Correspondence Theory for Intuitionistic Łukasiewicz Logic

Andrew Lewis-Smith[1], Zhiguang Zhao[2]

[1]Middlesex University, United Kingdom
[2]Taishan University, P.R. China

Abstract
In the present paper, we develop the correspondence theory for $\mathbf{GBL}_{ewf}$, which can be viewed as the intuitionistic counterpart of Łukasiewicz logic. Our methodology follows [1], adopting the algorithmic correspondence theory method. We identify the class of inductive formulas and develop an algorithm for computing the many-valued first-order correspondents of given inductive formulas. The key step is to identify interpretations of nominals and conominals (corresponding to singletons and their complements) in $\mathbf{GBL}_{ewf}$ that are first order definable.

Keywords
Intuitionistic Łukasiewicz logic, Correspondence theory, Inductive formulas

1. Introduction

Intuitionistic Łukasiewicz logic. The system $\mathbf{GBL}_{ewf}$ [2], or the equivalent natural deduction presentation intuitionistic Łukasiewicz logic, can be viewed as a fuzzy logic [3, 4, 5] which can serve as a base system for Hajek's $\mathbf{BL}$ [6], Gödel logic, and other fuzzy logics involving the axiom of divisibility [7, 8, 9]. It is is also the intuitionistic counterpart of Łukasiewicz logic [10]: it has a relational semantics [2] in which the frame forms a partial order, similar to intuitionistic logic, but each node is $[0, 1]$-valued rather than $\{0, 1\}$-valued. As such, it is natural to investigate how correspondence theory for modal and intuitionistic logics can be generalized to this fuzzy logic and its axiomatic extensions.

Correspondence theory. Correspondence theory [11, 12] takes as a starting point the observation from modal logic that the frames of Kripke semantics can be understood as having first-order relational equivalents. *Sahlqvist* theory specifies syntactic conditions for identifying first-order correspondents of frame conditions given by formulas in modal logic. The *unified correspondence* theory [13, 14, 15], proposes a general algebraic definition for inductive formulas that uniformly applies to many logics.

Recent Relevant Work. Britz et al. [1] extended Sahlqvist correspondence theory to the many-valued modal logics as defined by Fitting [16], assuming a perfect Heyting algebra as truth value space. They present standard translations between many-valued modal languages and suitably defined first and second-order correspondence languages, proving correctness of the translation. They define many-valued analogues of the Sahlqvist and inductive formulas and adapt the well-known ALBA algorithm to that setting, effectively computing many-valued local frame correspondents for all many-valued Sahlqvist and inductive formulas.

Our work. Our approach in this present work is *algorithmic*: termination reduces to success of the algorithm on inductive formulas, i.e. there is a run terminating on inductive formula inputs. Present work extends the recent research in correspondence theory, particularly Britz et. al [1] and Fussner [17] on relational semantics for $\mathbf{GBL}_{ewf}$ and extensions to algorithmically identify frame conditions corresponding to formulas in $\mathbf{GBL}_{ewf}$. The key step is to construct interpretations of nominals which

Workshop Proceedings, 2026
a.lewis-smith@mdx.ac.uk (A. Lewis-Smith); zhaozhiguang23@gmail.com (Z. Zhao)
0009-0000-0203-0615 (A. Lewis-Smith); 0000-0001-5637-945X (Z. Zhao)

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-07-23

act as the counterpart of singletons in $\mathbf{GBL}_{ewf}$, whose standard translations will be essentially first-order. The present paper generalises insights from [1] with respect to perfect Heyting algebras to $[0,1]_{\mathbf{MV}}$, and identifies the Sahlqvist and inductive formulas in line with contemporary correspondence theory.

Structure of the paper. In Sections 2 and 3, we give all the necessary preliminaries for $\mathbf{GBL}_{ewf}$ and correspondence theory, respectively. Section 4 gives the definition of inductive formulas for $\mathbf{GBL}_{ewf}$. Section 5 defines the algorithm ALBA for $\mathbf{GBL}_{ewf}$. Section 6 proves the soundness of the algorithm with respect to the semantics. Section 7 shows that the algorithm succeeds on all inductive formulas.

2. Preliminaries

In this section, we provide preliminaries for $\mathbf{GBL}_{ewf}$. For more details, see [2].

2.1. $\mathbf{GBL}_{ewf}$

The formulas of $\mathbf{GBL}_{ewf}$ are defined as follows:

$$\phi := p|\bot|\phi \vee \phi|\phi \wedge \phi|\phi \otimes \phi|\phi \to \phi$$

We use $\mathcal{L}_{\otimes}$ to denote the language and $\mathsf{Form}(\mathcal{L}_{\otimes})$ to denote the set of all $\mathcal{L}_{\otimes}$-formulas. We use $\neg\phi$ to denote $\phi \to \bot$ and $\top$ to denote $\bot \to \bot$.

A word of interpretation regarding operations introduced above: one can understand $\to, \otimes$ as forming a residuated pair, i.e. $\varphi \otimes \psi \leq \theta$ iff $\varphi \leq \psi \to \theta$, in which $\otimes$ is understood as a second kind of conjunction, in some sense logically stronger than $\wedge$ – this $\otimes$ intuitively results from removing the rule of contraction from Intuitionistic logic. From a residuated lattices perspective (for which $\mathbf{GBL}_{ewf}$-algebras form the corresponding algebraic system), $\otimes$ is the monoidal multiplication and $\to$ is the corresponding division operation.

We present the Hilbert system for $\mathbf{GBL}_{ewf}$:

(A1) $\phi \to \phi$
(A2) $(\phi \to \psi) \to ((\psi \to \chi) \to (\phi \to \chi))$
(A3) $(\phi \otimes \psi) \to (\psi \otimes \phi)$
(A4) $(\phi \otimes \psi) \to \psi$
(A5) $(\phi \to (\psi \to \chi)) \to ((\phi \otimes \psi) \to \chi))$
(A6) $((\phi \otimes \psi) \to \chi)) \to (\phi \to (\psi \to \chi))$
(A7) $(\phi \otimes (\phi \to \psi)) \to (\phi \wedge \psi)$
(A8) $(\phi \wedge \psi) \to (\phi \otimes (\phi \to \psi))$
(A9) $(\phi \wedge \psi) \to (\psi \wedge \phi)$
(A10) $\phi \to (\phi \vee \psi)$
(A11) $\psi \to (\phi \vee \psi)$
(A12) $((\phi \to \psi) \wedge (\chi \to \psi)) \to ((\phi \vee \chi) \to \psi)$
(A13) $\bot \to \phi$
(MP) From $\phi, \phi \to \psi$ infer ψ.

We use $\vdash \phi$ to denote that ϕ is provable in this system.

2.2. Semantics for $\mathbf{GBL}_{ewf}$

In this section, we give the semantics for $\mathbf{GBL}_{ewf}$.

Definition 1. $\mathcal{A} = \langle A, \wedge, \vee, \otimes, 1, \to \rangle$ is called a *(commutative) residuated lattice* if

- $\langle A, \wedge, \vee, \otimes, 1 \rangle$ is a (commutative) lattice-ordered monoid.
- $x \otimes y \leq z$ if and only if $x \leq y \to z$.

We often use A to denote the (commutative) residuated lattice.

Definition 2 ($\mathbf{GBL}_{ewf}$-algebra)**.** A $\mathbf{GBL}_{ewf}$-algebra is a commutative residuated lattice satisfying the divisibility property: if $x \leq y$ then $y \otimes (y \to x) = x$.

Definition 3 ($\mathbf{MV}$-algebra)**.** An $\mathbf{MV}$-algebra is a commutative residuated lattice satisfying the following properties:

- divisibility property: if $x \leq y$ then $y \otimes (y \to x) = x$;
- pre-linearity: $(x \to y) \vee (y \to x)$;
- bounded from below by $\bot$, i.e. $\bot \leq x$ for all $x \in A$;
- integrality: 1 is the top element of the lattice, i.e. $x \leq 1$ for all $x \in A$ (here 1 is also denoted as $\top$).
- $(x \to \bot) \to \bot = x$, for all $x \in A$.

An $\mathbf{MV}$-algebra is an $\mathbf{MV}$-chain if its underlying order is linear.

