the first positive occurrence of \vee or negative occurrence of \wedge in θ which is not in the scope of any positive occurrence of \rightarrow or a negative occurrence of \circ . Letting $\theta(\alpha \diamond \beta)$ denote θ with the occurrence of the found subterm, where $\diamond \in \{\vee, \wedge\}$, replace $\theta \leq \chi$ in X by $\theta(\alpha) \leq \chi, \theta(\beta) \leq \chi$.
 - (b) For any $(\theta \leq \chi) \in X$, find the first positive occurrence of \wedge or negative occurrence of \vee in χ which is not in the scope of any negative occurrence of \rightarrow or a positive occurrence of \circ . Again letting $\chi(\alpha \diamond \beta)$ denote χ with the found subterm, replace $\theta \leq \chi$ in X by $\theta \leq \chi(\alpha), \theta \leq \chi(\beta)$.
The preceding two “splitting” steps are justified by the distributivity of the operations \circ, \rightarrow, \sim and the adjunction rules (\vee Adj) and (\wedge Adj).
- (c) Apply the monotone variable elimination rules to all inequalities in X where they apply, replacing the involved inequalities in X with the results.
4. Proceed separately in each inequality $\phi_i \leq \psi_i$ in X . Apply the first-approximation rule to $\phi_i \leq \psi_i$ to produce the quasi-inequality $\mathbf{i} \leq \phi_i, \psi_i \leq \mathbf{m} \vdash \mathbf{i} \leq \mathbf{m}$.
5. As long as one of χ, ϕ in the approximation rules is matched by a subformula that is neither a nominal or conominal, apply these rules exhaustively to this quasi-inequality, interleaved with the splitting steps 3a-3b, where X is the set of premises. The resulting quasi-inequality has premises that are irreducible with respect to the approximation steps and splittings. This step terminates since approximation rules are only applied downwards and splittings eliminate a \wedge or \vee -symbol.
6. For each variable p in the quasi-inequality, and for each choice of polarity, $+p$ or $-p$, check if the right Ackermann-rule (for $+p$) or the left Ackermann-rule (for $-p$) can be applied to eliminate p from the premises of the quasi-inequality. This is done by applying the residuation and \sim -adjunction rules exhaustively to all premises that contain exactly one occurrence of $+p$ (or $-p$) to solve the inequality for p (if possible) and checking that p only occurs (if at all) with the opposite sign in all other premises. If possible, apply the right or left Ackermann-rule. Otherwise, p cannot be eliminated, in which case the next variable is tried. Backtracking is used to attempt to eliminate all variables in all possible orders and with either positive or negative polarity. If a variable cannot be eliminated in some particular quasi-inequality, then the algorithm stops and reports this failure.
7. If the elimination phase has succeeded on all quasi-inequalities, the algorithm proceeds to post-processing, including simplification and translation phases.

5.2 Post-processing and translation to first-order logic

This phase⁸ applies if/when the algorithm succeeds to eliminate all variables, thus ending with **pure quasi-inequalities**, containing only nominals and co-nominals, but no variables. The purpose of the post-processing is to produce a first-order condition equivalent to the pure quasi-inequality produced as a result of the main phase described in Section 5.1, and hence to the input formula. Each pure quasi-inequality produced in the elimination phase is post-processed separately to produce a corresponding first-order condition, and all these are then taken conjunctively to produce the corresponding first-order condition of the input formula. So, we focus on the case of a single pure quasi-inequality. Computing a first-order equivalent of any pure quasi-inequality can be done by straightforward application of the standard translation, but the result would usually be unnecessarily long and complicated. This can be compensated by additional post-translation equivalent simplifications in first-order logic, also taking into account the monotonicity conditions in Routley-Meyer frames. Instead, we have chosen to first apply some pre-translation simplifications of the pure quasi-inequality, using again some of the PEARL rules, and then to modify the standard translation by applying it to pure inequalities, rather than to formulas, and by extending it with a number of additional clauses dependent on the type (main connective) of the formulas on both sides of these inequalities, thus applying simplifications on the fly. For lack of space we have omitted the list of these additional post-processing rules, which can be found in the full paper [8].

