A variant of bi-intuitionistic logic

Heinrich Wansing
Ruhr-University Bochum
Department of Philosophy II
Heinrich.Wansing@rub.de

Maltsev Meeting, Novosibirsk, November 12, 2013

Trivializing and paraconsistent unary connectives

trivializing $*$	paraconsistent $*$
$A, *A \models B$	$A, *A \not\models B$
$A, *A \models^d B$	$A, *A \not\models^d B$

classical negation	parac. strong neg.
$A, \sim_c A \models B$	$A, \sim_s A \not\models B$
$A, \sim_c A \models^d B$	$A, \sim_s A \not\models^d B$

intuitionistic negation	co-negation
$A, \neg A \models B$	$A, -A \not\models B$
$A, \neg A \not\models^d B$	$A, -A \models^d B$

Trivializing and paraconsistent unary connectives

trivializing $*$	paraconsistent $*$
$A, *A \models B$	$A, *A \not\models B$
$A, *A \models^d B$	$A, *A \not\models^d B$

classical negation	parac. strong neg.
$A, \sim_c A \models B$	$A, \sim_s A \not\models B$
$A, \sim_c A \models^d B$	$A, \sim_s A \not\models^d B$

intuitionistic negation	co-negation
$A, \neg A \models B$	$A, -A \not\models B$
$A, \neg A \not\models^d B$	$A, -A \models^d B$

Trivializing and paraconsistent unary connectives

trivializing $*$	paraconsistent $*$
$A, *A \models B$	$A, *A \not\models B$
$A, *A \models^d B$	$A, *A \not\models^d B$

classical negation	parac. strong neg.
$A, \sim_c A \models B$	$A, \sim_s A \not\models B$
$A, \sim_c A \models^d B$	$A, \sim_s A \not\models^d B$

intuitionistic negation	co-negation
$A, \neg A \models B$	$A, -A \not\models B$
$A, \neg A \not\models^d B$	$A, -A \models^d B$

Trivializing and paraconsistent unary connectives

trivializing $*$	paraconsistent $*$
$A, *A \models B$	$A, *A \not\models B$
$A, *A \models^d B$	$A, *A \not\models^d B$

classical negation	parac. strong neg.
$A, \sim_c A \models B$	$A, \sim_s A \not\models B$
$A, \sim_c A \models^d B$	$A, \sim_s A \not\models^d B$

intuitionistic negation	co-negation
$A, \neg A \models B$	$A, -A \not\models B$
$A, \neg A \not\models^d B$	$A, -A \models^d B$

Trivializing and paraconsistent unary connectives

trivializing $*$	paraconsistent $*$
$A, *A \models B$	$A, *A \not\models B$
$A, *A \models^d B$	$A, *A \not\models^d B$

classical negation	parac. strong neg.
$A, \sim_c A \models B$	$A, \sim_s A \not\models B$
$A, \sim_c A \models^d B$	$A, \sim_s A \not\models^d B$

intuitionistic negation	co-negation
$A, \neg A \models B$	$A, -A \not\models B$
$A, \neg A \not\models^d B$	$A, -A \models^d B$

Trivializing and paraconsistent unary connectives

trivializing $*$	paraconsistent $*$
$A, *A \models B$	$A, *A \not\models B$
$A, *A \models^d B$	$A, *A \not\models^d B$

classical negation	parac. strong neg.
$A, \sim_c A \models B$	$A, \sim_s A \not\models B$
$A, \sim_c A \models^d B$	$A, \sim_s A \not\models^d B$

intuitionistic negation	co-negation
$A, \neg A \models B$	$A, -A \not\models B$
$A, \neg A \not\models^d B$	$A, -A \models^d B$

Trivializing and paraconsistent unary connectives

trivializing $*$	paraconsistent $*$
$A, *A \models B$	$A, *A \not\models B$
$A, *A \models^d B$	$A, *A \not\models^d B$

classical negation	parac. strong neg.
$A, \sim_c A \models B$	$A, \sim_s A \not\models B$
$A, \sim_c A \models^d B$	$A, \sim_s A \not\models^d B$

intuitionistic negation	co-negation
$A, \neg A \models B$	$A, -A \not\models B$
$A, \neg A \not\models^d B$	$A, -A \models^d B$

Trivializing and paraconsistent unary connectives

trivializing $*$	paraconsistent $*$
$A, *A \models B$	$A, *A \not\models B$
$A, *A \models^d B$	$A, *A \not\models^d B$

classical negation	parac. strong neg.
$A, \sim_c A \models B$	$A, \sim_s A \not\models B$
$A, \sim_c A \models^d B$	$A, \sim_s A \not\models^d B$

intuitionistic negation	co-negation
$A, \neg A \models B$	$A, -A \not\models B$
$A, \neg A \not\models^d B$	$A, -A \models^d B$

Trivializing and paraconsistent unary connectives

trivializing $*$	paraconsistent $*$
$A, *A \models B$	$A, *A \not\models B$
$A, *A \models^d B$	$A, *A \not\models^d B$

classical negation	parac. strong neg.
$A, \sim_c A \models B$	$A, \sim_s A \not\models B$
$A, \sim_c A \models^d B$	$A, \sim_s A \not\models^d B$

intuitionistic negation	co-negation
$A, \neg A \models B$	$A, -A \not\models B$
$A, \neg A \not\models^d B$	$A, -A \models^d B$

Trivializing and paraconsistent unary connectives

trivializing $*$	paraconsistent $*$
$A, *A \models B$	$A, *A \not\models B$
$A, *A \models^d B$	$A, *A \not\models^d B$

classical negation	parac. strong neg.
$A, \sim_c A \models B$	$A, \sim_s A \not\models B$
$A, \sim_c A \models^d B$	$A, \sim_s A \not\models^d B$

intuitionistic negation	co-negation
$A, \neg A \models B$	$A, -A \not\models B$
$A, \neg A \not\models^d B$	$A, -A \models^d B$

The rest of the talk is devoted to making sense of dual entailment.

The idea is to begin with natural deduction and to develop a notion of dual proof (falsification) in addition to the more familiar notion of proof (verification).

If we look at the falsification conditions of implications, it turns out that the standard conception requires reference to both the notion of falsification and the notion of verification:

A falsification of an implication $(A \to B)$ is a pair consisting of a verification of A and a falsification of B.

The rest of the talk is devoted to making sense of dual entailment.

The idea is to begin with natural deduction and to develop a notion of dual proof (falsification) in addition to the more familiar notion of proof (verification).

If we look at the falsification conditions of implications, it turns out that the standard conception requires reference to both the notion of falsification and the notion of verification:

A falsification of an implication ($A \rightarrow B$) is a pair consisting of a verification of A and a falsification of B.

The rest of the talk is devoted to making sense of dual entailment.

The idea is to begin with natural deduction and to develop a notion of dual proof (falsification) in addition to the more familiar notion of proof (verification).

If we look at the falsification conditions of implications, it turns out that the standard conception requires reference to both the notion of falsification and the notion of verification:

A falsification of an implication $(A \to B)$ is a pair consisting of a verification of A and a falsification of B.

The rest of the talk is devoted to making sense of dual entailment.

The idea is to begin with natural deduction and to develop a notion of dual proof (falsification) in addition to the more familiar notion of proof (verification).

If we look at the falsification conditions of implications, it turns out that the standard conception requires reference to both the notion of falsification and the notion of verification:

A falsification of an implication ($A \rightarrow B$) is a pair consisting of a verification of A and a falsification of B.

If the falsificationist has to specify falsification conditions for *implications*, then, it may be claimed, the verificationist must specify verification conditions for *co-implications*.

The role of implication in verificationism is dual to the role of co-implication in falsificationism. The connectives internalize the respective variety of reasoning into the object language. Whereas implication $\to$ is the residuum of additive conjunction, co-implication $\prec$ is the residuum of additive disjunction:

$$(A \wedge B) \vdash C \quad \text{iff} \quad A \vdash (B \to C) \quad \text{iff} \quad B \vdash (A \to C),$$

$$C \vdash (A \vee B) \quad \text{iff} \quad (C \prec A) \vdash B \quad \text{iff} \quad (C \prec B) \vdash A.$$

The formula $(C \prec B)$ is read as "B co-implies C" or as "C excludes B".

If the falsificationist has to specify falsification conditions for *implications*, then, it may be claimed, the verificationist must specify verification conditions for *co-implications*.

The role of implication in verificationism is dual to the role of co-implication in falsificationism. The connectives internalize the respective variety of reasoning into the object language. Whereas implication $\rightarrow$ is the residuum of additive conjunction, co-implication $\prec$ is the residuum of additive disjunction:

$$
\begin{array}{ccccc}
(A \wedge B) \vdash C & \text{iff} & A \vdash (B \rightarrow C) & \text{iff} & B \vdash (A \rightarrow C), \\
C \vdash (A \vee B) & \text{iff} & (C \prec A) \vdash B & \text{iff} & (C \prec B) \vdash A.
\end{array}
$$

The formula $(C \prec B)$ is read as "B co-implies C" or as "C excludes B".

If the falsificationist has to specify falsification conditions for *implications*, then, it may be claimed, the verificationist must specify verification conditions for *co-implications*.

The role of implication in verificationism is dual to the role of co-implication in falsificationism. The connectives internalize the respective variety of reasoning into the object language. Whereas implication $\to$ is the residuum of additive conjunction, co-implication $\prec$ is the residuum of additive disjunction:

$$
\begin{array}{ccccc}
(A \wedge B) \vdash C & \text{iff} & A \vdash (B \to C) & \text{iff} & B \vdash (A \to C), \\
C \vdash (A \vee B) & \text{iff} & (C \prec A) \vdash B & \text{iff} & (C \prec B) \vdash A.
\end{array}
$$

The formula $(C \prec B)$ is read as "B co-implies C" or as "C excludes B".

We will pursue the idea of combining the notions of verification and falsification in a single system of natural deduction.

One might expect that this move will in lead to a logic that is known as bi-intuitionistic logic **BiInt**.

It will not.

We will obtain another kind of a bi-intuitionistic propositional logic that combines verification and its dual. This logic, **2Int**, is motivated by a certain dualization of the natural deduction rules for intuitionistic propositional logic. It is shown that **2Int** can be faithfully embedded into intuitionistic logic **Int** wrt to validity and that it can be faithfully embedded into dual intuitionistic logic **DualInt** wrt to dual validity.

A certain subsystem of **BiInt** has recently been investigated under the name 'da Costa logic', **daC**. A natural deduction calculus for **daC** can be found in (Priest 2009).

We will pursue the idea of combining the notions of verification and falsification in a single system of natural deduction.

One might expect that this move will in lead to a logic that is known as bi-intuitionistic logic **BiInt**.

It will not.

We will obtain another kind of a bi-intuitionistic propositional logic that combines verification and its dual. This logic, **2Int**, is motivated by a certain dualization of the natural deduction rules for intuitionistic propositional logic. It is shown that **2Int** can be faithfully embedded into intuitionistic logic **Int** wrt to validity and that it can be faithfully embedded into dual intuitionistic logic **DualInt** wrt to dual validity.

A certain subsystem of **BiInt** has recently been investigated under the name 'da Costa logic', **daC**. A natural deduction calculus for **daC** can be found in (Priest 2009).

We will pursue the idea of combining the notions of verification and falsification in a single system of natural deduction.

One might expect that this move will in lead to a logic that is known as bi-intuitionistic logic **BiInt**.