Definition 4 (Standard $\mathbf{MV}$-chain)**.** For $x \in [0,1]$, let $\overline{x} := 1 - x$. The *standard* $\mathbf{MV}$*-chain* $[0,1]_{\mathrm{MV}} := ([0,1], \top, \bot, \wedge, \vee, \otimes, \to)$ is defined as follows: $[0,1]$ is the unit interval, and the constants and binary operations are defined as follows:

$$\begin{aligned}
\top &:= 1\\
\bot &:= 0\\
x \wedge y &:= \min\{x, y\}\\
x \vee y &:= \max\{x, y\}\\
x \otimes y &:= \max\{0, \overline{\overline{x} + \overline{y}}\} = \max\{0, x + y - 1\}\\
x \to y &:= \min\{1, \overline{\overline{y} - \overline{x}}\} = \min\{1, y - x + 1\}.
\end{aligned}$$

Lemma 5. *The followings hold in the standard MV-chain* $[0,1]_{\mathrm{MV}}$*:*

(i) *For all* $n \geq 2$, $x_1 \otimes \ldots \otimes x_n = \max\{0, \overline{\overline{x_1} + \ldots + \overline{x_n}}\}$.

(ii) $x_1 \otimes \ldots \otimes x_n \leq x_i$, *for* $i \in \{1, \ldots, n\}$

(iii) *if* $x \leq y \vee z$ *and* $u \otimes y \leq v$ *and* $u \otimes z \leq v$ *then* $u \otimes x \leq v$.

(iv) *If* $x \leq y$ *and* $u \leq y \to z$ *then* $x \otimes u \leq z$.

(v) *If* $x \leq y$ *and* $z \leq w$ *then* $x \otimes z \leq y \otimes w$.

(vi) *If* $x \leq y$ *and* $v \otimes y \leq z$ *then* $v \otimes x \leq z$.

(vii) $x \otimes (x \to y) = y \otimes (y \to x)$.

Proof. See [2, Lemma 2.8]. □

To interpret the classical Łukasiewicz logic, the standard MV-chain is enough. To interpret $\mathbf{GBL}_{ewf}$, we need an additional partial order such that each propositional variable is interpreted as the following *sloping function*:

Definition 6 (Sloping function)**.** Let $\mathcal{W} = \langle W, \preceq \rangle$ be a partial order, and let $v \prec w$ indicate that $v \preceq w$ and $v \neq w$. A function $f : W \to [0,1]$ is *monotone* if $v \preceq w$ implies $f(v) \leq f(w)$ for all $v, w \in W$. A monotone function $f : W \to [0,1]$ is a *sloping function* if for all $w \in W$, $f(w) > \bot$ implies $\forall v \succ w(f(v) = \top)$.

Remark 7. The definition implies that if $f : W \to [0,1]$ is a sloping function and $f(w) < \top$ then $\forall v \prec w(f(v) = \bot)$. That is, along any increasing chain $w_1 \prec w_2 \prec \ldots \prec w_n$, there can only be at most one point w_i such that $\bot < f(w_i) < \top$, and for $j < i$ we must have $f(w_j) = \bot$, and for $j > i$ we must have $f(w_j) = \top$.

Lemma 8. *If $f, g : W \to [0,1]$ are sloping functions, then so are $f \wedge g$, $f \vee g$ and $f \otimes g$:*

$$\begin{array}{lcl} (f \wedge g)(w) & := & \min\{f(w), g(w)\} \\ (f \vee g)(w) & := & \max\{f(w), g(w)\} \\ (f \otimes g)(w) & := & \max\{0, \overline{\overline{f(w)} + \overline{g(w)}}\}. \end{array}$$

Proof. See [2, Lemma 3.2]. □

Definition 9 (Bova-Montagna structure)**.** A *Bova-Montagna structure* (BM-structure for short) is a pair $\mathcal{M} = \langle \mathcal{W}, V \rangle$ where $\mathcal{W} = \langle W, \preceq \rangle$ is a poset, and for any propositional variable p, the function $\lambda w.V(p, w) : W \to [0,1]$ is a sloping function.

Definition 10 (Floor function)**.** Let $\lfloor \cdot \rfloor$ be the *floor function* on $[0,1]_{\mathrm{MV}}$ defined as follows:

$$\lfloor x \rfloor := \begin{cases} \top & \text{if} \quad x = \top \\ \bot & \text{if} \quad x < \top. \end{cases}$$

which is known as the "Monteiro-Baaz Δ-operator." Given any (not necessarily sloping) function $f : W \to [0,1]$ and any $w \in W$, define $\lfloor \inf \rfloor_{v \succeq w} f(v)$ as follows:

$$\lfloor \inf \rfloor_{v \succeq w} f(v) := \min\{f(w), \inf_{v \succ w} \lfloor f(v) \rfloor\}$$

where $\inf_{v \succ w} \lfloor f(v) \rfloor$ is the infimum of the set $\{\lfloor f(v) \rfloor : v \succ w\} \subseteq [0,1]$.

Lemma 11. *This definition of $\lfloor \inf \rfloor_{v \succeq w}$ can also be equivalently written as*

$$\lfloor \inf \rfloor_{v \succeq w} f(v) := \begin{cases} f(w) & \text{if} \, \forall v \succ w(f(v) = \top) \\ \bot & \text{if} \, \exists v \succ w(f(v) < \top) \end{cases}$$

and for any $f : W \to [0,1]$ the function $\lambda w.\lfloor \inf \rfloor_{v \succeq w} f(v)$ is a sloping function.

Proof. See [2, Lemma 3.5]. □

Definition 12 (Kripke semantics for $\mathcal{L}_{\otimes}$)**.** Given a BM-structure $\mathcal{M} = \langle \mathcal{W}, V \rangle$, the valuation function $V(p, w)$ on propositional variables p can be extended to all $\mathcal{L}_{\otimes}$-formulas as:

$$\begin{array}{lcl} V(\bot, w) & := & \bot \\ V(\phi \wedge \psi, w) & := & V(\phi, w) \wedge V(\psi, w) \\ V(\phi \vee \psi, w) & := & V(\phi, w) \vee V(\psi, w) \\ V(\phi \otimes \psi, w) & := & V(\phi, w) \otimes V(\psi, w) \\ V(\phi \to \psi, w) & := & \lfloor \inf \rfloor_{v \succeq w}(V(\phi, v) \to V(\psi, v)) \end{array}$$

where the operations on the right-hand side are the operations on the standard MV-chain $[0,1]_{\mathrm{MV}}$, and $\lfloor \inf \rfloor_{v \succeq w}$ is as in Definition 10.

Lemma 13. *For any formula ϕ the function $\lambda w.V(\phi, w) : W \to [0,1]$ is a sloping function.*

Proof. See [2, Lemma 3.7]. □

Definition 14. Let $\Gamma = \psi_1, \ldots, \psi_n$. We say that:

- ϕ holds in a BM-structure $\mathcal{M}$ (notation: $\mathcal{M} \Vdash \phi$) if $V(\phi, w) = \top$ for all $w \in W$.
- ϕ is valid on a poset $\mathcal{W} = \langle W, \preceq \rangle$ (notation: $\mathcal{W} \Vdash \phi$) if $\mathcal{M} \Vdash \phi$ for all BM-structures based on $\mathcal{W}$.
- ϕ is valid (notation: $\Vdash \phi$), if $\mathcal{W} \Vdash \phi$ for all posets $\mathcal{W}$.

We have the following soundness and completeness theorem for $\mathbf{GBL}_{ewf}$:

Theorem 15. $\vdash \phi$ *iff* $\Vdash \phi$.

Proof. See [2, Theorems 3.13 and 3.14]. □

2.3. Poset Product Construction

We briefly discuss an important algebraic construction due to Peter Jipsen and Franco Montagna [5]. Here we follow Jipsen and Montagna and Fussner's [17] presentation of the idea.

Let $X = (X, \leq)$ be a poset and let $\{A_x : x \in X\}$ be a collection of residuated lattices indexed by the poset X. We assume that all A_x share the same neutral element $\top$ and that all A_x that are bounded share the same minimum element $\bot$.