The resulting modified translation Tr is not restricted to pure quasi-inequalities and can be applied to arbitrary pure formulas.

The post-processing of the pure quasi-inequality produced in Example 6 using the translation Tr is illustrated in the full paper [8]. The resulting first-order formula is $\forall x_i, x_j, x_{j_1}, y_{n_1} (Rx_i x_{j_1} y_{n_1} \rightarrow \exists x_j (Rx_i x_{j_1} x_j \wedge Rx_i x_j y_{n_1}))$ which is equivalent to the first-order condition known from [31] for the axiom B2, and to the one computed by the implementation of PEARL reported here.

6 Implementation of PEARL

Here we give a brief description of an implementation of PEARL in Python, based on the description given in Section 5. The input is a \LaTeX string using the standard syntax of relevance logic expressions. Intuitionistic implication \Rightarrow , coimplication \multimap , the right residual \multimap of \circ , and the adjoints \sim^{\sharp} and \sim^{\flat} can also appear in an input formula. The expression is parsed with a simple top-down Pratt parser [29] using standard rules of precedence. For well-formed formulas, an abstract syntax tree (AST) based on Python dictionaries and lists of arguments is created for each formula.

Five short recursive Python functions are used to transform the AST representation step-by-step according to the specific groups of PEARL transformation rules. The function `preprocess(st)` takes a \LaTeX string `st` as input and parses

⁸ This is an optimised version of the post-processing procedure outlined in [7].

it to an AST which we refer to as A . If the formula A is not well-formed, an error-string is returned. If it has a top-level \rightarrow symbol, it is replaced with a \leq to turn the formula into an inequality, and otherwise the equivalent inequality $\mathbf{t} \leq A$ is constructed. Subsequently the splitting rules and monotonicity rules from Section 4 are applied and the resulting list of inequalities is returned.

For example, with `r"p\to q\land\mathbf{t}"` as input, the formula is parsed, rewritten as $p \leq q \wedge \mathbf{t}$, then the splitting rules produce the list $[p \leq q, p \leq \mathbf{t}]$ and monotonicity returns $[\top \leq \perp, \top \leq \mathbf{t}]$.

The function `approximate(As)` takes this list as input, and applies the first approximation rule to each formula, followed by all possible left and right approximations interleaved with further applications of the splitting rule. The result is a list of quasi-equations that always have conclusion $\mathbf{i} \leq \mathbf{m}$ and premises that are irreducible with respect to the approximation and splitting rules.

The function `eliminate(As)` then attempts to apply the Ackermann-rules to each quasi-equations by selecting each variable, first with positive polarity and, if that does not succeed, then with negative polarity. Backtracking is used to ensure that all variables are tried in all possible orders. If for some quasi-equations none of the variable orders allow all variables to be eliminated, then the function reports this result. On the other hand, if for each quasi-equations some variable order succeeds to eliminate all formula variables then the resulting list of pure quasi-equations (i.e., containing no formula variables, but only nominals or co-nominals) is returned.

Since these pure quasi-equations contain redundant premises, the function `simplify(As)` is used to eliminate them, and to also apply the left and right simplification rules. Finally the variant of the standard translation described in Section 5.2 is applied to the pure quasi-equations and produces a first-order formula on the Routley-Meyer frames.

The Python code can be used in any Jupyter notebook, with the output displayed in standard mathematical notation. No special installation is needed to use the program in a personal Jupyter notebook or in a public cloud-based notebook such as Colab.google.com, and the output can be pasted into standard L^AT_EX documents. Moreover the program can be easily extended to handle the syntax of other suitable logics and lattice-ordered algebras. The resulting formula can also be translated to TPTP, Prover9 or SPASS syntax. The Python code is available at github.com/jipsen/PEARL in the form of a Jupyter notebook. It can also be copied and used directly in a browser at <https://colab.research.google.com/drive/1pOPTkmyq7vTWgYDxCTFHVRwjaLeT45uX?usp=sharing>. In the full paper [8] we provide some examples of output from the PEARL implementation.