It will not.

We will obtain another kind of a bi-intuitionistic propositional logic that combines verification and its dual. This logic, **2Int**, is motivated by a certain dualization of the natural deduction rules for intuitionistic propositional logic. It is shown that **2Int** can be faithfully embedded into intuitionistic logic **Int** wrt to validity and that it can be faithfully embedded into dual intuitionistic logic **DualInt** wrt to dual validity.

A certain subsystem of **BiInt** has recently been investigated under the name 'da Costa logic', **daC**. A natural deduction calculus for **daC** can be found in (Priest 2009).

We will pursue the idea of combining the notions of verification and falsification in a single system of natural deduction.

One might expect that this move will in lead to a logic that is known as bi-intuitionistic logic **BiInt**.

It will not.

We will obtain another kind of a bi-intuitionistic propositional logic that combines verification and its dual. This logic, **2Int**, is motivated by a certain dualization of the natural deduction rules for intuitionistic propositional logic. It is shown that **2Int** can be faithfully embedded into intuitionistic logic **Int** wrt to validity and that it can be faithfully embedded into dual intuitionistic logic **DualInt** wrt to dual validity.

A certain subsystem of **BiInt** has recently been investigated under the name 'da Costa logic', **daC**. A natural deduction calculus for **daC** can be found in (Priest 2009).

We will pursue the idea of combining the notions of verification and falsification in a single system of natural deduction.

One might expect that this move will in lead to a logic that is known as bi-intuitionistic logic **BiInt**.

It will not.

We will obtain another kind of a bi-intuitionistic propositional logic that combines verification and its dual. This logic, **2Int**, is motivated by a certain dualization of the natural deduction rules for intuitionistic propositional logic. It is shown that **2Int** can be faithfully embedded into intuitionistic logic **Int** wrt to validity and that it can be faithfully embedded into dual intuitionistic logic **DualInt** wrt to dual validity.

A certain subsystem of **BiInt** has recently been investigated under the name 'da Costa logic', **daC**. A natural deduction calculus for **daC** can be found in (Priest 2009).

Definition

Let Φ be a denumerable set of atomic formulas. Elements from Φ will be denoted $p, q, r, p_1, p_2, \ldots$ etc. Formulas generated from Φ will be denoted $A, B, C, D, A_1, A_2, \ldots$ etc. The propositional languages $\mathcal{L}_{\mathbf{Int}}$, $\mathcal{L}_{\mathbf{DualInt}}$, and $\mathcal{L}_{\mathbf{BiInt}}$ of **Int**, **DualInt**, and **BiInt** are defined in Backus–Naur form as follows:

$$
\begin{array}{rll}
\textbf{Int} & A ::= & p \mid \bot \mid \top \mid (A \wedge A) \mid (A \vee A) \mid (A \rightarrow A) \\
\textbf{DualInt} & A ::= & p \mid \bot \mid \top \mid (A \wedge A) \mid (A \vee A) \mid (A \prec A) \\
\textbf{BiInt} & A ::= & p \mid \bot \mid \top \mid (A \wedge A) \mid (A \vee A) \mid (A \rightarrow A) \mid (A \prec A).
\end{array}
$$

Syntactically, the language $\mathcal{L}_{\mathbf{2Int}}$ of **2Int** is the same as $\mathcal{L}_{\mathbf{BiInt}}$. Note, however, that the interpretation of the symbol $\prec$ in **2Int** will be different from the interpretation of $\prec$ in **BiInt**.

Definition

Let Φ be a denumerable set of atomic formulas. Elements from Φ will be denoted $p, q, r, p_1, p_2, \ldots$ etc. Formulas generated from Φ will be denoted $A, B, C, D, A_1, A_2, \ldots$ etc. The propositional languages $\mathcal{L}_{\mathbf{Int}}$, $\mathcal{L}_{\mathbf{DualInt}}$, and $\mathcal{L}_{\mathbf{BiInt}}$ of **Int**, **DualInt**, and **BiInt** are defined in Backus–Naur form as follows:

$$
\begin{array}{rll}
\mathbf{Int} & A ::= & p \mid \bot \mid \top \mid (A \wedge A) \mid (A \vee A) \mid (A \rightarrow A) \\
\mathbf{DualInt} & A ::= & p \mid \bot \mid \top \mid (A \wedge A) \mid (A \vee A) \mid (A \prec A) \\
\mathbf{BiInt} & A ::= & p \mid \bot \mid \top \mid (A \wedge A) \mid (A \vee A) \mid (A \rightarrow A) \mid (A \prec A).
\end{array}
$$

Syntactically, the language $\mathcal{L}_{\mathbf{2Int}}$ of **2Int** is the same as $\mathcal{L}_{\mathbf{BiInt}}$. Note, however, that the interpretation of the symbol $\prec$ in **2Int** will be different from the interpretation of $\prec$ in **BiInt.**

In **BiInt**, $\top$ is definable as $(p \to p)$ and $\bot$ is definable as $(p \prec p)$, for some fixed atom p. In **BiInt** and **DualInt** the co-negation $-A$ of A is defined as $(\top \prec A)$. In **BiInt** and **Int** the intuitionistic negation $\neg A$ of A is defined as $(A \to \bot)$. Also in **2Int** we define $-A$ as $(\top \prec A)$ and $\neg A$ as $(A \to \bot)$; however $\bot$ is primitive in **2Int**.

The idea now is to dualize the familiar introduction and elimination rules for intuitionistic propositional logic by employing a primitive notion of dual proof (or reductio).

If we use single-line rules for proofs and double-line rules for dual proofs, we may transform the intuitionistic proof rules into dual proof rules by replacing $\top$, $\bot$, $\wedge$, $\vee$, and $\to$ by their respective duals and single lines by double lines:

In **BiInt**, $\top$ is definable as $(p \to p)$ and $\bot$ is definable as $(p \prec p)$, for some fixed atom p. In **BiInt** and **DualInt** the co-negation $-A$ of A is defined as $(\top \prec A)$. In **BiInt** and **Int** the intuitionistic negation $\neg A$ of A is defined as $(A \to \bot)$. Also in **2Int** we define $-A$ as $(\top \prec A)$ and $\neg A$ as $(A \to \bot)$; however $\bot$ is primitive in **2Int**.

The idea now is to dualize the familiar introduction and elimination rules for intuitionistic propositional logic by employing a primitive notion of dual proof (or reductio).

If we use single-line rules for proofs and double-line rules for dual proofs, we may transform the intuitionistic proof rules into dual proof rules by replacing $\top$, $\bot$, $\wedge$, $\vee$, and $\to$ by their respective duals and single lines by double lines:

In **BiInt**, $\top$ is definable as $(p \to p)$ and $\bot$ is definable as $(p \prec p)$, for some fixed atom p. In **BiInt** and **DualInt** the co-negation $-A$ of A is defined as $(\top \prec A)$. In **BiInt** and **Int** the intuitionistic negation $\neg A$ of A is defined as $(A \to \bot)$. Also in **2Int** we define $-A$ as $(\top \prec A)$ and $\neg A$ as $(A \to \bot)$; however $\bot$ is primitive in **2Int**.

The idea now is to dualize the familiar introduction and elimination rules for intuitionistic propositional logic by employing a primitive notion of dual proof (or reductio).

If we use single-line rules for proofs and double-line rules for dual proofs, we may transform the intuitionistic proof rules into dual proof rules by replacing $\top$, $\bot$, $\wedge$, $\vee$, and $\to$ by their respective duals and single lines by double lines:

$$
\begin{array}{c}\hline \top\end{array}
\quad\rightsquigarrow\quad
\begin{array}{c}\hline\hline \bot\end{array}
\qquad\qquad
\begin{array}{c}\begin{array}{c}\hline \bot\end{array}\\ \hline A\end{array}
\quad\rightsquigarrow\quad
\begin{array}{c}\begin{array}{c}\hline\hline \top\end{array}\\ \hline\hline A\end{array}
$$

$$
\begin{array}{c}\overline{A} \quad \overline{B}\\ \hline (A \wedge B)\end{array}
\quad\rightsquigarrow\quad
\begin{array}{c}\begin{array}{c}\hline\hline A\end{array} \quad \begin{array}{c}\hline\hline B\end{array}\\ \hline\hline (A \vee B)\end{array}
$$

$$
\begin{array}{c}\overline{(A \wedge B)}\\ \hline A\end{array}
\quad\rightsquigarrow\quad
\begin{array}{c}\begin{array}{c}\hline\hline (A \vee B)\end{array}\\ \hline\hline A\end{array}
\qquad
\begin{array}{c}\overline{(A \wedge B)}\\ \hline B\end{array}
\quad\rightsquigarrow\quad
\begin{array}{c}\begin{array}{c}\hline\hline (A \vee B)\end{array}\\ \hline\hline B\end{array}
$$

$$
\begin{array}{c}\overline{A}\\ \hline (A \vee B)\end{array}
\quad\rightsquigarrow\quad
\begin{array}{c}\begin{array}{c}\hline\hline A\end{array}\\ \hline\hline (A \wedge B)\end{array}
\qquad
\begin{array}{c}\overline{B}\\ \hline (A \vee B)\end{array}
\quad\rightsquigarrow\quad
\begin{array}{c}\begin{array}{c}\hline\hline B\end{array}\\ \hline\hline (A \wedge B)\end{array}
$$

$$\dfrac{\overline{(A \vee B)} \qquad \dfrac{\begin{matrix}[A]\\ \vdots\end{matrix}}{C} \qquad \dfrac{\begin{matrix}[B]\\ \vdots\end{matrix}}{C}}{C} \quad \rightsquigarrow \quad \dfrac{\overline{\overline{(A \wedge B)}} \qquad \dfrac{\begin{matrix}[\![A]\!]\\ \vdots\end{matrix}}{C} \qquad \dfrac{\begin{matrix}[\![B]\!]\\ \vdots\end{matrix}}{C}}{C}$$

$$\dfrac{\dfrac{\begin{matrix}[A]\\ \vdots\end{matrix}}{B}}{(A \to B)} \quad \rightsquigarrow \quad \dfrac{\dfrac{\begin{matrix}[\![A]\!]\\ \vdots\end{matrix}}{B}}{(B \mathrel{-\!\!<} A)}$$

$$\dfrac{\overline{A} \qquad \overline{(A \to B)}}{B} \quad \rightsquigarrow \quad \dfrac{\overline{\overline{A}} \qquad \overline{\overline{(B \mathrel{-\!\!<} A)}}}{B}$$

Here $[A]$ indicates the cancellation of A as an assumption (a formula taken to be true) and $[\![A]\!]$ indicates the cancellation of A as a counterassumption (a formula taken to be false).

A dualization of natural deduction for **Int** can also be found in a recent paper by Luca Tranchini (2012), where natural deduction proofs in intuitionistic logic are inverted and formulas are replaced by their duals.

This leads to a *multiple-conclusion* natural deduction proof system. Tranchini observes that the dualization of formulas gives rise to an isomorphism between proofs in the language $\mathcal{L}_{\mathbf{Int}}$ and dualized proofs in the language $\mathcal{L}_{\mathbf{DualInt}}$. It is not difficult to see that the dualization of $\mathcal{L}_{\mathbf{Int}}$-formulas also gives rise to an isomorphism between natural deduction proofs in intuitionistic propositional logic and the above single-conclusion dualized proofs.