Suppose that if x is not minimal, then A_x is integral and if x is not maximal then A_x is bounded. The *poset product* $\prod_{x \in (X,\leq)} A_x$ is the algebra defined as follows[1]:

Definition 16. The domain of $\prod_{x \in (X,\leq)} A_x$ is the set of all maps h on X such that for all $x \in X$:

1. $h(x) \in A_x$ and
2. if $h(x) \neq \top$ then for all $y < x$, $h(y) = \bot$
3. The monoid operation and the lattice operations are defined pointwise.
4. The residuals are defined by:

$$(h \to g)(x) = \begin{cases} h(x) \to g(x) & \text{if} \quad g(y) \leq h(y) \quad \text{for all} \quad y > x \\ \bot & \text{otherwise .} \end{cases}$$

Now let $(X, \leq)$ be a poset and $\{A_x : x \in X\}$ is an indexed collection of integral bounded residuated lattices. We set $B = \prod_{x \in (X,\leq)} A_x$ and the map $\sigma \colon B \to B$ we define by

$$\sigma(f)(x) = \begin{cases} f(x) & \text{if } f(y) = 1 \text{ for all } y > x \\ 0 & \text{if there exists } y > x \text{ with } f(y) \neq 1. \end{cases}$$

The map σ is a conucleus on B by [5], and the *poset product* of this indexed family $\{A_x : x \in X\}$ is the algebra B_σ, which in the literature (see e.g. [17]) is sometimes denoted

$$B_\sigma = \prod_{(X,\leq)} A_x.$$

Thus one can view poset products as direct products with an indexing set that is a poset $(X, \leq)$ rather than a set.

This σ or $\lfloor \inf \rfloor$ resembles a box modality; See Fussner's [17] for discussion. From [5] and [17], one can find following results (given in Fussner's Lemma 3.5 and Jipsen and Montagna's Theorem 6.2):

[1] NB. the note on notation in [17]. We provide here the standard notation from the literature and inform the reader where we occasionally depart from that standard.

Theorem 17. *Let $X = (X, \leq)$ be a poset and let $\{A_x : x \in X\}$ be a collection of residuated lattices indexed by the poset X. Again, set:*

$$B = \prod_{x \in X} A_x$$

If A_x is an MV-algebra for each $x \in X$, then B is a GBL-algebra.

3. Preliminaries on correspondence theory

Hereon, we study the correspondence theory of $\mathbf{GBL}_{ewf}$ using algorithmic and algebraic correspondence theory in [13, 14]. In developing the correspondence theory of $\mathbf{GBL}_{ewf}$, we must select correct candidates for the interpretations of nominals, subject to:

- The interpretation of nominals should be related to singletons, so that their standard translations into the many-valued first-order language can be expressed in a first-order language without employing second-order variables (i.e. unary predicate symbols).
- The possible interpretations of nominals should be join-dense in the complete algebra of possible values of propositional variables, i.e. each possible interpretation of propositional variables can be written as the join of nominals.

In what follows, we find possible interpretations for nominals and conominals, then define the expanded propositional language for the correspondence algorithm and semantics. We define the first-order correspondence language and interpretation as well as the standard translation. Again, we use a many-valued first-order language, where formula values range over $[0, 1]$.

3.1. Finding the interpretations for nominals and conominals

We must ensure the set of possible interpretations of nominals is join-dense in the complete algebra of possible values as a condition for algorithm correctness. So we take the sloping functions having a least element in the poset with non-zero value. Then the possible interpretations of nominals resemble principal upsets with the generating point having a non-zero value, and each sloping function can be the join of this kind of particular kind of sloping function.

Definition 18. Let $\mathcal{W} = \langle W, \preceq \rangle$ be a partial order, and $\mathsf{Slop}(\mathcal{W})$ be the set of all sloping functions. We define $\mathsf{J}(\mathcal{W})$ and $\mathsf{M}(\mathcal{W})$[2] as follows:

- $\mathsf{J}(\mathcal{W})$ consists of all sloping functions of the form $f_{w,a} : W \to [0, 1]_{\mathrm{MV}}$ where $w \in W$ and $a \in (0, 1]$ such that
$$f_{w,a}(v) = \begin{cases} 1 & \text{if } v > w \\ a & \text{if } v = w \\ 0 & \text{otherwise.} \end{cases}$$
- $\mathsf{M}(\mathcal{W})$ consists of all sloping functions of the form $g_{w,a} : W \to [0, 1]_{\mathrm{MV}}$ where $w \in W$ and $a \in [0, 1)$ such that
$$g_{w,a}(v) = \begin{cases} 0 & \text{if } v < w \\ a & \text{if } v = w \\ 1 & \text{otherwise.} \end{cases}$$

Indeed, we can prove that every sloping function $f \in \mathsf{Slop}(\mathcal{W})$ can be represented as the join of sloping functions in $\mathsf{J}(\mathcal{W})$.[3] Let us consider $\mathsf{Slop}(\mathcal{W})$ as a complete lattice, where for any $\{f_i \mid i \in I\} \subseteq \mathsf{Slop}(\mathcal{W})$, any $w \in \mathcal{W}$,

[2]Here J and M mean that the respective sets are join-dense and meet-dense in the complete algebra of all sloping functions.

[3]An important note: $g_{w,a}$ and $f_{w,a}$ might appear to be the same, the only thing changing being the domain of the functions $[0, 1)$ or $(0, 1]$. But: f generates an upset, and g generates the complement of a downset. Also, sometimes v, w are not comparable, so that f, g are distinct.

- $(\bigvee_{i\in I} f_i)(w) = \bigvee_{i\in I} f_i(w)$,
- $(\bigwedge_{i\in I} f_i)(w) = \bigwedge_{i\in I} f_i(w)$.

Since the $\mathbf{MV}$-chain $[0,1]_{\mathrm{MV}}$ is complete as a lattice, the definition above is well-defined, and it is easy to see that $\bigvee_{i\in I} f_i$ and $\bigwedge_{i\in I} f_i$ are also sloping functions on $\mathcal{W}$.

Proposition 19. *For any sloping function $f \in \mathsf{Slop}(\mathcal{W})$, let $S(f) := \{w \in W \mid f(w) > 0\}$ be the* support *of f, and let $F := \{f_{w,f(w)} \mid w \in S(f)\}$, then $f = \bigvee F$ and $F \subseteq \mathsf{J}(\mathcal{W})$.*

Proof. We invite the reader to inspect the full proof given in [18].

□

Proposition 20. *For any sloping function $f \in \mathsf{Slop}(\mathcal{W})$, let $C(f) := \{w \in W \mid f(w) < 1\}$ be the* cosupport *of f, and let $G := \{g_{w,f(w)} \mid w \in C(f)\}$, then $f = \bigwedge G$ and $G \subseteq \mathsf{M}(\mathcal{W})$.*

Proof. The interested reader can inspect the full proof given in [18].

□

3.2. The expanded propositional language $\mathcal{L}_{\otimes}^{+}$

The expanded language $\mathcal{L}_{\otimes}^{+}$ is obtained from the basic propositional language $\mathcal{L}_{\otimes}$ by adding nominals and conominals.

Definition 21 (Formulas of $\mathcal{L}_{\otimes}^{+}$)**.** The formulas of $\mathcal{L}_{\otimes}^{+}$ are defined as follows:

$$\phi := p|\bot|\top|\mathbf{i}|\mathbf{m}|\phi\vee\phi|\phi\wedge\phi|\phi\otimes\phi|\phi\to\phi$$

where $\mathbf{i} \in \mathsf{Nom}$, $\mathbf{m} \in \mathsf{Cnom}$, $p \in \mathsf{Prop}$ and Nom, Cnom and Prop are disjoint sets of *nominals*, *conominals* and propositional variables, respectively. We use $\mathsf{Form}(\mathcal{L}_{\otimes}^{+})$ to denote the set of all $\mathcal{L}_{\otimes}^{+}$-formulas. We say that a formula is *pure* if it does not contain any propositional variables in Prop.