7 Canonicity and applications to BI-logic and relation algebras

Here we report on some new theoretical and practical results related to the theory and implementation of PEARL. We begin with a theoretical result, which, for lack of space, we only sketch here.

Theorem 7. *The validity of all \mathcal{L}_R^+ -formulas on which PEARL succeeds is preserved under canonical extensions of relevant algebras.*

Proof. Let $\phi \leq \psi$ be an \mathcal{L}_R -inequality on which PEARL succeeds and let \mathbb{A} be a relevant algebra. Let $\text{PEARL}(\phi \leq \psi)$ denote the purified quasi-inequality produced from input $\phi \leq \psi$. For any \mathcal{L}_R^+ quasi-inequality $\Gamma \implies \alpha \leq \beta$, we write $\mathbb{A}^\delta \models_{\mathbb{A}} \Gamma \implies \alpha \leq \beta$ to indicate that $\Gamma \implies \alpha \leq \beta$ is true in \mathbb{A}^δ under all assignments that send propositional variables to elements of the original algebra \mathbb{A} (and nominals to $J^\infty(\mathbb{A})$ and co-nominals to $M^\infty(\mathbb{A})$) while, as usual, $\mathbb{A}^\delta \models \Gamma \implies \alpha \leq \beta$ indicates truth under *all* assignments. The following chain of equivalences establishes the canonicity of $\phi \leq \psi$:

$$\begin{array}{ccc}
\mathbb{A} \models \phi \leq \psi & & \mathbb{A}^\delta \models \phi \leq \psi \\
\Downarrow & & \\
\mathbb{A}^\delta \models_{\mathbb{A}} \phi \leq \psi & & \Downarrow \\
\Downarrow & & \\
\mathbb{A}^\delta \models_{\mathbb{A}} \text{PEARL}(\phi \leq \psi) & \Leftrightarrow & \mathbb{A}^\delta \models \text{PEARL}(\phi \leq \psi)
\end{array}$$

The uppermost bi-implication on the left is immediate by the way we defined $\models_{\mathbb{A}}$ and the fact that \mathbb{A} is a subalgebra of \mathbb{A}^δ . The lower bi-implication on the left follows by that fact that, if a quasi-inequality $\Delta' \implies \gamma' \leq \chi'$ is obtained from another, $\Delta \implies \gamma \leq \chi$, through the application of PEARL rules, then $\mathbb{A}^\delta \models_{\mathbb{A}} \Delta \implies \gamma \leq \chi$ iff $\mathbb{A}^\delta \models_{\mathbb{A}} \Delta' \implies \gamma' \leq \chi'$. This is straightforward to check for all rules except the Ackermann-rules. We refer the reader to [10] and/or [11] for the details of the latter. The horizontal bi-implication follows from the facts that, by assumption, $\text{PEARL}(\phi \leq \psi)$ is pure, and that restricting assignments of propositional variables to elements of \mathbb{A} is vacuous for pure formulas, as they contain no propositional variables. The bi-implication on the right follows by the soundness of all PEARL rules on perfect algebras, which is routine to verify.

Via the discrete duality between perfect relevant algebras and Routley-Meyer frames established in Section 3, it follows that all \mathcal{L}_R^+ -formulas on which PEARL succeeds axiomatise logics which are complete with respect to their respective first-order definable classes of Routley-Meyer frames.

As mentioned in the introduction, a large syntactically defined class of *inductive relevance formulas* in \mathcal{L}_R is defined in [7], where it is shown that PEARL succeeds for all such formulas and correctly computes their equivalent with respect to frame validity first-order definable conditions on Routley-Meyer frames. Therefore, all inductive \mathcal{L}_R^+ -formulas are canonical. This result generalizes the “canonicity via correspondence” result in [37], applied there to the fragment of \mathcal{L}_R involving of all specific relevance logic connectives only the fusion.