A dualization of natural deduction for **Int** can also be found in a recent paper by Luca Tranchini (2012), where natural deduction proofs in intuitionistic logic are inverted and formulas are replaced by their duals.

This leads to a *multiple-conclusion* natural deduction proof system. Tranchini observes that the dualization of formulas gives rise to an isomorphism between proofs in the language $\mathcal{L}_{\mathbf{Int}}$ and dualized proofs in the language $\mathcal{L}_{\mathbf{DualInt}}$. It is not difficult to see that the dualization of $\mathcal{L}_{\mathbf{Int}}$-formulas also gives rise to an isomorphism between natural deduction proofs in intuitionistic propositional logic and the above single-conclusion dualized proofs.

The idea of dualization does not induce rules for the falsification of implications and the verification of co-implications, because co-implication is not among the connectives of **Int**, and implication is not among the connectives of **DualInt.**

Moreover, the verification conditions for co-implications and the falsification conditions for implications will comprise both verification and falsification conditions.

Obtaining a natural deduction proof system for the combined vocabulary of **Int** and **DualInt** by dualization of the proof rules of **Int** (or the falsification rules of **DualInt**) will be simpler if all rules are single-conclusion rules in comparison with defining a proof system that combines single-conclusion and multiple-conclusion rules.

The idea of dualization does not induce rules for the falsification of implications and the verification of co-implications, because co-implication is not among the connectives of **Int**, and implication is not among the connectives of **DualInt.**

Moreover, the verification conditions for co-implications and the falsification conditions for implications will comprise both verification and falsification conditions.

Obtaining a natural deduction proof system for the combined vocabulary of **Int** and **DualInt** by dualization of the proof rules of **Int** (or the falsification rules of **DualInt**) will be simpler if all rules are single-conclusion rules in comparison with defining a proof system that combines single-conclusion and multiple-conclusion rules.

The idea of dualization does not induce rules for the falsification of implications and the verification of co-implications, because co-implication is not among the connectives of **Int**, and implication is not among the connectives of **DualInt.**

Moreover, the verification conditions for co-implications and the falsification conditions for implications will comprise both verification and falsification conditions.

Obtaining a natural deduction proof system for the combined vocabulary of **Int** and **DualInt** by dualization of the proof rules of **Int** (or the falsification rules of **DualInt**) will be simpler if all rules are single-conclusion rules in comparison with defining a proof system that combines single-conclusion and multiple-conclusion rules.

We will here take an orthodox stance on the falsification of implications and the verification of co-implications. This is not the only option, but the choice of the following inference rules is suggested, for example, by comments Dag Prawitz has made on canonical refutations of implications:

$$\begin{array}{c} \overline{A} \quad \overline{\overline{B}} \\ \hline\hline (A \to B) \end{array} \qquad \begin{array}{c} \overline{\overline{(A \to B)}} \\ \hline A \end{array} \qquad \begin{array}{c} \overline{\overline{(A \to B)}} \\ \hline\hline B \end{array}$$

$$\begin{array}{c} \overline{A} \quad \overline{\overline{B}} \\ \hline (A \mathrel{-\!\!\!<} B) \end{array} \qquad \begin{array}{c} \overline{(A \mathrel{-\!\!\!<} B)} \\ \hline A \end{array} \qquad \begin{array}{c} \overline{(A \mathrel{-\!\!\!<} B)} \\ \hline\hline B \end{array}$$

We will here take an orthodox stance on the falsification of implications and the verification of co-implications. This is not the only option, but the choice of the following inference rules is suggested, for example, by comments Dag Prawitz has made on canonical refutations of implications:

$$
\begin{array}{c} \overline{A} \quad \overline{\overline{B}} \\ \hline\hline (A \to B) \end{array}
\qquad\qquad
\begin{array}{c} \overline{\overline{(A \to B)}} \\ \hline A \end{array}
\qquad\qquad
\begin{array}{c} \overline{\overline{(A \to B)}} \\ \hline\hline B \end{array}
$$

$$
\begin{array}{c} \overline{A} \quad \overline{\overline{B}} \\ \hline (A \relbar\!\!\prec B) \end{array}
\qquad\qquad
\begin{array}{c} \overline{(A \relbar\!\!\prec B)} \\ \hline A \end{array}
\qquad\qquad
\begin{array}{c} \overline{(A \relbar\!\!\prec B)} \\ \hline\hline B \end{array}
$$

Let us refer to the above set of rules as the natural deduction proof system N2Int, and let NJ denote the familiar natural deduction proof system for intuitionistic propositional logic **Int** in the language $\mathcal{L}_{\mathbf{Int}}$.

We may define by a simultaneous induction the following two predicates:

$\mathrm{P}(\Pi, A, (\Delta; \Gamma))$ ("Π is a proof of A from the finite set of assumptions Δ and the finite set of counterassumptions Γ")

and

$\mathrm{DP}(\Pi, A, (\Delta; \Gamma))$ ("Π is a dual proof of A from the finite set of assumptions Δ and the finite set of counterassumptions Γ").

Let us refer to the above set of rules as the natural deduction proof system N2Int, and let NJ denote the familiar natural deduction proof system for intuitionistic propositional logic **Int** in the language $\mathcal{L}_{\mathbf{Int}}$.

We may define by a simultaneous induction the following two predicates:

$\mathrm{P}(\Pi, A, (\Delta; \Gamma))$ ("Π is a proof of *A* from the finite set of assumptions Δ and the finite set of counterassumptions Γ")

and

$\mathrm{DP}(\Pi, A, (\Delta; \Gamma))$ ("Π is a dual proof of *A* from the finite set of assumptions Δ and the finite set of counterassumptions Γ").

A formula A is provable (dually provable) in N2Int from the pair $(\Delta; \Gamma)$ iff there exists a proof (dual proof) Π with $\mathrm{P}(\Pi, A, (\Delta; \Gamma))$ $(\mathrm{DP}(\Pi, A, (\Delta; \Gamma)))$.

The following derivations are examples of derivations in N2Int.

$$\dfrac{\dfrac{\overline{\top} \quad \dfrac{[((B \to B) \prec A)]^1}{A}}{(\top \prec A)}}{(((B \to B) \prec A) \to (\top \prec A))}\ 1$$

A formula A is provable (dually provable) in N2Int from the pair $(\Delta; \Gamma)$ iff there exists a proof (dual proof) Π with $\mathrm{P}(\Pi, A, (\Delta; \Gamma))$ $(\mathrm{DP}(\Pi, A, (\Delta; \Gamma)))$.

The following derivations are examples of derivations in N2Int.

$$\dfrac{\dfrac{\dfrac{}{\top} \quad \dfrac{[((B \to B) \prec A)]^1}{A}}{(\top \prec A)}}{(((B \to B) \prec A) \to (\top \prec A))}\,1$$

$$
\begin{prooftree}
\AxiomC{}
\UnaryInfC{$\top$}
\AxiomC{$[((\top \mathrel{-\!\!\!<} A) \wedge (\top \mathrel{-\!\!\!<} B))]^1$}
\UnaryInfC{$(\top \mathrel{-\!\!\!<} A)$}
\doubleLine
\UnaryInfC{A}
\AxiomC{$[((\top \mathrel{-\!\!\!<} A) \wedge (\top \mathrel{-\!\!\!<} B))]^1$}
\UnaryInfC{$(\top \mathrel{-\!\!\!<} B)$}
\doubleLine
\UnaryInfC{B}
\doubleLine
\BinaryInfC{$(A \vee B)$}
\BinaryInfC{$(\top \mathrel{-\!\!\!<} (A \vee B))$}
\RightLabel{1}
\UnaryInfC{$(((\top \mathrel{-\!\!\!<} A) \wedge (\top \mathrel{-\!\!\!<} B)) \rightarrow -(A \vee B))$}
\end{prooftree}
$$

The dualization gives rise to all the De Morgan laws for co-negation, in particular, as shown by the last example, the De Morgan law $(-A \wedge -B) \rightarrow -(A \vee B)$ is provable in N2Int (from $(\varnothing; \varnothing)$).

$$
\dfrac{\dfrac{\dfrac{}{\top} \qquad \dfrac{\dfrac{\dfrac{[((\top \mathrel{-\!\!<} A) \wedge (\top \mathrel{-\!\!<} B))]^1}{(\top \mathrel{-\!\!<} A)}}{A} \qquad \dfrac{\dfrac{[((\top \mathrel{-\!\!<} A) \wedge (\top \mathrel{-\!\!<} B))]^1}{(\top \mathrel{-\!\!<} B)}}{B}}{(A \vee B)}}{(\top \mathrel{-\!\!<} (A \vee B))}}{(((\top \mathrel{-\!\!<} A) \wedge (\top \mathrel{-\!\!<} B)) \to -(A \vee B))}\ 1
$$

The dualization gives rise to all the De Morgan laws for co-negation, in particular, as shown by the last example, the De Morgan law $(-A \wedge -B) \to -(A \vee B)$ is provable in N2Int (from $(\varnothing; \varnothing)$).

What we have done so far may also be described in more general terms by remarking that we are dealing with *two primitive sorts* of derivations, namely proofs and dual proofs, that can be combined with each other to obtain more complex proofs and dual proofs. In this way, verification and a weak notion of falsification are taken to be primitive and *pari passu*.

Definition

A frame is a pre-order $\langle I, \leq \rangle$. We may think of I as a non-empty set of information states and may view the reflexive and transitive relation $\leq$ as a relation of possible expansion of information states. Instead of $w \leq w'$, we also write $w' \geq w$.

What we have done so far may also be described in more general terms by remarking that we are dealing with *two primitive sorts* of derivations, namely proofs and dual proofs, that can be combined with each other to obtain more complex proofs and dual proofs. In this way, verification and a weak notion of falsification are taken to be primitive and *pari passu*.

Definition

A frame is a pre-order $\langle I, \leq \rangle$. We may think of I as a non-empty set of information states and may view the reflexive and transitive relation $\leq$ as a relation of possible expansion of information states. Instead of $w \leq w'$, we also write $w' \geq w$.

Definition

A model for **Int** is a structure $\langle I, \leq, v^+ \rangle$, where $\langle I, \leq \rangle$ is a frame and v^+ is a function that maps every atom p to a subset of I (namely the states that support the truth of p). The function v^+ satisfies the following persistence (or heredity) property: for every $w' \geq w$ and every $p \in \Phi$, $w \in v^+(p)$ implies $w' \in v^+(p)$. The relation $\mathcal{M}, w \models^+ A$ ('state w supports the truth of $\mathcal{L}_{\mathbf{Int}}$-formula A in model $\mathcal{M}$') is inductively defined as follows:

$$
\begin{array}{lll}
\mathcal{M}, w \models^+ p & \text{iff} & w \in v^+(p) \\
\mathcal{M}, w \models^+ \top & & \\
\mathcal{M}, w \not\models^+ \bot & & \\
\mathcal{M}, w \models^+ (A \wedge B) & \text{iff} & \mathcal{M}, w \models^+ A \text{ and } \mathcal{M}, w \models^+ B \\
\mathcal{M}, w \models^+ (A \vee B) & \text{iff} & \mathcal{M}, w \models^+ A \text{ or } \mathcal{M}, w \models^+ B \\
\mathcal{M}, w \models^+ \neg A & \text{iff} & \text{for every } w' \geq w : \mathcal{M}, w' \not\models^+ A \\
\mathcal{M}, w \models^+ (A \rightarrow B) & \text{iff} & \text{for every } w' \geq w : \mathcal{M}, w' \not\models^+ A \text{ or } \mathcal{M}, w' \models^+ B
\end{array}
$$

where $\mathcal{M}, w \not\models^+ A$ is the classical negation of $\mathcal{M}, w \models^+ A$.
An $\mathcal{L}_{\mathbf{Int}}$-formula A is valid in a model for **Int** $\mathcal{M} = \langle I, \leq, v \rangle$ iff for every $w \in I$, $\mathcal{M}, w \models^+ A$; A is valid in **Int** iff it is valid in every model for **Int**. We write $\models_{\mathbf{Int}} A$ to mean that A is valid in **Int**.