Definition 22 (Inequalities and quasi-inequalities)**.** An *inequality* is of the form $\phi \leq \psi$ where $\phi, \psi \in \mathsf{Form}(\mathcal{L}_{\otimes}^{+})$. A *quasi-inequality* is of the form $\phi_1 \leq \psi_1 \ \&\ \ldots\ \&\ \phi_n \leq \psi_n \ \Rightarrow\ \phi \leq \psi$ where $\phi, \phi_1, \ldots, \phi_n, \psi, \psi_1, \ldots, \psi_n \in \mathsf{Form}(\mathcal{L}_{\otimes}^{+})$. Pure inequalities and pure quasi-inequalities are the ones which do not contain any propositional variables.

Definition 23 (Kripke semantics for $\mathcal{L}_{\otimes}^{+}$)**.** Given any BM-structure $\mathcal{M} = \langle \mathcal{W}, V\rangle$, the valuation function is extended to Nom and Cnom such that $\lambda w.V(\mathbf{i}, w) \in \mathsf{J}(\mathcal{W})$ and $\lambda w.V(\mathbf{m}, w) \in \mathsf{M}(\mathcal{W})$.

The valuation function $V(\phi, w)$ can be extended to all $\mathcal{L}_{\otimes}^{+}$-formulas as in Definition 12.

Definition 24 (Semantics for inequalities and quasi-inequalities)**.** Given any BM-structure $\mathcal{M} = \langle \mathcal{W}, V\rangle$,

- For inequalities, $\mathcal{M} \Vdash \phi \leq \psi$ iff $V(\phi, w) \leq V(\psi, w)$ for all $w \in W$, and $\mathcal{W} \Vdash \phi \leq \psi$ iff $\langle \mathcal{W}, V\rangle \Vdash \phi \leq \psi$ for all valuations V on $\mathcal{W}$.
- For quasi-inequalities, $\mathcal{M} \Vdash \phi_1 \leq \psi_1 \,\&\ldots\&\, \phi_n \leq \psi_n \ \Rightarrow\ \phi \leq \psi$ iff $\mathcal{M} \Vdash \phi \leq \psi$ holds whenever $\mathcal{M} \Vdash \phi_i \leq \psi_i$ for all $i = 1, \ldots, n$, and $\mathcal{W} \Vdash \phi_1 \leq \psi_1 \,\&\ldots\&\, \phi_n \leq \psi_n \ \Rightarrow\ \phi \leq \psi$ iff $\langle \mathcal{W}, V\rangle \Vdash \phi_1 \leq \psi_1 \,\&\ldots\&\, \phi_n \leq \psi_n \ \Rightarrow\ \phi \leq \psi$ for all valuations V on $\mathcal{W}$.

Since $\mathsf{J}(\mathcal{W}), \mathsf{M}(\mathcal{W}) \subseteq \mathsf{Slop}(\mathcal{W})$, we have the following lemma:

Lemma 25. *For any $\mathcal{L}_{\otimes}^{+}$-formula ϕ, the function $\lambda w.V(\phi, w) : W \to [0,1]$ is a sloping function.*

3.3. First-order correspondence language $\mathcal{L}^{\mathsf{FO}}_{\otimes}$

In the first-order correspondence language $\mathcal{L}^{\mathsf{FO}}_{\otimes}$, we have:

- Logical symbols:
  - A set of individual variables $\mathsf{Var} = \{x_1, x_2, \ldots\}$.
  - Logical connectives $\vee, \wedge, \otimes, \rightarrow$.
  - First-order quantifiers $\forall, \exists$.
  - Equality symbol $=$.
  - Truth constants $\mathbf{t}$ for each $t \in \{0, 1\}$.
  - A binary connective $\sqsubseteq$.[4]
- Non-logical symbols:
  - A set of unary predicate symbols $\mathsf{Pred} = \{P_1, P_2, \ldots\}$, each of which corresponds to a propositional variable with the same subscript.
  - Individual symbols $\mathbf{c_i}$ and $\mathbf{c_m}$, each of which corresponds to a nominal in Nom or a conominal in Cnom.
  - Truth value symbols $\mathbf{C_m}$, $\mathbf{C_i}$, each of which corresponds to a nominal in Nom or a conominal in Cnom.
  - A binary relation symbol $\preceq$ corresponding to the partial order.[5]

We use $s, s', \ldots$ to denote an individual variable in Var or $\mathbf{c_i}$ for some $\mathbf{i} \in \mathsf{Nom}$ or $\mathbf{c_m}$ for some $\mathbf{m} \in \mathsf{Cnom}$. Atomic formulas of $\mathcal{L}^{\mathsf{FO}}_{\otimes}$ are defined as follows:

$$\beta ::= \mathbf{t} \mid s = s' \mid s \preceq s' \mid P_i s \mid \mathbf{C_i} \mid \mathbf{C_m}$$

where $t \in \{0, 1\}$, $P_i \in \mathsf{Pred}$, s, s' are as described above, $\mathbf{i} \in \mathsf{Nom}$ and $\mathbf{m} \in \mathsf{Cnom}$. Formulas of $\mathcal{L}^{\mathsf{FO}}_{\otimes}$ are defined as follows:

$$\alpha ::= \beta \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \mid \alpha \otimes \alpha \mid \alpha \rightarrow \alpha \mid \alpha \sqsubseteq \alpha \mid \forall x_i \alpha \mid \exists x_i \alpha \mid \forall \mathbf{c_i} \alpha \mid \forall \mathbf{c_m} \alpha \mid \forall \mathbf{C_i} \alpha \mid \forall \mathbf{C_m} \alpha$$

where β is an atomic formula, $x_i \in \mathsf{Var}$, $\mathbf{i} \in \mathsf{Nom}$ and $\mathbf{m} \in \mathsf{Cnom}$. We use $\mathsf{Form}(\mathcal{L}^{\mathsf{FO}}_{\otimes})$ to denote the set of all $\mathcal{L}^{\mathsf{FO}}_{\otimes}$-formulas. We also consider the second-order correspondence language $\mathcal{L}^{\mathsf{SO}}_{\otimes}$, which further allows second-order quantifiers $\forall P_n$ and $\exists P_n$, and for the non-atomic formulas, we have the following additional clause:

- If α is an $\mathcal{L}^{\mathsf{SO}}_{\otimes}$-formula and $P_n \in \mathsf{Pred}$, then $\forall P_n \alpha$ and $\exists P_n \alpha$ are also $\mathcal{L}^{\mathsf{SO}}_{\otimes}$-formulas.

Definition 26 (Interpretation of $\mathcal{L}^{\mathsf{FO}}_{\otimes}$)**.** An *interpretation* I of $\mathcal{L}^{\mathsf{FO}}_{\otimes}$ consists of the follows:

1. A non-empty set W^I, called the domain of I.
2. For each unary predicate symbol $P_n \in \mathsf{Pred}$, a sloping function $P_n^I \in \mathsf{Slop}(W^I)$.
3. For the binary relation symbol $\preceq$, a partial order $\preceq^I$ on W^I.
4. For the individual symbols $\mathbf{c_i}$, $\mathbf{c_m}$ where $\mathbf{i} \in \mathsf{Nom}$ and $\mathbf{m} \in \mathsf{Cnom}$, elements $c^I_{\mathbf{i}} \in W^I$ and $c^I_{\mathbf{m}} \in W^I$, respectively.
5. For the truth value symbols $\mathbf{C_i}$, $\mathbf{C_m}$ where $\mathbf{i} \in \mathsf{Nom}$ and $\mathbf{m} \in \mathsf{Cnom}$, truth values $C^I_i, C^I_m \in [0, 1]$, respectively.

Two interpretations I and I' are $\mathbf{c_i}$-*variants* of each other (notation: $I \sim_{\mathbf{c_i}} I'$) if I and I' differ at most at $\mathbf{c_i}$. The notion of $\mathbf{c_m}$-, P_n-variants are defined in a similar way, using the notations $\sim_{\mathbf{c_m}}$, $\sim_{P_n}$, respectively.

[4] One can understand $\sqsubseteq$ as value inequality; see [1] for further information and explanation.

[5] This $\preceq$ is the position inequality within the partial order. Again, for further intuition on these matters, see [1].

Definition 27 (Assignment)**.** Given an interpretation I of $\mathcal{L}_{\otimes}^{\mathsf{FO}}$, an *assignment* on I is a function $\sigma : \mathsf{Var} \to W^I$. Two assignments σ, σ' on I are x_n*-variants* of each other (notation: $\sigma \sim_{x_n} \sigma'$) if σ and σ' differ at most at $x_n \in \mathsf{Var}$.