We can now state the results above applied to the specific implementation of PEARL reported here. However, the proof of the correctness of the implementation is beyond the scope of this paper. Still, we can report that the implementation has succeeded on all axioms A1-A9, B1-B30, and D1-D8 listed in the appendix of [7], copied there from [31], and has computed first-order conditions equivalent to those known from the literature.

Bunched implication logic [30] is closely related to a negation-free relevance logic. The algebraic semantics of bunched implication logic is given by bunched implication algebras, or BI-algebras. They are defined by axioms 1-3 and 7-9 of Definition 1 together with a new binary operation symbol \Rightarrow such that

10. $a \wedge b \leq c$ iff $a \leq b \Rightarrow c$ (hence \Rightarrow is a Heyting algebra implication) 11. $(a \circ b) \circ c = a \circ (b \circ c)$,
 12. $a \circ b = b \circ a$.

The steps of the PEARL algorithm are not affected by these addition axioms (although additional rules for the associativity and commutativity of \circ could be added), and the relational semantic structures of BI-logic and BI-algebras are precisely Routley-Meyer frames. However in BI-logic the notation differs slightly, since $\rightarrow, \circ, \Rightarrow$ are replaced by $\multimap, *, \rightarrow$, and this alternative notation is user-selectable in the implementation.

Lastly, we note that the algorithm PEARL can also be applied to relation algebras, as they form a subvariety of relevant algebras extended with a Heyting implication \Rightarrow . An axiomatization of relation algebras in this setting consists of axioms of relevant algebras (1-9 from Definition 1), 10, 11 above and⁹

13. $(x \Rightarrow \perp) \Rightarrow \perp = x$ 14. $x \rightarrow y = \sim(\sim y \circ x)$,
 (hence \Rightarrow is a classical implication and $x \Rightarrow \perp$ is denoted $\neg x$), 15. $x^\smile = \sim(x \Rightarrow \perp)$,
 16. $(x \circ y)^\smile = y^\smile \circ x^\smile$.

Axiom 13 ensures that the lattice structure is a Boolean algebra, hence the partial order in the Routley-Meyer frames of a relation algebra is an antichain. In the theory of relation algebras these frames are known as ‘atom structures’, defined in [27, Def. 2.1]. For the application of PEARL to relation algebras, it suffices to replace the converse operation by the term $\sim(x \Rightarrow \perp)$ and to interpret any \preceq symbol in the resulting first-order formula as an equality symbol. Note that relevant negation $\sim x$ can, in turn, also be defined via the relation algebra term $(\neg x)^\smile$. While there is a long history of Sahlqvist formulas and correspondence theory for Boolean algebras with operators [13,25], it is interesting to note that the PEARL algorithm and its implementation can be adapted to relation algebras and covers the more general class of inductive formulas.

8 Concluding remarks

In this paper we have re-interpreted the algorithm PEARL from [7] as an algorithm which manipulates quasi-inequalities interpreted over perfect relevant algebras. Implementing the algorithm in a way that produces reasonably optimal (in size) versions of first-order correspondents required detailed specifications and strategic choices in the pre-processing, main, and post-processing phases (Sections 5.1 and 5.2) and in the specialized post-processing and translation procedure, refining the normal standard translation, developed in Section 5.2. It is easy to see that the complexity of the problem solved by PEARL is in NP-time

⁹ While this equational basis for relation algebras appears to be quite long, it can be shown that axioms 3-7 are redundant. Hence, it is comparable in length to the original axiomatization of relation algebras.

because, once the correct ordering or elimination of the variables is selected, PEARL completes its work in polynomial time. However, theoretically, it may take trying an exponential number of such orderings until success. Whether this is possible is not yet known, so the optimal complexity of the problem is still under investigation.