Definition

A model for **DualInt** is a structure $\langle I, \leq, v^- \rangle$, where $\langle I, \leq \rangle$ is a frame and v^- is a function that maps every atom p to a subset of I (namely the states that support the falsity of p). The function v^- satisfies the following persistence (or heredity) property: for every $w' \geq w$ and every $p \in \Phi$, $w \in v^-(p)$ implies $w' \in v^-(p)$. The relation $\mathcal{M}, w \models^- A$ ('state w supports the falsity of $\mathcal{L}_{\mathbf{DualInt}}$-formula A in model $\mathcal{M}$') is inductively defined as follows:

$$
\begin{array}{lll}
\mathcal{M}, w \models^- p & \text{iff} & w \in v^-(p) \\
\mathcal{M}, w \not\models^- \top & & \\
\mathcal{M}, w \models^- \bot & & \\
\mathcal{M}, w \models^- (A \wedge B) & \text{iff} & \mathcal{M}, w \models^- A \text{ or } \mathcal{M}, w \models^- B \\
\mathcal{M}, w \models^- (A \vee B) & \text{iff} & \mathcal{M}, w \models^- A \text{ and } \mathcal{M}, w \models^- B \\
\mathcal{M}, w \models^- -A & \text{iff} & \text{for every } w' \geq w : \mathcal{M}, w' \not\models^- A \\
\mathcal{M}, w \models^- (A \multimap\!\!\!< B) & \text{iff} & \text{for every } w' \geq w : \mathcal{M}, w' \not\models^- B \text{ or } \mathcal{M}, w' \models^- A
\end{array}
$$

where $\mathcal{M}, w \not\models^- A$ is the classical negation of $\mathcal{M}, w \models^- A$.

If we denote the relation $\models^-$ as $\not\models^+$ (and $\not\models^-$ as $\models^+$), then we can rewrite the previous definition as follows:

Definition

A model for **DualInt** is a structure $\langle I, \leq, v^- \rangle$, where $\langle I, \leq \rangle$ is a frame and v^- is a function that maps every atom p to a subset of I (namely the states that support the falsity of p). The function v^- satisfies the following persistence (or heredity) property: for every $w' \geq w$ and every $p \in \Phi$, $w \in v^-(p)$ implies $w' \in v^-(p)$. The relation $\mathcal{M}, w \models^- A$ ('state w supports the falsity of $\mathcal{L}_{\mathbf{DualInt}}$-formula A in model $\mathcal{M}$') is inductively defined as follows:

$$
\begin{array}{lll}
\mathcal{M}, w \models^- p & \text{iff} & w \in v^-(p) \\
\mathcal{M}, w \not\models^- \top & & \\
\mathcal{M}, w \models^- \bot & & \\
\mathcal{M}, w \models^- (A \wedge B) & \text{iff} & \mathcal{M}, w \models^- A \text{ or } \mathcal{M}, w \models^- B \\
\mathcal{M}, w \models^- (A \vee B) & \text{iff} & \mathcal{M}, w \models^- A \text{ and } \mathcal{M}, w \models^- B \\
\mathcal{M}, w \models^- -A & \text{iff} & \text{for every } w' \geq w : \mathcal{M}, w' \not\models^- A \\
\mathcal{M}, w \models^- (A \prec B) & \text{iff} & \text{for every } w' \geq w : \mathcal{M}, w' \not\models^- B \text{ or } \mathcal{M}, w' \models^- A
\end{array}
$$

where $\mathcal{M}, w \not\models^- A$ is the classical negation of $\mathcal{M}, w \models^- A$.

If we denote the relation $\models^-$ as $\not\models^+$ (and $\not\models^-$ as $\models^+$), then we can rewrite the previous definition as follows:

Definition

A model for **DualInt** is a structure $\langle I, \leq, v^- \rangle$, where $\langle I, \leq \rangle$ is a frame and v^- is a function that maps every atom p to a subset of I (namely the states that support the falsity of p). The function v^- satisfies persistence: for every $w' \geq w$ and every $p \in \Phi$, $w \in v^-(p)$ implies $w' \in v^-(p)$. The relation $\mathcal{M}, w \models^+ A$ ('state w fails to support the falsity of $\mathcal{L}_{\mathbf{DualInt}}$-formula A in model $\mathcal{M}$') is inductively defined as follows:

$$
\begin{array}{lll}
\mathcal{M}, w \models^+ p & \text{iff} & w \notin v^-(p) \\
\mathcal{M}, w \models^+ \top & & \\
\mathcal{M}, w \not\models^+ \bot & & \\
\mathcal{M}, w \models^+ (A \wedge B) & \text{iff} & \mathcal{M}, w \models^+ A \text{ and } \mathcal{M}, w \models^+ B \\
\mathcal{M}, w \models^+ (A \vee B) & \text{iff} & \mathcal{M}, w \models^+ A \text{ or } \mathcal{M}, w \models^+ B \\
\mathcal{M}, w \models^+ -A & \text{iff} & \text{there exists } w' \geq w : \mathcal{M}, w' \not\models^+ A \\
\mathcal{M}, w \models^+ (A \prec B) & \text{iff} & \text{there exists } w' \geq w : \mathcal{M}, w' \models^+ A \text{ and } \mathcal{M}, w' \not\models^+ B.
\end{array}
$$

Definition

An $\mathcal{L}_{\mathbf{DualInt}}$-formula A is valid in a model for **DualInt** $\mathcal{M} = \langle I, \leq, v^- \rangle$ iff for every $w \in I$, $\mathcal{M}, w \models^+ A$; A is valid in **DualInt** iff it is valid in every model for **DualInt**. We write $\models_{\mathbf{DualInt}} A$ to mean that A is valid in **DualInt**.

Definition

A model for **DualInt** is a structure $\langle I, \leq, v^- \rangle$, where $\langle I, \leq \rangle$ is a frame and v^- is a function that maps every atom p to a subset of I (namely the states that support the falsity of p). The function v^- satisfies persistence: for every $w' \geq w$ and every $p \in \Phi$, $w \in v^-(p)$ implies $w' \in v^-(p)$. The relation $\mathcal{M}, w \models^+ A$ ('state w fails to support the falsity of $\mathcal{L}_{\mathbf{DualInt}}$-formula A in model $\mathcal{M}$') is inductively defined as follows:

$$
\begin{array}{lll}
\mathcal{M}, w \models^+ p & \text{iff} & w \notin v^-(p) \\
\mathcal{M}, w \models^+ \top & & \\
\mathcal{M}, w \not\models^+ \bot & & \\
\mathcal{M}, w \models^+ (A \wedge B) & \text{iff} & \mathcal{M}, w \models^+ A \text{ and } \mathcal{M}, w \models^+ B \\
\mathcal{M}, w \models^+ (A \vee B) & \text{iff} & \mathcal{M}, w \models^+ A \text{ or } \mathcal{M}, w \models^+ B \\
\mathcal{M}, w \models^+ -A & \text{iff} & \text{there exists } w' \geq w : \mathcal{M}, w' \not\models^+ A \\
\mathcal{M}, w \models^+ (A \prec B) & \text{iff} & \text{there exists } w' \geq w : \mathcal{M}, w' \models^+ A \text{ and } \mathcal{M}, w' \not\models^+ B.
\end{array}
$$

Definition

An $\mathcal{L}_{\mathbf{DualInt}}$-formula A is valid in a model for **DualInt** $\mathcal{M} = \langle I, \leq, v^- \rangle$ iff for every $w \in I$, $\mathcal{M}, w \models^+ A$; A is valid in **DualInt** iff it is valid in every model for **DualInt**. We write $\models_{\mathbf{DualInt}} A$ to mean that A is valid in **DualInt**.

Let $\langle I_1, \leq_1, v^+\rangle$ be a model for **Int** and let $\langle I_2, \leq_2, v^-\rangle$ be a model for **DualInt**. To obtain models for a *bi-intuitionistic* logic, it is natural to envisage a combination of both kinds of models.

What is done in the literature is to define a merged notion. A single non-empty set I of information states is assumed and a single pre-order $\leq$ (on I), with the understanding that $\leq$ plays the role of $\leq_1$ and $\leq^{-1}$ plays the role of $\leq_2$, where $\leq^{-1}$ is the inverse of $\leq$.

Moreover, a single valuation function v is assumed, with the understanding that (i) v takes the part of v^+ and (ii) v^- is still around if one thinks of v^- as defined by postulating that for every $w \in I$: $w \in v^-(p)$ iff $w \notin v(p)$.

Let $\langle I_1, \leq_1, v^+\rangle$ be a model for **Int** and let $\langle I_2, \leq_2, v^-\rangle$ be a model for **DualInt**. To obtain models for a *bi-intuitionistic* logic, it is natural to envisage a combination of both kinds of models.

What is done in the literature is to define a merged notion. A single non-empty set I of information states is assumed and a single pre-order $\leq$ (on I), with the understanding that $\leq$ plays the role of $\leq_1$ and $\leq^{-1}$ plays the role of $\leq_2$, where $\leq^{-1}$ is the inverse of $\leq$.

Moreover, a single valuation function v is assumed, with the understanding that (i) v takes the part of v^+ and (ii) v^- is still around if one thinks of v^- as defined by postulating that for every $w \in I$: $w \in v^-(p)$ iff $w \notin v(p)$.

Let $\langle I_1, \leq_1, v^+ \rangle$ be a model for **Int** and let $\langle I_2, \leq_2, v^- \rangle$ be a model for **DualInt**. To obtain models for a *bi-intuitionistic* logic, it is natural to envisage a combination of both kinds of models.

What is done in the literature is to define a merged notion. A single non-empty set I of information states is assumed and a single pre-order $\leq$ (on I), with the understanding that $\leq$ plays the role of $\leq_1$ and $\leq^{-1}$ plays the role of $\leq_2$, where $\leq^{-1}$ is the inverse of $\leq$.

Moreover, a single valuation function v is assumed, with the understanding that (i) v takes the part of v^+ and (ii) v^- is still around if one thinks of v^- as defined by postulating that for every $w \in I$: $w \in v^-(p)$ iff $w \notin v(p)$.