Definition 28 (Valuation function for $\mathcal{L}_{\otimes}^{\mathsf{FO}}$-formulas)**.** Given any interpretation I of $\mathcal{L}_{\otimes}^{\mathsf{FO}}$ and any assignment σ on I, we use $v(x_i)$, $v(\mathbf{c_i})$ and $v(\mathbf{c_m})$ to denote $\sigma(x_i)$, $c_{\mathbf{i}}^I$, $c_{\mathbf{m}}^I$, respectively. The valuation function for $\mathcal{L}_{\otimes}^{\mathsf{FO}}$-formulas is a function $v^{I,\sigma} : \mathsf{Form}(\mathcal{L}_{\otimes}^{\mathsf{FO}}) \to [0,1]$ defined as follows:

- $v^{I,\sigma}(\mathbf{t}) = t$.
- $v^{I,\sigma}(s_i = s_j) = \begin{cases} 1 & \text{if } v(s_i) = v(s_j), \\ 0 & \text{otherwise.} \end{cases}$
- $v^{I,\sigma}(s_i \preceq s_j) = \begin{cases} 1 & \text{if } v(s_i) \preceq^I v(s_j), \\ 0 & \text{otherwise.} \end{cases}$
- $v^{I,\sigma}(P_i(s_j)) = P_i^I(v(s_j))$.
- $v^{I,\sigma}(\mathbf{C_i}) = C_{\mathbf{i}}^I$.
- $v^{I,\sigma}(\mathbf{C_m}) = C_{\mathbf{m}}^I$.
- $v^{I,\sigma}(\alpha \star \beta) = v^{I,\sigma}(\alpha) \star v^{I,\sigma}(\beta)$, where $\star \in \{\wedge, \vee, \otimes, \to\}$.
- $v^{I,\sigma}(\alpha \sqsubseteq \beta) = \begin{cases} 1 & \text{if } v^{I,\sigma}(\alpha) \leq v^{I,\sigma}(\beta), \\ 0 & \text{otherwise.} \end{cases}$
- $v^{I,\sigma}(\forall s\alpha) = \bigwedge\{v'(\alpha) : v' \sim_s v^{I,\sigma}\}$.
- $v^{I,\sigma}(\exists s\alpha) = \bigvee\{v'(\alpha) : v' \sim_s v^{I,\sigma}\}$.

For $\mathcal{L}_{\otimes}^{\mathsf{SO}}$-formulas, we have the following additional clauses:

- $v^{I,\sigma}(\forall P_n\alpha) = \bigwedge\{v'(\alpha) : v' \sim_{P_n} v^{I,\sigma}\}$.
- $v^{I,\sigma}(\exists P_n\alpha) = \bigvee\{v'(\alpha) : v' \sim_{P_n} v^{I,\sigma}\}$.

3.4. Standard translation

We define the standard translation of the expanded propositional language into the first-order correspondence language. Here $s \prec s'$ abbreviates $(s \preceq s') \wedge ((s = s') \to \bot)$.

Definition 29 (Standard translation)**.** The standard translation is defined as follows:

1. $ST_x(p_n) := \mathbf{P}_n(x)$.
2. $ST_x(\bot) := \bot$.
3. $ST_x(\top) := \top$.
4. $ST_x(\mathbf{i}) := (\mathbf{c_i} \prec x) \vee (x = \mathbf{c_i} \wedge \mathbf{C_i})$.
5. $ST_x(\mathbf{m}) := ((x \preceq \mathbf{c_m}) \to \bot) \vee (x = \mathbf{c_m} \wedge \mathbf{C_m})$.
6. $ST_x(\phi \star \psi) := ST_x(\phi) \star ST_x(\psi)$ for $\star \in \{\wedge, \vee, \otimes\}$.
7. $ST_x(\phi \to \psi) := (ST_x(\phi) \to ST_x(\psi)) \wedge \forall y((x \prec y) \to (ST_y(\phi) \sqsubseteq ST_y(\psi)))$.

For inequalities and quasi-inequalities, the standard translation is defined globally:

1. $ST(\phi \leq \psi) := \forall x(ST_x(\phi) \sqsubseteq ST_x(\psi))$.
2. $ST(\phi_1 \leq \psi_1 \,\&\ldots\&\, \phi_n \leq \psi_n \;\Rightarrow\; \phi \leq \psi) := (ST(\phi_1 \leq \psi_1) \wedge \ldots \wedge ST(\phi_n \leq \psi_n)) \sqsubseteq ST(\phi \leq \psi)$.

Definition 30 (Interpretation)**.** Given any BM-structure $\mathcal{M} = \langle \mathcal{W}, V \rangle$ where $\mathcal{W} = \langle W, \preceq \rangle$ is a poset, we define its associated interpretation $I^{\mathcal{M}}$ such that

- $W^{I^{\mathcal{M}}} = W$.
- $P_n^{I^{\mathcal{M}}} = \lambda w.V(p_n, w)$ for all propositional variables p_n.
- $\preceq^{I^{\mathcal{M}}} = \preceq$.
- $\lambda w.V(\mathbf{i}, w) = f_{c_{\mathbf{i}}^{I^{\mathcal{M}}}, C_{\mathbf{i}}^{I^{\mathcal{M}}}}$ for all nominals $\mathbf{i}$.
- $\lambda w.V(\mathbf{m}, w) = f_{c_{\mathbf{m}}^{I^{\mathcal{M}}}, C_{\mathbf{m}}^{I^{\mathcal{M}}}}$ for all conominals $\mathbf{m}$.

Definition 31 (Truth of predicate formulas)**.** Given any poset $\mathcal{W} = \langle W, \preceq\rangle$ and any $\mathcal{L}_{\otimes}^{\mathsf{SO}}$-formula α, we say that α is true on $\mathcal{W}$ (notation: $\mathcal{W} \models \alpha$), if for any valuation V on $\mathcal{W}$ and any assignment σ, $v^{I^{\langle \mathcal{W}, V\rangle}, \sigma}(\alpha) = 1$.

Proposition 32 (Soundness of translation)**.** *Given any BM-structure $\mathcal{M} = \langle W, \preceq, V\rangle$ and its associated interpretation $I^{\mathcal{M}}$, any $w \in W$, any $\mathcal{L}_{\otimes}^{+}$-formula ϕ, any $\mathcal{L}_{\otimes}^{+}$-inequality $\phi \leq \psi$, any $\mathcal{L}_{\otimes}^{+}$-quasi-inequality* Quasi*, any assignment σ such that $\sigma(x) = w$, we have*

- $V(\phi, w) = v^{I^{\mathcal{M}}, \sigma}(ST_x(\phi))$.
- $\mathcal{M} \Vdash \phi$ *iff* $v^{I^{\mathcal{M}}, \sigma}(\forall x ST_x(\phi)) = 1$.
- $\mathcal{M} \Vdash \phi \leq \psi$ *iff* $v^{I^{\mathcal{M}}, \sigma}(ST(\phi \leq \psi)) = 1$.
- $\mathcal{M} \Vdash \phi_1 \leq \psi_1 \,\&\ldots\&\, \phi_n \leq \psi_n \;\Rightarrow\; \phi \leq \psi$ *iff* $v^{I^{\mathcal{M}}, \sigma}(ST(\phi_1 \leq \psi_1 \,\&\ldots\&\, \phi_n \leq \psi_n \;\Rightarrow\; \phi \leq \psi)) = 1$.
- $\mathcal{W} \Vdash \phi$ *iff* $\mathcal{W} \models \forall \overline{P} \forall \overline{c}_{\mathbf{i}} \forall \overline{C}_{\mathbf{i}} \forall \overline{c}_{\mathbf{m}} \forall \overline{C}_{\mathbf{m}} \forall x ST_x(\phi)$*, where $\overline{p}$, $\overline{\mathbf{i}}$, $\overline{\mathbf{m}}$ are all propositional variables, nominals and conominals occurring in ϕ, respectively.*
- $\mathcal{W} \Vdash \phi \leq \psi$ *iff* $\mathcal{W} \models \forall \overline{P} \forall \overline{c}_{\mathbf{i}} \forall \overline{C}_{\mathbf{i}} \forall \overline{c}_{\mathbf{m}} \forall \overline{C}_{\mathbf{m}} ST(\phi \leq \psi)$*, where $\overline{p}$, $\overline{\mathbf{i}}$, $\overline{\mathbf{m}}$ are all propositional variables, nominals and conominals occurring in $\phi \leq \psi$, respectively.*
- $\mathcal{W} \Vdash$ Quasi *iff* $\mathcal{W} \models \forall \overline{P} \forall \overline{c}_{\mathbf{i}} \forall \overline{C}_{\mathbf{i}} \forall \overline{c}_{\mathbf{m}} \forall \overline{C}_{\mathbf{m}} ST(\mathsf{Quasi})$*, where $\overline{p}$, $\overline{\mathbf{i}}$, $\overline{\mathbf{m}}$ are all propositional variables, nominals and conominals occurring in* Quasi*, respectively.*

Proof. First item is by induction on complexity of formulas, the rest follows immediately. □

4. Inductive formulas

We now define inductive formulas in $\mathbf{GBL}_{ewf}$. Our presentation follows [14, 19].