References

1. Badia, G.: On Sahlqvist formulas in relevant logic. *Journal of Philosophical Logic* **47**(4), 673–691 (2018)
2. van Benthem, J.: *Modal Correspondence Theory*. Ph.D. thesis, Mathematisch Instituut & Instituut voor Grondslagenonderzoek, University of Amsterdam (1976)
3. van Benthem, J.: A note on dynamic arrow logic. Tech. Rep. LP-92-11, ILLC, University of Amsterdam (1992)
4. van Benthem, J.: Correspondence theory. In: Gabbay, D., Guenther, F. (eds.) *Handbook of Philosophical Logic*, 2nd ed., vol. 3, pp. 325–408. Springer - Science+Business Media, B.V. (2001)
5. Bimbó, K., Dunn, J.M., Maddux, R.D.: Relevance logics and relation algebras. *Rev. Symb. Log.* **2**(1), 102–131 (2009)
6. Conradie, W., Ghilardi, S., Palmigiano, A.: Unified correspondence. In: Baltag, A., Smets, S. (eds.) *Johan van Benthem on Logic and Information Dynamics, Outstanding Contributions to Logic*, vol. 5, pp. 933–975. Springer (2014)
7. Conradie, W., Goranko, V.: Algorithmic correspondence for relevance logics I. The algorithm PEARL. In: Düntsch, I., Mares, E. (eds.) *Alasdair Urquhart on Nonclassical and Algebraic Logic and Complexity of Proofs*, pp. 163–209. Springer (2021), <https://www2.philosophy.su.se/goranko/papers/PEARL.pdf>
8. Conradie, W., Goranko, V., Jipsen, P.: Algorithmic correspondence for relevance logics, bunched implication logics, and relation algebras: the algorithm PEARL and its implementation (technical report). Technical report, available from <https://arxiv.org/abs/2108.06603> (2021)
9. Conradie, W., Goranko, V., Vakarelov, D.: Algorithmic correspondence and completeness in modal logic, I. The core algorithm SQEMA. *Logical Methods in Computer Science* **2**(1:5) (2006)
10. Conradie, W., Palmigiano, A.: Algorithmic correspondence and canonicity for distributive modal logic. *Ann. Pure Appl. Logic* **163**(3), 338 – 376 (2012)
11. Conradie, W., Palmigiano, A.: Algorithmic correspondence and canonicity for non-distributive logics. *Ann. Pure Appl. Logic* **170**(9), 923–974 (2019). <https://doi.org/10.1016/j.apal.2019.04.003>
12. Dahlqvist, F., Pym, D.: Coalgebraic completeness-via-canonicity for distributive substructural logics. *J. Log. Algebr. Methods Program.* **93**, 1–22 (2017). <https://doi.org/10.1016/j.jlamp.2017.07.002>
13. de Rijke, M., Venema, Y.: Sahlqvist’s theorem for Boolean algebras with operators with an application to cylindric algebras. *Stud. Log.* **54**(1), 61–78 (1995). <https://doi.org/10.1007/BF01058532>
14. Doumane, A., Pous, D.: Non axiomatisability of positive relation algebras with constants, via graph homomorphisms. In: Konnov, I., Kovács, L. (eds.) *Proc. of CONCUR 2020. LIPIcs, Schloss Dagstuhl*, vol. 171, pp. 29:1–29:16 (2020)
15. Dunn, J.M.: Arrows pointing at arrows: Arrow logic, relevance logic, and relation algebras. In: Baltag, A., Smets, S. (eds.) *Johan van Benthem on Logic and Information Dynamics*, pp. 881–894. Springer (2014)