Definition

A model for **BiInt** is a structure $\langle I, \leq, v\rangle$, where $\langle I, \leq\rangle$ is a frame and v is a function that maps every atom p to a subset of I (namely the states that support the truth of p). The function v satisfies the following persistence property: for every $w' \geq w$ and every $p \in \Phi$, $w \in v(p)$ implies $w' \in v(p)$. The relation $\mathcal{M}, w \models A$ ('state w supports the truth of $\mathcal{L}_{\mathbf{BiInt}}$-formula A in model $\mathcal{M}$') is inductively defined as follows:

$$
\begin{array}{lll}
\mathcal{M}, w \models p & \text{iff} & w \in v(p) \\
\mathcal{M}, w \models \top & & \\
\mathcal{M}, w \not\models \bot & & \\
\mathcal{M}, w \models (A \wedge B) & \text{iff} & \mathcal{M}, w \models A \text{ and } \mathcal{M}, w \models B \\
\mathcal{M}, w \models (A \vee B) & \text{iff} & \mathcal{M}, w \models A \text{ or } \mathcal{M}, w \models B \\
\mathcal{M}, w \models \neg A & \text{iff} & \text{for every } w' \geq w : \mathcal{M}, w' \not\models A \\
\mathcal{M}, w \models (A \rightarrow B) & \text{iff} & \text{for every } w' \geq w : \mathcal{M}, w' \not\models A \text{ or } \mathcal{M}, w' \models B \\
\mathcal{M}, w \models -A & \text{iff} & \text{there exists } w' \leq w : \mathcal{M}, w' \not\models A \\
\mathcal{M}, w \models (A \prec B) & \text{iff} & \text{there exists } w' \leq w \text{ with } \mathcal{M}, w' \models A \text{ and } \mathcal{M}, w' \not\models B.
\end{array}
$$

An $\mathcal{L}_{\mathbf{BiInt}}$-formula A is valid in a model for **BiInt** $\mathcal{M} = \langle I, \leq, v\rangle$ iff for every $w \in I$, $\mathcal{M}, w \models A$; A is valid in **BiInt** iff it is valid in every model for **BiInt**. We write $\models_{\mathbf{BiInt}} A$ to mean that A is valid in **BiInt**.

In (Goré 2000), **DualInt** is defined as the set of all $\mathcal{L}_{\textbf{DualInt}}$-formulas valid in **BiInt**. For $\mathcal{L}_{\textbf{DualInt}}$-formulas, validity in **BiInt** and validity in **DualInt** coincide.

Lemma

Let $\mathcal{M} = \langle I, \leq, v^- \rangle$ be a model for **DualInt**. *The structure $\mathcal{M}' = \langle I, \leq^{-1}, v \rangle$, where for every $p \in \Phi$ and every $w \in I$, $w \in v(p)$ iff $w \notin v^-(p)$ is a model for* **BiInt** *(since the inverse of a pre-order is again a pre-order and since the persistence property holds). Then for every $\mathcal{L}_{\textbf{DualInt}}$-formula A, and every $w \in I$,*

$$\mathcal{M}, w \models^+ A \text{ iff } \mathcal{M}', w \models A.$$

Lemma

Let $\mathcal{M} = \langle I, \leq, v \rangle$ be a model for **BiInt**. *The structure $\mathcal{M}' = \langle I, \leq^{-1}, v^- \rangle$, where for every $p \in \Phi$ and every $w \in I$, $w \in v^-(p)$ iff $w \notin v(p)$ is a model for* **DualInt**. *Then for every $\mathcal{L}_{\textbf{DualInt}}$-formula A, and every $w \in I$,*

$$\mathcal{M}, w \models A \text{ iff } \mathcal{M}', w \models^+ A.$$

Proposition

For every $\mathcal{L}_{\textbf{DualInt}}$-formula A, $\models_{\textbf{DualInt}} A$ iff $\models_{\textbf{BiInt}} A$.

In (Goré 2000), **DualInt** is defined as the set of all $\mathcal{L}_{\textbf{DualInt}}$-formulas valid in **BiInt**. For $\mathcal{L}_{\textbf{DualInt}}$-formulas, validity in **BiInt** and validity in **DualInt** coincide.

Lemma

Let $\mathcal{M} = \langle I, \leq, v^- \rangle$ be a model for **DualInt**. *The structure $\mathcal{M}' = \langle I, \leq^{-1}, v \rangle$, where for every $p \in \Phi$ and every $w \in I$, $w \in v(p)$ iff $w \notin v^-(p)$ is a model for* **BiInt** *(since the inverse of a pre-order is again a pre-order and since the persistence property holds). Then for every $\mathcal{L}_{\textbf{DualInt}}$-formula A, and every $w \in I$,*

$$\mathcal{M}, w \models^+ A \text{ iff } \mathcal{M}', w \models A.$$

Lemma

Let $\mathcal{M} = \langle I, \leq, v \rangle$ be a model for **BiInt**. *The structure $\mathcal{M}' = \langle I, \leq^{-1}, v^- \rangle$, where for every $p \in \Phi$ and every $w \in I$, $w \in v^-(p)$ iff $w \notin v(p)$ is a model for* **DualInt**. *Then for every $\mathcal{L}_{\textbf{DualInt}}$-formula A, and every $w \in I$,*

$$\mathcal{M}, w \models A \text{ iff } \mathcal{M}', w \models^+ A.$$

Proposition

For every $\mathcal{L}_{\textbf{DualInt}}$-formula A, $\models_{\textbf{DualInt}} A$ iff $\models_{\textbf{BiInt}} A$.

In (Goré 2000), **DualInt** is defined as the set of all $\mathcal{L}_{\mathbf{DualInt}}$-formulas valid in **BiInt**. For $\mathcal{L}_{\mathbf{DualInt}}$-formulas, validity in **BiInt** and validity in **DualInt** coincide.

Lemma

Let $\mathcal{M} = \langle I, \leq, v^- \rangle$ be a model for **DualInt**. *The structure $\mathcal{M}' = \langle I, \leq^{-1}, v \rangle$, where for every $p \in \Phi$ and every $w \in I$, $w \in v(p)$ iff $w \notin v^-(p)$ is a model for* **BiInt** *(since the inverse of a pre-order is again a pre-order and since the persistence property holds). Then for every $\mathcal{L}_{\mathbf{DualInt}}$-formula A, and every $w \in I$,*

$$\mathcal{M}, w \models^+ A \text{ iff } \mathcal{M}', w \models A.$$

Lemma

Let $\mathcal{M} = \langle I, \leq, v \rangle$ be a model for **BiInt**. *The structure $\mathcal{M}' = \langle I, \leq^{-1}, v^- \rangle$, where for every $p \in \Phi$ and every $w \in I$, $w \in v^-(p)$ iff $w \notin v(p)$ is a model for* **DualInt**. *Then for every $\mathcal{L}_{\mathbf{DualInt}}$-formula A, and every $w \in I$,*

$$\mathcal{M}, w \models A \text{ iff } \mathcal{M}', w \models^+ A.$$

Proposition

For every $\mathcal{L}_{\mathbf{DualInt}}$-formula A, $\models_{\mathbf{DualInt}} A$ iff $\models_{\mathbf{BiInt}} A$.

In (Goré 2000), **DualInt** is defined as the set of all $\mathcal{L}_{\textbf{DualInt}}$-formulas valid in **BiInt**. For $\mathcal{L}_{\textbf{DualInt}}$-formulas, validity in **BiInt** and validity in **DualInt** coincide.

Lemma

Let $\mathcal{M} = \langle I, \leq, v^{-} \rangle$ be a model for **DualInt***. The structure $\mathcal{M}' = \langle I, \leq^{-1}, v \rangle$, where for every $p \in \Phi$ and every $w \in I$, $w \in v(p)$ iff $w \notin v^{-}(p)$ is a model for* **BiInt** *(since the inverse of a pre-order is again a pre-order and since the persistence property holds). Then for every $\mathcal{L}_{\textbf{DualInt}}$-formula A, and every $w \in I$,*

$$\mathcal{M}, w \models^{+} A \text{ iff } \mathcal{M}', w \models A.$$

Lemma

Let $\mathcal{M} = \langle I, \leq, v \rangle$ be a model for **BiInt***. The structure $\mathcal{M}' = \langle I, \leq^{-1}, v^{-} \rangle$, where for every $p \in \Phi$ and every $w \in I$, $w \in v^{-}(p)$ iff $w \notin v(p)$ is a model for* **DualInt***. Then for every $\mathcal{L}_{\textbf{DualInt}}$-formula A, and every $w \in I$,*

$$\mathcal{M}, w \models A \text{ iff } \mathcal{M}', w \models^{+} A.$$

Proposition

For every $\mathcal{L}_{\textbf{DualInt}}$-formula A, $\models_{\textbf{DualInt}} A$ iff $\models_{\textbf{BiInt}} A$.

Definition

A model for **2Int** is a structure $\langle I, \leq, v^+, v^- \rangle$, where $\langle I, \leq \rangle$ is a frame and v^+, v^- are functions form the set Φ of atomic formulas to subsets of I. The relations $\mathcal{M}, w \models^+ A$ ('state w supports the truth of $\mathcal{L}_{\mathbf{2Int}}$-formula A in model $\mathcal{M}$') and $\mathcal{M}, w \models^- A$ ('state w supports the falsity of $\mathcal{L}_{\mathbf{2Int}}$-formula A in model $\mathcal{M}$') are inductively defined as follows:

$$
\begin{array}{lll}
\mathcal{M}, w \models^+ p & \text{iff} & \text{for every } w' \geq w : w' \in v^+(p) \\
\mathcal{M}, w \models^- p & \text{iff} & \text{for every } w' \geq w : w' \in v^-(p) \\
\mathcal{M}, w \models^+ \top & & \\
\mathcal{M}, w \not\models^- \top & & \\
\mathcal{M}, w \not\models^+ \bot & & \\
\mathcal{M}, w \models^- \bot & & \\
\mathcal{M}, w \models^+ (A \wedge B) & \text{iff} & \mathcal{M}, w \models^+ A \text{ and } \mathcal{M}, w \models^+ B \\
\mathcal{M}, w \models^- (A \wedge B) & \text{iff} & \mathcal{M}, w \models^- A \text{ or } \mathcal{M}, w \models^- B \\
\mathcal{M}, w \models^+ (A \vee B) & \text{iff} & \mathcal{M}, w \models^+ A \text{ or } \mathcal{M}, w \models^+ B \\
\mathcal{M}, w \models^- (A \vee B) & \text{iff} & \mathcal{M}, w \models^- A \text{ and } \mathcal{M}, w \models^- B \\
\mathcal{M}, w \models^+ (A \rightarrow B) & \text{iff} & \text{for every } w' \geq w : \mathcal{M}, w' \not\models^+ A \text{ or } \mathcal{M}, w' \models^+ B \\
\mathcal{M}, w \models^- (A \rightarrow B) & \text{iff} & \mathcal{M}, w \models^+ A \text{ and } \mathcal{M}, w \models^- B
\end{array}
$$

Definition

Previous definition continued:

$$
\begin{array}{lll}
\mathcal{M}, w \models^{+} \neg A & \text{iff} & \text{for every } w' \geq w : \mathcal{M}, w' \not\models^{+} A \\
\mathcal{M}, w \models^{-} \neg A & \text{iff} & \mathcal{M}, w \models^{+} A \\
\mathcal{M}, w \models^{+} -A & \text{iff} & \mathcal{M}, w \models^{-} A \\
\mathcal{M}, w \models^{-} -A & \text{iff} & \text{for every } w' \geq w : \mathcal{M}, w' \not\models^{-} A \\
\mathcal{M}, w \models^{+} (A \prec B) & \text{iff} & \mathcal{M}, w \models^{+} A \text{ and } \mathcal{M}, w \models^{-} B \\
\mathcal{M}, w \models^{-} (A \prec B) & \text{iff} & \text{for every } w' \geq w : \mathcal{M}, w' \not\models^{-} B \text{ or } \mathcal{M}, w' \models^{-} A.
\end{array}
$$

Definition

An $\mathcal{L}_{\mathbf{2Int}}$-formula A is valid in a model for **2Int** $\mathcal{M} = \langle I, \leq, v^{+}, v^{-} \rangle$ iff for every $w \in I$, $\mathcal{M}, w \models^{+} A$; A is valid in **2Int** iff it is valid in every model for **2Int**. We write $\models_{\mathbf{2Int}} A$ to mean that A is valid in **2Int**.