4.1. Inductive Formulas for $\mathbf{GBL}_{ewf}$

Herein, we will define inductive formulas for $\mathbf{GBL}_{ewf}$.

Definition 33 (Order-type of propositional variables, page 346 in [14])**.** Fix a sequence $\overline{p} = (p_1, \ldots, p_n)$ of propositional variables.

- An *order-type* ϵ of $\overline{p}$ is an element in $\{+, -\}^n$.
- p_i has order-type $+$ if the i-th coordinate of ϵ is $+$ (notation: $\epsilon(p_i) = +$).
- p_i has order-type $-$ if the i-th coordinate of ϵ is $-$ (notation: $\epsilon(p_i) = -$).
- $\epsilon^{\partial} \in \{+, -\}^n$ is the *opposite order-type* of ϵ, i.e. the i-th coordinate of ϵ^{∂} is $+$ iff the i-th coordinate of ϵ is $-$.

Inductive formulas and inequalities requires generation trees:

Definition 34 (Generation tree)**.** The generation trees of formulas are inductively defined as follows:

- The generation tree of a propositional variable p or $\perp$ is a node labelled with the formula itself.
- The generation tree of a formula $\phi \heartsuit \psi$ (where $\heartsuit \in \{\vee, \wedge, \otimes, \to\}$) is defined as a father node labelled with $\heartsuit$ together with two subtrees, where the generation tree of ϕ is the left subtree, and the generation tree of ψ is the right subtree.

outer				inner		
$+$	$\vee$	$\wedge$	$\otimes$	$+$	$\wedge$	$\rightarrow$
$-$	$\wedge$	$\vee$	$\rightarrow$	$-$	$\vee$	$\otimes$

Table 1
Inner and outer nodes

The next definition concerns signed generation trees which are integral to the definition of inductive inequalities defined later:

Definition 35 (Signed generation tree, cf. Definition 2.4 in [19])**.** The *positive/negative generation tree* of any given formula ϕ is defined by first assigning the root of the generation tree of ϕ with $+/-$ and then:

- Assign the same sign to the children nodes of any node labelled with $\vee, \wedge, \otimes$.
- Assign the opposite sign to the first child node and the same sign to the second child node of any node labelled with $\rightarrow$.

Nodes in signed generation trees are *positive/negative* if they are signed $+/-$. We will also use $+\phi$ and $-\phi$ to denote the positive and negative generation trees, respectively. Given $*, \star \in \{+, -\}$, we use $\star\psi \prec *\phi$ to indicate that in the signed generation tree $*\phi$, the root of the subtree of a subformula ψ has the sign $\star$. In particular, we say that an occurrence of p in ϕ is *positive/negative* if for this occurrence, $\star p \prec +\phi$ for $\star = +/-$.

We will use the positive generation tree $+\phi$ and the negative generation tree $-\psi$ in the inequality $\phi \leq \psi$.

Definition 36 (Uniformity, page 8 in [19])**.**

- $\phi \leq \psi$ is *uniform* in p_i if all occurrences of p_i in $+\phi$ and $-\psi$ have the same sign.
- $\phi \leq \psi$ is *uniform* in $\overline{p} = (p_1, \ldots, p_n)$ if $\phi \leq \psi$ is *uniform* in each p_i for $i = 1, \ldots, n$.
- $\phi \leq \psi$ is ϵ-*uniform* in $\overline{p} = (p_1, \ldots, p_n)$ if $\phi \leq \psi$ is uniform in $\overline{p}$ and for $i = 1, \ldots, n$, p_i has the sign $\epsilon(p_i)$.

Definition 37 (ϵ-criticality, page 8 in [19])**.** Fix a formula $\phi(p_1, \ldots p_n)$, an order-type ϵ of $(p_1, \ldots p_n)$, $i \in \{1, \ldots, n\}$ and $* \in \{+, -\}$.

- An ϵ-*critical node* in the signed generation tree $*\phi$ is a leaf node $+p_i$ if $\epsilon(p_i) = +$ and $-p_i$ if $\epsilon(p_i) = -$.
- An ϵ-*critical branch* in a signed generation tree is a branch from an ϵ-critical node. These branches will be used to compute the minimal valuation in the correspondence algorithm.
- $*\phi$ *agrees with* ϵ (notation: $\epsilon(*\phi)$[6]), if every leaf node in $*\phi$ is ϵ-critical.
- We write $\epsilon(\gamma) \prec *\phi$ to indicate that the signed generation subtree of γ in the signed generation tree of $*\phi$ agrees with ϵ.

Defining inductive formulas, we use this classification table:

Definition 38 (Classification table, cf. Definition 2.5 in [19])**.** Nodes in signed generation trees are called inner and outer nodes, according to Table 1. In this definition, $+\vee, +\wedge, +\otimes, -\wedge, -\vee, -\rightarrow$ are classified as outer nodes, and $+\wedge, +\rightarrow, -\vee, -\otimes$ are classified as inner nodes. Notice that $+\wedge, -\vee$ are classified both as outer nodes and as inner nodes. This definition is more like a qualification definition, in the sense that these labelled connectives are qualified as outer nodes or inner nodes or both.

[6]Notice that there is an abuse of notation that we both use $\epsilon(p_i)$ and $\epsilon(*\phi)$, but it should be clear from the context which one is used, since the former is either $+$ or $-$, and the latter is a statement about a fact.

Definition 39 (Good branches)**.** A branch in a signed generation tree is *good* if it is the concatenation of two paths P_1 and P_2, one of which might be of length 0, such that P_1 is a path from the leaf consisting (apart from variable nodes) of inner nodes only, and P_2 consists (apart from variable nodes) of outer nodes only.

Now we can define the inductive inequalities:

Definition 40 (Inductive inequalities, cf. Definition 2.6 in [19])**.** For any order-type ϵ and any irreflexive and transitive binary relation $<_\Omega$ (which is called the dependence order) on $p_1, \ldots p_n$, the signed generation tree $*\phi$ (where $* \in \{-, +\}$) of a formula $\phi(p_1, \ldots p_n)$ is (Ω, ϵ)-*inductive* if:

1. Every ϵ-critical branch with leaf p_i (where $1 \leq i \leq n$) is good.
2. Every inner node $+ \rightarrow, -\otimes$ in an ϵ-critical branch with corresponding subformula of the form $\star(\gamma, \beta)$ or $\star(\beta, \gamma)$ is such that:
   a) $\star$ is a binary connective.
   b) The ϵ-critical branch goes through β with leaf node p_i.
   c) $\epsilon^\partial(\gamma) \prec *\phi$.
   d) $p_k <_\Omega p_i$ for every p_k that occurs in γ.

An inequality $\phi \leq \psi$ is (Ω, ϵ)-*inductive* if both $+\phi$ and $-\psi$ are (Ω, ϵ)-inductive. An inequality $\phi \leq \psi$ is *inductive* if it is (Ω, ϵ)-inductive for some (Ω, ϵ). A formula $\phi \rightarrow \psi$ is *inductive* if $\phi \leq \psi$ is inductive.