16. Dunn, J.M., Gehrke, M., Palmigiano, A.: Canonical extensions and relational completeness of some substructural logics. *J. Symb. Logic* pp. 713–740 (2005)
17. Dunn, J., Restall, G.: Relevance logic. In: Gabbay, D., Guenther, F. (eds.) *Handbook of Philosophical Logic*, 2nd Edition, vol. 6, pp. 1–128. Springer - Science+Business Media, B.V. (2002)
18. Galatos, N., Jipsen, P., Kowalski, T., Ono, H.: *Residuated lattices: an algebraic glimpse at substructural logics*. Elsevier (2007)
19. Gehrke, M., Nagahashi, H., Venema, Y.: A Sahlqvist theorem for distributive modal logic. *Annals of Pure and Applied Logic* **131**, 65–102 (2005)
20. Gehrke, M., Jónsson, B.: Bounded distributive lattice expansions. *Mathematica Scandinavica* pp. 13–45 (2004)
21. Goranko, V., Vakarelov, D.: Sahlqvist formulae in hybrid polyadic modal languages. *Journal of Logic and Computation* **11(5)**, 737–754 (2001)
22. Goranko, V., Vakarelov, D.: Sahlqvist formulas unleashed in polyadic modal languages. In: Wolter, F., Wansing, H., de Rijke, M., Zakharyashev, M. (eds.) *Advances in Modal Logic*. vol. 3, pp. 221–240. World Scientific, Singapore (2002)
23. Goranko, V., Vakarelov, D.: Elementary canonical formulae: Extending Sahlqvist’s theorem. *Ann. Pure Appl. Logic* **141(1-2)**, 180–217 (2006)
24. Hirsch, R., Mikuláš, S.: Positive fragments of relevance logic and algebras of binary relations. *Rev. Symb. Log.* **4(1)**, 81–105 (2011)
25. Jónsson, B.: On the canonicity of Sahlqvist identities. *St.Log.* **53(4)**, 473–491 (1994)
26. Kowalski, T.: Relevant logic and relation algebras. In: Galatos, N., Kurz, A., Tsinkakis, C. (eds.) *TACL 2013. Sixth International Conference on Topology, Algebra and Categories in Logic*. EPiC Series in Computing, vol. 25, pp. 125–128 (2014)
27. Maddux, R.: Some varieties containing relation algebras. *Trans. Am. Math. Soc.* **272**, 501–526 (1982)
28. Maddux, R.D.: Relevance logic and the calculus of relations. *Rev. Symb. Log.* **3(1)**, 41–70 (2010), <https://doi.org/10.1017/S1755020309990293>
29. Pratt, V.R.: Top down operator precedence. In: Fischer, P.C., Ullman, J.D. (eds.) *Conference Record of the ACM Symposium on Principles of Programming Languages*, Boston, Massachusetts, USA, October 1973. pp. 41–51. ACM Press (1973)
30. Pym, D.: *The Semantics and Proof Theory of the Logic of Bunched Implications*. Applied Logic Series, Springer Netherlands (2002)
31. Routley, R., Meyer, R., Plumwood, V., Brady, R.: *Relevant Logics and its Rivals (Volume I)*. Ridgeview, CA (1982)
32. Sahlqvist, H.: Correspondence and completeness in the first and second-order semantics for modal logic. In: Kanger, S. (ed.) *Proc. of the 3rd Scandinavian Logic Symposium*, Uppsala 1973. pp. 110–143. Springer-Verlag, Amsterdam (1975)
33. Sambin, G., Vaccaro, V.: A new proof of Sahlqvist’s theorem on modal definability and completeness. *The Journal of Symbolic Logic* **54(3)**, 992–999 (1989)
34. Seki, T.: A Sahlqvist theorem for relevant modal logics. *Studia Logica* **73(3)**, 383–411 (2003)
35. Suzuki, T.: Canonicity results of substructural and lattice-based logics. *The Review of Symbolic Logic* **4(1)**, 1–42 (2011). <https://doi.org/10.1017/S1755020310000201>
36. Suzuki, T.: A Sahlqvist theorem for substructural logic. *The Review of Symbolic Logic* **6(2)**, 229–253 (2013). <https://doi.org/10.1017/S1755020313000026>
37. Urquhart, A.: Duality for algebras of relevant logics. *Studia Logica* **56(1/2)**, 263–276 (1996), <https://doi.org/10.1007/BF00370149>