An $\mathcal{L}_{\mathbf{2Int}}$-formula A is dually valid in a model for **2Int** $\mathcal{M} = \langle I, \leq, v^{+}, v^{-} \rangle$ iff for every $w \in I$, $\mathcal{M}, w \models^{-} A$ (iff for every $w \in I$, $\mathcal{M}, w \models^{+} -A$); A is dually valid in **2Int** iff it is dually valid in every model for **2Int**. We write $\models^{d}_{\mathbf{2Int}} A$ to mean that A is valid in **2Int**.

Definition

Previous definition continued:

$$
\begin{array}{lll}
\mathcal{M}, w \models^+ \neg A & \text{iff} & \text{for every } w' \geq w : \mathcal{M}, w' \not\models^+ A \\
\mathcal{M}, w \models^- \neg A & \text{iff} & \mathcal{M}, w \models^+ A \\
\mathcal{M}, w \models^+ -A & \text{iff} & \mathcal{M}, w \models^- A \\
\mathcal{M}, w \models^- -A & \text{iff} & \text{for every } w' \geq w : \mathcal{M}, w' \not\models^- A \\
\mathcal{M}, w \models^+ (A \prec\!\!\!- B) & \text{iff} & \mathcal{M}, w \models^+ A \text{ and } \mathcal{M}, w \models^- B \\
\mathcal{M}, w \models^- (A \prec\!\!\!- B) & \text{iff} & \text{for every } w' \geq w : \mathcal{M}, w' \not\models^- B \text{ or } \mathcal{M}, w' \models^- A.
\end{array}
$$

Definition

An $\mathcal{L}_{\mathbf{2Int}}$-formula A is valid in a model for **2Int** $\mathcal{M} = \langle I, \leq, v^+, v^- \rangle$ iff for every $w \in I$, $\mathcal{M}, w \models^+ A$; A is valid in **2Int** iff it is valid in every model for **2Int**. We write $\models_{\mathbf{2Int}} A$ to mean that A is valid in **2Int**.

An $\mathcal{L}_{\mathbf{2Int}}$-formula A is dually valid in a model for **2Int** $\mathcal{M} = \langle I, \leq, v^+, v^- \rangle$ iff for every $w \in I$, $\mathcal{M}, w \models^- A$ (iff for every $w \in I$, $\mathcal{M}, w \models^+ -A$); A is dually valid in **2Int** iff it is dually valid in every model for **2Int**. We write $\models^d_{\mathbf{2Int}} A$ to mean that A is valid in **2Int**.

Definition

Previous definition continued:

$$
\begin{array}{lll}
\mathcal{M}, w \models^{+} \neg A & \text{iff} & \text{for every } w' \geq w : \mathcal{M}, w' \not\models^{+} A \\
\mathcal{M}, w \models^{-} \neg A & \text{iff} & \mathcal{M}, w \models^{+} A \\
\mathcal{M}, w \models^{+} -A & \text{iff} & \mathcal{M}, w \models^{-} A \\
\mathcal{M}, w \models^{-} -A & \text{iff} & \text{for every } w' \geq w : \mathcal{M}, w' \not\models^{-} A \\
\mathcal{M}, w \models^{+} (A \prec B) & \text{iff} & \mathcal{M}, w \models^{+} A \text{ and } \mathcal{M}, w \models^{-} B \\
\mathcal{M}, w \models^{-} (A \prec B) & \text{iff} & \text{for every } w' \geq w : \mathcal{M}, w' \not\models^{-} B \text{ or } \mathcal{M}, w' \models^{-} A.
\end{array}
$$

Definition

An $\mathcal{L}_{\mathbf{2Int}}$-formula A is valid in a model for **2Int** $\mathcal{M} = \langle I, \leq, v^{+}, v^{-} \rangle$ iff for every $w \in I$, $\mathcal{M}, w \models^{+} A$; A is valid in **2Int** iff it is valid in every model for **2Int**. We write $\models_{\mathbf{2Int}} A$ to mean that A is valid in **2Int**.

An $\mathcal{L}_{\mathbf{2Int}}$-formula A is dually valid in a model for **2Int** $\mathcal{M} = \langle I, \leq, v^{+}, v^{-} \rangle$ iff for every $w \in I$, $\mathcal{M}, w \models^{-} A$ (iff for every $w \in I$, $\mathcal{M}, w \models^{+} -A$); A is dually valid in **2Int** iff it is dually valid in every model for **2Int**. We write $\models^{d}_{\mathbf{2Int}} A$ to mean that A is valid in **2Int**.

Definition

Let $\Delta \cup \{A\}$ be a set of $\mathcal{L}_{\mathbf{2Int}}$-formulas. Δ entails A ($\Delta \models A$) iff for every model for **2Int** $\mathcal{M} = \langle I, \leq, v^+, v^- \rangle$ and every $u \in I$, it holds that if the truth of every element of Δ is supported by u, then the truth of A is supported by u.

Let $\Delta \cup \{A\}$ be a set of $\mathcal{L}_{\mathbf{2Int}}$-formulas. Δ dually entails A ($\Delta \models^d A$) iff for every model for **2Int** $\mathcal{M} = \langle I, \leq, v^+, v^- \rangle$ and every $u \in I$, it holds that if the falsity of every element of Δ is supported by u, then the falsity of A is supported by u.

intuitionistic negation	co-negation
$A, \neg A \models B$	$A, -A \not\models B$
$A, \neg A \not\models^d B$	$A, -A \models^d B$

Definition

Let $\Delta \cup \{A\}$ be a set of $\mathcal{L}_{\mathbf{2Int}}$-formulas. Δ entails A ($\Delta \models A$) iff for every model for **2Int** $\mathcal{M} = \langle I, \leq, v^+, v^- \rangle$ and every $u \in I$, it holds that if the truth of every element of Δ is supported by u, then the truth of A is supported by u.

Let $\Delta \cup \{A\}$ be a set of $\mathcal{L}_{\mathbf{2Int}}$-formulas. Δ dually entails A ($\Delta \models^d A$) iff for every model for **2Int** $\mathcal{M} = \langle I, \leq, v^+, v^- \rangle$ and every $u \in I$, it holds that if the falsity of every element of Δ is supported by u, then the falsity of A is supported by u.

intuitionistic negation	co-negation
$A, \neg A \models B$	$A, -A \not\models B$
$A, \neg A \not\models^d B$	$A, -A \models^d B$

Definition

Let $\Delta \cup \{A\}$ be a set of $\mathcal{L}_{\mathbf{2Int}}$-formulas. Δ entails A ($\Delta \models A$) iff for every model for **2Int** $\mathcal{M} = \langle I, \leq, v^+, v^- \rangle$ and every $u \in I$, it holds that if the truth of every element of Δ is supported by u, then the truth of A is supported by u.

Let $\Delta \cup \{A\}$ be a set of $\mathcal{L}_{\mathbf{2Int}}$-formulas. Δ dually entails A ($\Delta \models^d A$) iff for every model for **2Int** $\mathcal{M} = \langle I, \leq, v^+, v^- \rangle$ and every $u \in I$, it holds that if the falsity of every element of Δ is supported by u, then the falsity of A is supported by u.

intuitionistic negation	co-negation
$A, \neg A \models B$	$A, -A \not\models B$
$A, \neg A \not\models^d B$	$A, -A \models^d B$

Definition

Let $\Phi' = \{p' \mid p \in \Phi\}$. The translation τ from $\mathcal{L}_{\mathbf{2Int}}$ into $\mathcal{L}_{\mathbf{Int}}$ based on the set of atomic formulas $\Phi \cup \Phi'$ is defined as follows (some outermost brackets are omitted):

$$
\begin{array}{rcl@{\qquad}rcl}
\tau(p) & := & p & \tau(-p) & = & p' \\
\tau(\top) & := & \top & \tau(-\top) & := & \bot \\
\tau(\bot) & := & \bot & \tau(-\bot) & := & \top \\
\tau(A \wedge B) & := & \tau(A) \wedge \tau(B) & \tau(-(A \wedge B)) & := & \tau(-A) \vee \tau(-B) \\
\tau(A \vee B) & := & \tau(A) \vee \tau(B) & \tau(-(A \vee B)) & := & \tau(-A) \wedge \tau(-B) \\
\tau(A \rightarrow B) & := & \tau(A) \rightarrow \tau(B) & \tau(-(A \rightarrow B)) & := & \tau(A) \wedge \tau(-B) \\
\tau(A \prec B) & := & \tau(A) \wedge \tau(-B), & \tau(-(A \prec B)) & := & \tau(-B) \rightarrow \tau(-A) \\
\text{if } A \not\equiv \top & & & & &
\end{array}
$$

Proposition

The following equivalences are valid in **2Int**:

- $-\top \leftrightarrow \bot$
- $-\bot \leftrightarrow \top$
- $-\neg A \leftrightarrow A$
- $--A \leftrightarrow (-A \to \bot)$
- $-(A \wedge B) \leftrightarrow (-A \vee -B)$
- $-(A \vee B) \leftrightarrow (-A \wedge -B)$
- $-(A \to B) \leftrightarrow (A \wedge -B)$
- $-(A \prec B) \leftrightarrow (-B \to -A)$

An $\mathcal{L}_{\mathbf{2Int}}$-formula A is in negation normal form iff it contains occurrences of co-negation $-$ only as prefixes of atomic formulas.

Corollary

For every $\mathcal{L}_{\mathbf{2Int}}$-formula A, there is a formula A' such that $\models_{\mathbf{2Int}} A \leftrightarrow A'$ and A' is in negation normal form.

Proposition

The following equivalences are valid in **2Int***:*

- $-\top \leftrightarrow \bot$
- $-\bot \leftrightarrow \top$
- $-\neg A \leftrightarrow A$
- $- - A \leftrightarrow (-A \to \bot)$
- $-(A \wedge B) \leftrightarrow (-A \vee -B)$
- $-(A \vee B) \leftrightarrow (-A \wedge -B)$
- $-(A \to B) \leftrightarrow (A \wedge -B)$
- $-(A \prec B) \leftrightarrow (-B \to -A)$

An $\mathcal{L}_{\mathbf{2Int}}$-formula A is in negation normal form iff it contains occurrences of co-negation $-$ only as prefixes of atomic formulas.

Corollary

For every $\mathcal{L}_{\mathbf{2Int}}$-formula A, there is a formula A' such that $\models_{\mathbf{2Int}} A \leftrightarrow A'$ and A' is in negation normal form.

Lemma

Let τ be the mapping defined above and let $\mathcal{M}' = \langle I, \leq, v^+, v^- \rangle$ be a model for **2Int**. *Consider the language $\mathcal{L}_{\mathbf{Int}}$ based on the set of atomic formulas $\Phi \cup \Phi'$, and let $\mathcal{M}$ be the structure $\langle I, \leq, v \rangle$, where the mapping v from $\Phi \cup \Phi'$ into subsets of I is defined by requiring for every $w \in I$, every $p \in \Phi$: $w \in v(p)$ iff $w \in v^+(p)$; $w \in v(p')$ iff $w \in v^-(p)$. Clearly, $\mathcal{M}$ is a model for* **Int**. *For every $\mathcal{L}_{\mathbf{2Int}}$-formula A and every $w \in I$,*

- $\mathcal{M}', w \models^+ A$ *iff* $\mathcal{M}, w \models^+ \tau(A)$,
- $\mathcal{M}', w \models^- A$ *iff* $\mathcal{M}, w \models^+ \tau(-A)$.