5. The algorithm ALBA for $\mathbf{GBL}_{ewf}$

In the present section, we will give the algorithm ALBA for $\mathbf{GBL}_{ewf}$. This version here is similar to [15]. The algorithm receives an inductive formula $\phi \rightarrow \psi$ and transforms it into $\phi \leq \psi$. Then ALBA executes in three stages:

Stage 1: Preprocessing The algorithm applies the following rules exhaustively to $\phi \leq \psi$ with signed generation trees $+\phi$ and $-\psi$:

1. Distribution rules: For $+\phi$ and $-\psi$, we rewrite signed subformulas of the former form into signed subformulas of the latter form:
   a) $+\alpha \wedge (\beta \vee \gamma)$, $+(\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$.
   b) $+(\beta \vee \gamma) \wedge \alpha$, $+(\beta \wedge \alpha) \vee (\gamma \wedge \alpha)$.
   c) $+\alpha \otimes (\beta \vee \gamma)$, $+(\alpha \otimes \beta) \vee (\alpha \otimes \gamma)$.
   d) $+(\beta \vee \gamma) \otimes \alpha$, $+(\beta \otimes \alpha) \vee (\gamma \otimes \alpha)$.
   e) $-\alpha \vee (\beta \wedge \gamma)$, $-(\alpha \vee \beta) \wedge (\alpha \vee \gamma)$.
   f) $-(\beta \wedge \gamma) \vee \alpha$, $-(\beta \vee \alpha) \wedge (\gamma \vee \alpha)$.
   g) $-(\beta \vee \gamma) \rightarrow \alpha$, $-(\beta \rightarrow \alpha) \wedge (\gamma \rightarrow \alpha)$.
   h) $-\alpha \rightarrow (\beta \wedge \gamma)$, $-(\alpha \rightarrow \beta) \wedge (\alpha \rightarrow \gamma)$.
2. Splitting rules:

$$\frac{\alpha \leq \beta \wedge \gamma}{\alpha \leq \beta \quad \alpha \leq \gamma}$$

$$\frac{\alpha \vee \beta \leq \gamma}{\alpha \leq \gamma \quad \beta \leq \gamma}$$

3. Monotone and antitone variable-elimination rules:

$$\frac{\alpha(p) \leq \beta(p)}{\alpha(\bot) \leq \beta(\bot)}$$

$$\frac{\beta(p) \leq \alpha(p)}{\beta(\top) \leq \alpha(\top)}$$

where $\beta(p)$ and $\alpha(p)$ are positive and negative in p, respectively.

After applying the distribution rules, the splitting rules and the monotone and antitone variable elimination rules exhaustively, the input inequality $\phi \leq \psi$ is transformed into a set of inequalities $\{\phi'_i \leq \psi'_i \mid 1 \leq i \leq n\}$. Then each inequality is rewritten as an *initial quasi-inequality* $\& S_i \Rightarrow \mathsf{Ineq_i}$, abbreviated as $(S_i, \mathsf{Ineq_i})$ called *systems*, where $S_i = \emptyset$ and $\mathsf{Ineq_i}$ is $\phi'_i \leq \psi'_i$.

Stage 2: Reduction and elimination This stage aims at eliminating all propositional variables from a system (S, Ineq) by the following *reduction rules*, where the approximation rules and the Ackermann rules operate on the whole system and the residuation rules and the splitting rules operate on a single inequality in S.

1. Approximation rules: We have the following notational conventions:
   - $\phi(!x)$ indicates that x occurs only once in ϕ.
   - $\phi(\gamma/!x)$ is obtained from ϕ by substituting γ for x.
   - The branch of $*\phi(!x)$ starting at x means the path from x to the root in the signed generation tree $*\phi(!x)$, where $* \in \{+, -\}$.
   - γ represents a formula in the basic language $\mathcal{L}_\otimes$.
   - $\mathbf{i}$ represents a nominal variable not occurring in the system before the application of the rule.

   a) Left-positive approximation rule:
   $$\frac{(S,\ \ \phi'(\gamma/!x) \leq \psi')}{(S \cup \{\mathbf{i} \leq \gamma\},\ \ \phi'(\mathbf{i}/!x) \leq \psi')}$$
   where $+x \prec +\phi'(!x)$, the branch of $+\phi'(!x)$ starting at $+x$ consists of $+\wedge, +\otimes, -\vee, -\rightarrow$.[7]

   b) Left-negative approximation rule:
   $$\frac{(S,\ \ \phi'(\gamma/!x) \leq \psi')}{(S \cup \{\gamma \leq \mathbf{m}\},\ \ \phi'(\mathbf{m}/!x) \leq \psi')}$$
   where $-x \prec +\phi'(!x)$, the branch of $+\phi'(!x)$ starting at $-x$ consists of $+\wedge, +\otimes, -\vee, -\rightarrow$.

   c) Right-positive approximation rule:
   $$\frac{(S,\ \ \phi' \leq \psi'(\gamma/!x))}{(S \cup \{\mathbf{i} \leq \gamma\},\ \ \phi' \leq \psi'(\mathbf{i}/!x))}$$
   where $+x \prec -\psi'(!x)$, the branch of $-\psi'(!x)$ starting at $+x$ consists of $+\wedge, +\otimes, -\vee, -\rightarrow$.

   d) Right-negative approximation rule:
   $$\frac{(S,\ \ \phi' \leq \psi'(\gamma/!x))}{(S \cup \{\gamma \leq \mathbf{m}\},\ \ \phi' \leq \psi'(\mathbf{m}/!x))}$$
   where $-x \prec -\psi'(!x)$, the branch of $-\psi'(!x)$ starting at $-x$ consists of $+\wedge, +\otimes, -\vee, -\rightarrow$.

[7] In the success proofs, we will apply the approximation rules *pivotally* to nodes $!x$ in the sense that the branch starting at x is a *maximal* branch of outer nodes, i.e. if we extend the branch longer, then it will contain non-outer nodes. If a run of the algorithm only applies the approximation rules pivotally, then we say that the run is *pivotal*.

2. Residuation rules:

$$\frac{\gamma \otimes \delta \leq \beta}{\gamma \leq \delta \to \beta} \qquad \frac{\gamma \otimes \delta \leq \beta}{\delta \leq \gamma \to \beta} \qquad \frac{\gamma \leq \delta \to \beta}{\gamma \otimes \delta \leq \beta} \qquad \frac{\gamma \leq \delta \to \beta}{\delta \leq \gamma \to \beta}$$

3. Splitting rules:

$$\frac{\alpha \leq \beta \wedge \gamma}{\alpha \leq \beta \quad \alpha \leq \gamma} \qquad \frac{\alpha \vee \beta \leq \gamma}{\alpha \leq \gamma \quad \beta \leq \gamma}$$

4. Right Ackermann-rule:

$$\frac{(\{\beta_j(p) \leq \gamma_j(p) \mid 1 \leq j \leq n\} \cup \{\alpha_i \leq p \mid 1 \leq i \leq m\}, \ \ \mathsf{Ineq})}{(\{\beta_j(\bigvee_{i=1}^m \alpha_i) \leq \gamma_j(\bigvee_{i=1}^m \alpha_i) \mid 1 \leq j \leq n\}, \ \ \mathsf{Ineq})}$$

where:
- $\beta_j(p)$ (resp. $\gamma_j(p)$) are positive (resp. negative) in p for $1 \leq j \leq n$,
- p does not occur in Ineq or α_i for $1 \leq i \leq m$.

5. Left Ackermann-rule:

$$\frac{(\{\beta_j(p) \leq \gamma_j(p) \mid 1 \leq j \leq n\} \cup \{p \leq \alpha_i \mid 1 \leq i \leq m\}, \ \ \mathsf{Ineq})}{(\{\beta_j(\bigwedge_{i=1}^m \alpha_i) \leq \gamma_j(\bigwedge_{i=1}^m \alpha_i) \mid 1 \leq j \leq n\}, \ \ \mathsf{Ineq})}$$

where:
- $\beta_j(p)$ (resp. $\gamma_j(p)$) are negative (resp. positive) in p for $1 \leq j \leq n$,
- p does not occur in Ineq or α_i for $1 \leq i \leq m$.

Stage 3: Output If in every system produced in Stage 1, all propositional variables are eliminated, then for each i, the system becomes (S_i, Ineq_i). The algorithm outputs $\bigwedge_i \forall \overline{P} \forall \overline{c}_{\mathbf{i}} \forall \overline{C}_{\mathbf{i}} \forall \overline{c}_{\mathbf{m}} \forall \overline{C}_{\mathbf{m}} ST(\& S_i \ \Rightarrow \mathsf{Ineq}_i)$ (where $\overline{p}$, $\overline{\mathbf{i}}$, $\overline{\mathbf{m}}$ are all propositional variables, nominals and conominals occurring in $\& S_i \ \Rightarrow \mathsf{Ineq}_i$, respectively), which is denoted by $\mathsf{FO}(\phi \leq \psi)$. Otherwise, the algorithm halts and reports failure.

6. Soundness

Theorem 41 (Soundness). *If* ALBA *runs successfully on* $\phi \to \psi$ *and outputs* $\mathsf{FO}(\phi \leq \psi)$*, then for any poset* $\mathcal{W} = \langle W, \preceq \rangle$*,*

$$\mathcal{W} \Vdash \phi \to \psi \textit{ iff } \mathcal{W} \models \mathsf{FO}(\phi \leq \psi).$$

Proof. The proof is similar to [15, Theorem 6.1]. We direct the interested reader to [18] where the proof is given in full.

□

7. Success

Finally, ALBA succeeds: for inductive inequalities, there is a run where the algorithm terminates, returning the first-order correspondent. The proof structure is similar to [15, Section 8].

Definition 42 (Definite inductive inequality, cf. Definition 8.1 in [15])**.** An (Ω, ϵ)-inductive inequality is *definite* if all outer nodes on critical branches are neither $+\vee$ nor $-\wedge$.

Lemma 43. *Let $\{\phi_i \leq \psi_i\}$ be the set of inequalities obtained after applying the distribution rules, the splitting rules and the monotone and antitone variable elimination rules exhaustively in Stage 1 to an (Ω, ϵ)-inductive inequality $\phi \leq \psi$. Then each $\phi_i \leq \psi_i$ is a definite (Ω, ϵ)-inductive inequality.*

Proof. The proof is similar to [15, Lemma 8.2]. The details are in [18]. □

Definition 44 (Stripped system, cf. Definition 8.3 in [15])**.** A system (S, Ineq) is (Ω, ϵ)-*stripped* if Ineq is pure and for each $\xi \leq \chi \in S$, the following conditions hold:

1. one of $+\chi$ and $-\xi$ is pure, and the other is (Ω, ϵ)-inductive;
2. every node in an ϵ-critical branch in $+\chi$ and $-\xi$ is an inner node.

Lemma 45. *For any definite (Ω, ϵ)-inductive inequality $\phi \leq \psi$, the system $(\emptyset, \phi \leq \psi)$ can be rewritten into an (Ω, ϵ)-stripped system by exhaustive applications of approximation rules.*

Proof. The proof is similar to [15, Lemma 8.4], and is given in full in [18]. □

Definition 46 (Ackermann-ready system, cf. Definition 8.5 in [15])**.** An (Ω, ϵ)-stripped system (S, Ineq) is *Ackermann-ready* with respect to p_i with $\epsilon_i = 1$ (resp. $\epsilon_i = \partial$) if every $\xi \leq \chi \in S$ has one of the following forms:

1. $\xi \leq p_i$ with ξ pure (resp. $p_i \leq \chi$ with χ pure).
2. $\xi \leq \chi$ where $-\xi$ and $+\chi$ are ϵ^∂-uniform in p_i.

Lemma 47 (cf. Lemma 8.6 in [15])**.** *If (S, Ineq) is (Ω, ϵ)-stripped and p_i is Ω-minimal among the remaining propositional variables occurring in (S, Ineq), then (S, Ineq) can be rewritten by the application of residuation and splitting rules into a system which is Ackermann-ready with respect to p_i.*

Proof. We refer the interested reader to the full proof given in [18]. □

Lemma 48 (cf. Lemma 8.7 in [15])**.** *After applying the Ackermann rule to an (Ω, ϵ)-stripped system which is Ackermann-ready with respect to p_i, a system is obtained which is again (Ω, ϵ)-stripped.*

Proof. Given in full in [18]. □

So, all propositional variables can be eliminated from an (Ω, ϵ)-stripped system upon repeatedly applying Ackermann rules. Thus:

Theorem 49. ALBA *pivotally succeeds on all inductive inequalities.*

Proof. The proof is similar to [15, Theorem 8.8], and given in full in [18]. □

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

Acknowledgments

The research of Zhiguang Zhao is supported by Shandong Provincial Natural Science Foundation, China (project number: ZR2023QF021).

References

[1] C. Britz, W. Conradie, W. Morton, Correspondence theory for many-valued modal logic, 2025. URL: https://arxiv.org/abs/2401.07894. arXiv:2401.07894.

[2] A. Lewis-Smith, P. Oliva, E. Robinson, Kripke semantics for intuitionistic Łukasiewicz logic, Studia Logica 109 (2021) 313–339.

[3] S. Bova, F. Montagna, The consequence relation in the logic of commutative GBL-algebras is PSPACE-complete, Theoretical Computer Science 410 (2009) 1143–1158. doi:10.1016/j.tcs.2008.10.024.

[4] P. Jipsen, F. Montagna, On the structure of generalized BL-algebras, Algebra Universalis 55 (2006) 227–238.

[5] P. Jipsen, F. Montagna, Embedding theorems for classes of GBL-algebras, Journal of Pure and Applied Algebra 214 (2010) 1559–1575.

[6] A. Lewis-Smith, A Kripke Semantics for Hajek's BL, arXiv e-prints (2023) arXiv–2308.

[7] A. Lewis-Smith, Z. Zhao, A kripke semantics for Intuitionistic łukasiewicz Logic with Weak Excluded Middle, in: European Conference on Logics in Artificial Intelligence, Springer, 2025, pp. 275–289.

[8] A. Lewis-Smith, Z. Zhao, A Kripke Semantics for Monadic BL Chains, in: European Conference on Symbolic and Quantitative Approaches with Uncertainty, Springer, 2025, pp. 499–511.

[9] A. Lewis-Smith, Z. Zhao, A Kripke Semantics for commutative Generalised Basic Logic, in: 8th Asian Workshop on Philosophical Logic, 80qz2, 2026, pp. 1–12.

[10] J. Lukasiewicz, Selected Works (1970).

[11] H. Sahlqvist, Completeness and correspondence in the first and second order semantics for modal logic, in: Studies in Logic and the Foundations of Mathematics, volume 82, North Holland, 1975, pp. 110–143.

[12] J. van Benthem, Modal logic and classical logic, Bibliopolis, 1983.

[13] W. Conradie, S. Ghilardi, A. Palmigiano, Unified correspondence, in: A. Baltag, S. Smets (Eds.), Johan van Benthem on Logic and Information Dynamics, volume 5 of *Outstanding Contributions to Logic*, Springer International Publishing, 2014, pp. 933–975. doi:10.1007/978-3-319-06025-5_36.

[14] W. Conradie, A. Palmigiano, Algorithmic correspondence and canonicity for distributive modal logic, Annals of Pure and Applied Logic 163 (2012) 338 – 376.

[15] W. Conradie, A. Palmigiano, Algorithmic correspondence and canonicity for non-distributive logics, Annals of Pure and Applied Logic 170 (2019) 923 – 974. URL: http://www.sciencedirect.com/science/article/pii/S0168007219300314.

[16] M. C. Fitting, Many-valued modal logics, Fundamenta informaticae 15 (1991) 235–254.

[17] W. Fussner, Poset products as relational models, Studia Logica 110 (2021). doi:10.1007/s11225-021-09956-z.

[18] A. Lewis-Smith, Z. Zhao, Correspondence theory for intuitionistic Łukasiewicz logic (2026). doi:10.13140/RG.2.2.18933.13288.

[19] W. Conradie, A. Palmigiano, Z. Zhao, Sahlqvist via translation, Logical Methods in Computer Science Volume 15, Issue 1 (2019). doi:10.23638/LMCS-15(1:15)2019.