Lemma

Let τ be the mapping defined above and let $\mathcal{M} = \langle I, \leq, v\rangle$ be a model for **Int**. *Consider the language $\mathcal{L}_{\mathbf{Int}}$ based on the set of atomic formulas $\Phi \cup \Phi'$, and let $\mathcal{M}'$ be the structure $\langle I, \leq, v^+, v^-\rangle$, where the mappings v^+, v^- from Φ into subsets of I are defined by requiring for every $w \in I$, every $p \in \Phi$: $w \in v(p)$ iff $w \in v^+(p)$; $w \in v(p')$ iff $w \in v^-(p)$. Clearly, $\mathcal{M}'$ is a model for* **2Int**. *For every $\mathcal{L}_{\mathbf{2Int}}$-formula A and every $w \in I$,*

- $\mathcal{M}', w \models^+ A$ *iff* $\mathcal{M}, w \models^+ \tau(A)$,
- $\mathcal{M}', w \models^- A$ *iff* $\mathcal{M}, w \models^+ \tau(-A)$.

Theorem

For every $\mathcal{L}_{\mathbf{2Int}}$-formula A, $\models_{\mathbf{2Int}} A$ iff $\models_{\mathbf{Int}} \tau(A)$.

Lemma

Let τ be the mapping defined above and let $\mathcal{M} = \langle I, \leq, v\rangle$ be a model for **Int**. *Consider the language $\mathcal{L}_{\mathbf{Int}}$ based on the set of atomic formulas $\Phi \cup \Phi'$, and let $\mathcal{M}'$ be the structure $\langle I, \leq, v^+, v^-\rangle$, where the mappings v^+, v^- from Φ into subsets of I are defined by requiring for every $w \in I$, every $p \in \Phi$: $w \in v(p)$ iff $w \in v^+(p)$; $w \in v(p')$ iff $w \in v^-(p)$. Clearly, $\mathcal{M}'$ is a model for* **2Int**. *For every $\mathcal{L}_{\mathbf{2Int}}$-formula A and every $w \in I$,*

- $\mathcal{M}', w \models^+ A$ *iff* $\mathcal{M}, w \models^+ \tau(A)$,
- $\mathcal{M}', w \models^- A$ *iff* $\mathcal{M}, w \models^+ \tau(-A)$.

Theorem

For every $\mathcal{L}_{\mathbf{2Int}}$-formula A, $\models_{\mathbf{2Int}} A$ iff $\models_{\mathbf{Int}} \tau(A)$.

Conservativity

It can be shown that **2Int** is a conservative extension of **Int** with respect to validity and of **DualInt** with respect to dual validity.

Conservativity over **Int** with respect to validity follows immediately by the previous theorem and the observation that for every $\mathcal{L}_{\mathbf{Int}}$-formula A based on the set of atomic formulas Φ, $A = \tau(A)$.

Theorem

For every $\mathcal{L}_{\mathbf{Int}}$-formula A, $\models_{\mathbf{Int}} A$ iff $\models_{\mathbf{2Int}} A$.

Conservativity

It can be shown that **2Int** is a conservative extension of **Int** with respect to validity and of **DualInt** with respect to dual validity.

Conservativity over **Int** with respect to validity follows immediately by the previous theorem and the observation that for every $\mathcal{L}_{\mathbf{Int}}$-formula A based on the set of atomic formulas Φ, $A = \tau(A)$.

Theorem

For every $\mathcal{L}_{\mathbf{Int}}$-formula A, $\models_{\mathbf{Int}} A$ iff $\models_{\mathbf{2Int}} A$.

Conservativity

It can be shown that **2Int** is a conservative extension of **Int** with respect to validity and of **DualInt** with respect to dual validity.

Conservativity over **Int** with respect to validity follows immediately by the previous theorem and the observation that for every $\mathcal{L}_{\mathbf{Int}}$-formula A based on the set of atomic formulas Φ, $A = \tau(A)$.

Theorem

For every $\mathcal{L}_{\mathbf{Int}}$-formula A, $\models_{\mathbf{Int}} A$ iff $\models_{\mathbf{2Int}} A$.

Conservativity

Definition

Let $\Phi' = \{p' \mid p \in \Phi\}$. The translation ζ from $\mathcal{L}_{\mathbf{2Int}}$ into $\mathcal{L}_{\mathbf{DualInt}}$ based on the set of atomic formulas $\Phi \cup \Phi'$ is defined as follows (some outermost brackets are omitted):

$$
\begin{array}{rcl}
\zeta(p) & := & p \\
\zeta(\top) & := & \top \\
\zeta(\bot) & := & \bot \\
\zeta(A \wedge B) & := & \zeta(A) \wedge \zeta(B) \\
\zeta(A \vee B) & := & \zeta(A) \vee \zeta(B) \\
\zeta(A \prec B) & := & \zeta(A) \prec \zeta(B) \\
\zeta(A \to B) & := & \zeta(\neg A) \vee \zeta(B), \text{ if } B \not\equiv \bot
\end{array}
\qquad
\begin{array}{rcl}
\zeta(\neg p) & = & p' \\
\zeta(\neg\top) & := & \bot \\
\zeta(\neg\bot) & := & \top \\
\zeta(\neg(A \wedge B)) & := & \zeta(\neg A) \vee \zeta(\neg B) \\
\zeta(\neg(A \vee B)) & := & \zeta(\neg A) \wedge \zeta(\neg B) \\
\zeta(\neg(A \prec B)) & := & \zeta(\neg A) \vee \zeta(B) \\
\zeta(\neg(A \to B)) & := & \zeta(\neg B) \prec \zeta(\neg A)
\end{array}
$$

Conservativity

Definition

Let $\Phi' = \{p' \mid p \in \Phi\}$. The translation ζ from $\mathcal{L}_{\mathbf{2Int}}$ into $\mathcal{L}_{\mathbf{DualInt}}$ based on the set of atomic formulas $\Phi \cup \Phi'$ is defined as follows (some outermost brackets are omitted):

$$
\begin{array}{rcl}
\zeta(p) & := & p \\
\zeta(\top) & := & \top \\
\zeta(\bot) & := & \bot \\
\zeta(A \wedge B) & := & \zeta(A) \wedge \zeta(B) \\
\zeta(A \vee B) & := & \zeta(A) \vee \zeta(B) \\
\zeta(A \prec B) & := & \zeta(A) \prec \zeta(B) \\
\zeta(A \rightarrow B) & := & \zeta(\neg A) \vee \zeta(B), \text{ if } B \not\equiv \bot
\end{array}
\qquad
\begin{array}{rcl}
\zeta(\neg p) & = & p' \\
\zeta(\neg \top) & := & \bot \\
\zeta(\neg \bot) & := & \top \\
\zeta(\neg(A \wedge B)) & := & \zeta(\neg A) \vee \zeta(\neg B) \\
\zeta(\neg(A \vee B)) & := & \zeta(\neg A) \wedge \zeta(\neg B) \\
\zeta(\neg(A \prec B)) & := & \zeta(\neg A) \vee \zeta(B) \\
\zeta(\neg(A \rightarrow B)) & := & \zeta(\neg B) \prec \zeta(\neg A)
\end{array}
$$

Conservativity

Theorem

For every $\mathcal{L}_{\mathbf{2Int}}$-formula A, $\models^{d}_{\mathbf{2Int}} A$ iff $\models^{d}_{\mathbf{DualInt}} \zeta(A)$.

Since for every $\mathcal{L}_{\mathbf{DualInt}}$-formula A based on Φ, $A = \zeta(A)$, we obtain the following conservativity result.

Theorem

For every $\mathcal{L}_{\mathbf{DualInt}}$-formula A, $\models^{d}_{\mathbf{DualInt}} A$ iff $\models^{d}_{\mathbf{2Int}} A$.

Conservativity

Theorem

For every $\mathcal{L}_{\mathbf{2Int}}$-formula A, $\models^{d}_{\mathbf{2Int}} A$ iff $\models^{d}_{\mathbf{DualInt}} \zeta(A)$.

Since for every $\mathcal{L}_{\mathbf{DualInt}}$-formula A based on Φ, $A = \zeta(A)$, we obtain the following conservativity result.

Theorem

For every $\mathcal{L}_{\mathbf{DualInt}}$-formula A, $\models^{d}_{\mathbf{DualInt}} A$ iff $\models^{d}_{\mathbf{2Int}} A$.

Conservativity

Theorem

For every $\mathcal{L}_{\mathbf{2Int}}$-formula A, $\models^d_{\mathbf{2Int}} A$ iff $\models^d_{\mathbf{DualInt}} \zeta(A)$.

Since for every $\mathcal{L}_{\mathbf{DualInt}}$-formula A based on Φ, $A = \zeta(A)$, we obtain the following conservativity result.

Theorem

For every $\mathcal{L}_{\mathbf{DualInt}}$-formula A, $\models^d_{\mathbf{DualInt}} A$ iff $\models^d_{\mathbf{2Int}} A$.

Lemma

Let Δ, Γ *be any finite sets of* $\mathcal{L}_{\mathbf{2Int}}$*-formulas. Let* $-\Gamma := \{-A \mid A \in \Gamma\}$. *If* $\Gamma = \varnothing$, *then* $-\Gamma := -\bot$. *Let* $\bigwedge\{A_1, \ldots, A_n\} := (A_1 \wedge (A_2 \wedge (\ldots (A_{n-1} \wedge A_n) \ldots)))$, *and let* $\bigwedge \varnothing := \top$. *Let* A *be any* $\mathcal{L}_{\mathbf{2Int}}$*-formula.*

- *If* $\mathrm{P}(\Pi, A, (\Delta; \Gamma))$, *then* $\models_{\mathbf{2Int}} (\bigwedge \Delta \wedge \bigwedge -\Gamma) \rightarrow A$.
- *If* $\mathrm{DP}(\Pi, A, (\Delta; \Gamma))$, *then* $\models_{\mathbf{2Int}} (\bigwedge \Delta \wedge \bigwedge -\Gamma) \rightarrow -A$.

Lemma

Let A be any $\mathcal{L}_{\mathbf{2Int}}$-formula A, let Γ be any finite set of $\mathcal{L}_{\mathbf{2Int}}$-formulas, let $\tau(\Gamma) := \{\tau(A) \mid A \in \Gamma\}$ if $\Gamma \neq \varnothing$, and let $\tau(\Gamma) := \top$ if $\Gamma = \varnothing$. If $\tau(\Gamma) \vdash_{\mathrm{NJ}} \tau(A)$, then there exists a derivation Π in $\mathrm{N2Int}$ with $\mathsf{P}(\Pi, A, (\Gamma; \varnothing))$.

Theorem

Let A be any $\mathcal{L}_{\mathbf{2Int}}$-formula A. There is a derivation Π with $\mathsf{Pr}(\Pi, A, (\varnothing; \varnothing))$ iff $\models_{\mathbf{2Int}} A$.

Lemma

Let A be any $\mathcal{L}_{\mathbf{2Int}}$-formula A, let Γ be any finite set of $\mathcal{L}_{\mathbf{2Int}}$-formulas, let $\tau(\Gamma) := \{\tau(A) \mid A \in \Gamma\}$ if $\Gamma \neq \varnothing$, and let $\tau(\Gamma) := \top$ if $\Gamma = \varnothing$. If $\tau(\Gamma) \vdash_{\mathrm{NJ}} \tau(A)$, then there exists a derivation Π in $\mathrm{N2Int}$ with $\mathsf{P}(\Pi, A, (\Gamma; \varnothing))$.

Theorem

Let A be any $\mathcal{L}_{\mathbf{2Int}}$-formula A. There is a derivation Π with $\mathsf{Pr}(\Pi, A, (\varnothing; \varnothing))$ iff $\models_{\mathbf{2Int}} A$.

The logic **2Int** bears a close resemblance to a certain extension of David Nelson's four-valued constructive logic **N4**, namely the system $\mathbf{N4}^{\perp}$ introduced by Sergei Odintsov (2008).

The language $\mathcal{L}_{\mathbf{N4}^{\perp}}$ of $\mathbf{N4}^{\perp}$ is defined in Backus–Naur form as follows:

$$A ::= \; p \mid \perp \mid \top \mid \sim \mid (A \wedge A) \mid (A \vee A) \mid (A \rightarrow A)$$

where $\sim$ is a primitive operation of so-called strong negation.

The following equivalences are valid in $\mathbf{N4}^{\perp}$:

- $\sim\top \leftrightarrow \perp$
- $\sim\perp \leftrightarrow \top$
- $\sim(A \wedge B) \leftrightarrow (\sim A \vee \sim B)$
- $\sim(A \vee B) \leftrightarrow (\sim A \wedge \sim B)$
- $\sim(A \rightarrow B) \leftrightarrow (A \wedge \sim B)$.

The logic **2Int** bears a close resemblance to a certain extension of David Nelson's four-valued constructive logic **N4**, namely the system $\mathbf{N4}^{\perp}$ introduced by Sergei Odintsov (2008).

The language $\mathcal{L}_{\mathbf{N4}^{\perp}}$ of $\mathbf{N4}^{\perp}$ is defined in Backus–Naur form as follows:

$$A ::= \quad p \mid \perp \mid \top \mid \sim \mid (A \wedge A) \mid (A \vee A) \mid (A \rightarrow A)$$

where $\sim$ is a primitive operation of so-called strong negation.

The following equivalences are valid in $\mathbf{N4}^{\perp}$:

- $\sim\top \leftrightarrow \perp$
- $\sim\perp \leftrightarrow \top$
- $\sim(A \wedge B) \leftrightarrow (\sim A \vee \sim B)$
- $\sim(A \vee B) \leftrightarrow (\sim A \wedge \sim B)$
- $\sim(A \rightarrow B) \leftrightarrow (A \wedge \sim B)$.

The logic **2Int** bears a close resemblance to a certain extension of David Nelson's four-valued constructive logic **N4**, namely the system $\mathbf{N4}^{\perp}$ introduced by Sergei Odintsov (2008).

The language $\mathcal{L}_{\mathbf{N4}^{\perp}}$ of $\mathbf{N4}^{\perp}$ is defined in Backus–Naur form as follows:

$$A ::= \quad p \mid \perp \mid \top \mid \sim \mid (A \wedge A) \mid (A \vee A) \mid (A \to A)$$

where $\sim$ is a primitive operation of so-called strong negation.

The following equivalences are valid in $\mathbf{N4}^{\perp}$:

- $\sim\top \leftrightarrow \perp$
- $\sim\perp \leftrightarrow \top$
- $\sim(A \wedge B) \leftrightarrow (\sim A \vee \sim B)$
- $\sim(A \vee B) \leftrightarrow (\sim A \wedge \sim B)$
- $\sim(A \to B) \leftrightarrow (A \wedge \sim B)$.

Whereas $\mathbf{N4}^{\perp}$ is paraconsistent wrt $\sim$ and entailment, **2Int** is paraconsistent wrt $-$: $(A \wedge -A) \rightarrow B$ is not a valid schema.

The system $\mathbf{N4}^{\perp}$ satisfies the constructible falsity property wrt strong negation, and the system **2Int** satisfies constructible falsity wrt to co-negation: for all **2Int**-formulas A and B,

$$\models_{\mathbf{2Int}} -(A \wedge B) \text{ iff } (\models_{\mathbf{2Int}} -A \text{ or } \models_{\mathbf{2Int}} -B).$$

In $\mathbf{N4}^{\perp}$ the contraposition rule fails to be valid wrt strong negation, and in **2Int** it fails to be valid wrt co-negation, i.e.,

$$\frac{A \rightarrow B}{-B \rightarrow -A}$$

is not valid in **2Int**.

Whereas **N4**$^\perp$ is paraconsistent wrt $\sim$ and entailment, **2Int** is paraconsistent wrt $-$: $(A \wedge -A) \to B$ is not a valid schema.

The system **N4**$^\perp$ satisfies the constructible falsity property wrt strong negation, and the system **2Int** satisfies constructible falsity wrt to co-negation: for all **2Int**-formulas A and B,

$$\models_{\mathbf{2Int}} -(A \wedge B) \text{ iff } (\models_{\mathbf{2Int}} -A \text{ or } \models_{\mathbf{2Int}} -B).$$

In **N4**$^\perp$ the contraposition rule fails to be valid wrt strong negation, and in **2Int** it fails to be valid wrt co-negation, i.e.,

$$\frac{A \to B}{-B \to -A}$$

is not valid in **2Int**.

Whereas $\mathbf{N4}^{\perp}$ is paraconsistent wrt $\sim$ and entailment, **2Int** is paraconsistent wrt $-$: $(A \wedge -A) \rightarrow B$ is not a valid schema.

The system $\mathbf{N4}^{\perp}$ satisfies the constructible falsity property wrt strong negation, and the system **2Int** satisfies constructible falsity wrt to co-negation: for all **2Int**-formulas A and B,

$$\models_{\mathbf{2Int}} -(A \wedge B) \text{ iff } (\models_{\mathbf{2Int}} -A \text{ or } \models_{\mathbf{2Int}} -B).$$

In $\mathbf{N4}^{\perp}$ the contraposition rule fails to be valid wrt strong negation, and in **2Int** it fails to be valid wrt co-negation, i.e.,

$$\frac{A \rightarrow B}{-B \rightarrow -A}$$

is not valid in **2Int**.

Moreover, in both logics the replacement rule

$$\frac{A \leftrightarrow B}{C(A) \leftrightarrow C(B)}$$

is not valid.

In Nelson's logics the replacement rule is valid for provable strict equivalences. Let $(A \leftrightarrow_s B) := (A \leftrightarrow B) \wedge (\sim A \leftrightarrow \sim B)$, then

$$\frac{A \leftrightarrow_s B}{C(A) \leftrightarrow_s C(B)}$$

is a valid rule.

Moreover, in both logics the replacement rule

$$\frac{A \leftrightarrow B}{C(A) \leftrightarrow C(B)}$$

is not valid.

In Nelson's logics the replacement rule is valid for provable strict equivalences. Let $(A \leftrightarrow_s B) := (A \leftrightarrow B) \wedge ({\sim}A \leftrightarrow {\sim}B)$, then

$$\frac{A \leftrightarrow_s B}{C(A) \leftrightarrow_s C(B)}$$

is a valid rule.

For **2Int** we may define strong equivalence with respect to co-negation: $(A \leftrightarrow_s B) := (A \leftrightarrow B) \wedge (-A \leftrightarrow -B)$.

Proposition

The replacement rule

$$\frac{A \leftrightarrow_s B}{C(A) \leftrightarrow_s C(B)}$$

is valid in **2Int**.

For **2Int** we may define strong equivalence with respect to co-negation: $(A \leftrightarrow_s B) := (A \leftrightarrow B) \wedge (-A \leftrightarrow -B)$.

Proposition

The replacement rule

$$\frac{A \leftrightarrow_s B}{C(A) \leftrightarrow_s C(B)}$$

is valid in **2Int**.

Every model for **2Int** is a model for **N4**$^\perp$ and vice versa. A central difference between **N4**$^\perp$ and **2Int** wrt support of truth and support of falsity conditions is that whereas in **2Int** we have $(\mathcal{M}, w \models^- -A$ iff for every $w' \geq w : \mathcal{M}, w' \not\models^- A)$, in **N4**$^\perp$ we have we have $(\mathcal{M}, w \models^- {\sim}A$ iff $\mathcal{M}, w \models^+ A)$.

In **N4**$^\perp$ the double-negation laws ${\sim}{\sim}A \to A$ and $A \to {\sim}{\sim}A$ are valid, whereas in **2Int** the equivalence $- - A \leftrightarrow (-A \to \bot)$ is valid, and with respect to co-negation, *neither* double-negation introduction $A \to - - A$ *nor* double negation elimination $- - A \to A$ is a valid schema.

Every model for **2Int** is a model for $\mathbf{N4}^{\perp}$ and vice versa. A central difference between $\mathbf{N4}^{\perp}$ and **2Int** wrt support of truth and support of falsity conditions is that whereas in **2Int** we have $(\mathcal{M}, w \models^{-} -A$ iff for every $w' \geq w : \mathcal{M}, w' \not\models^{-} A)$, in $\mathbf{N4}^{\perp}$ we have we have $(\mathcal{M}, w \models^{-} {\sim}A$ iff $\mathcal{M}, w \models^{+} A)$.

In $\mathbf{N4}^{\perp}$ the double-negation laws ${\sim}{\sim}A \to A$ and $A \to {\sim}{\sim}A$ are valid, whereas in **2Int** the equivalence $- - A \leftrightarrow (-A \to \bot)$ is valid, and with respect to co-negation, *neither* double-negation introduction $A \to - - A$ *nor* double negation elimination $- - A \to A$ is a valid schema.

Thus, in Nelson's logics with strong negation, strong negation internalizes support of truth from a falsificationist perspective ($\mathcal{M}, w \models^- \sim A$ iff $\mathcal{M}, w \models^+ A$) *and* support of falsity from a verificationist point of view ($\mathcal{M}, w \models^+ \sim A$ iff $\mathcal{M}, w \models^- A$).

In **2Int**, there is a division of labour. Whereas intuitionistic negation internalizes support of truth form a falsificationist standpoint ($\mathcal{M}, w \models^- \neg A$ iff $\mathcal{M}, w \models^+ A$), co-negation internalizes support of falsity from a verificationist perspective ($\mathcal{M}, w \models^+ -A$ iff $\mathcal{M}, w \models^- A$).

Thus, in Nelson's logics with strong negation, strong negation internalizes support of truth from a falsificationist perspective ($\mathcal{M}, w \models^- \sim A$ iff $\mathcal{M}, w \models^+ A$) *and* support of falsity from a verificationist point of view ($\mathcal{M}, w \models^+ \sim A$ iff $\mathcal{M}, w \models^- A$).

In **2Int**, there is a division of labour. Whereas intuitionistic negation internalizes support of truth form a falsificationist standpoint ($\mathcal{M}, w \models^- \neg A$ iff $\mathcal{M}, w \models^+ A$), co-negation internalizes support of falsity from a verificationist perspective ($\mathcal{M}, w \models^+ -A$ iff $\mathcal{M}, w \models^- A$).

Thank you for your attention.