A Propositional Constructive Logic of Evidence

Juan C. Agudelo-Agudelo

Instituto de Matemáticas
Universidad de Antioquia, Medellín, Colombia

XX SLALM - 2024

Outline

1. Introduction
2. Natural deduction system for LET_C
3. Adding λ-terms to LET_C
4. Realisability interpretation for LET_C
5. Constructible features of LET_C

Introduction

Carnielli, W. and Rodrigues, A (2019). An epistemic approach to paraconsistency: a logic of evidence and truth. *Synthese*, 196:3789–3813.

- Introduce the Basic Logic of Evidence BLE (equivalent to Nelson's paraconsistent logic N4).
- BLE is extended to the Logic of Evidence and Truth LET_J, by adding a primitive classicality operator $\circ$, which allows to recover simultaneously the explosiveness of contradictions and the excluded middle for some propositions.

UNIVERSIDAD DE ANTIOQUIA 1803

Introduction

Carnielli, W. and Rodrigues, A (2019). An epistemic approach to paraconsistency: a logic of evidence and truth. *Synthese*, 196:3789–3813.

- Introduce the Basic Logic of Evidence BLE (equivalent to Nelson's paraconsistent logic N4).
- BLE is extended to the Logic of Evidence and Truth $\mathsf{LET_J}$, by adding a primitive classicality operator $\circ$, which allows to recover simultaneously the explosiveness of contradictions and the excluded middle for some propositions.

Rodrigues, A., Bueno-Soler,J., and Carnielli, W. (2021). Measuring evidence: a probabilistic approach to an extension of Belnap-Dunn logic. *Synthese*, 198(Suppl 22):5451–5480.

- Presents another Logic of Evidence and Truth $\mathsf{LET_F}$, based on Belnap-Dunn logic, with classicality and non-classicality operators.
- Provides a probabilistic semantics for $\mathsf{LET_F}$.

UNIVERSIDAD DE ANTIOQUIA 1803

Introduction

Rodrigues, A., Coniglio, M. E., Antunes, H., Bueno-Soler, J., and Carnielli, W. (2023). Paraconsistence, evidence and abduction. In Magnani, L., Editor, *Handbook of Abductive Cognition*, pages 313–350. Springer.

- Presents a summary of work on logics of evidence and truth.
- Presents a new Logic of Evidence and Truth $\mathsf{LET_K}$ and its first-order extension $\mathsf{QLET_K}$, and an application of these logics to the problem of abduction.

Introduction

Rodrigues, A., Coniglio, M. E., Antunes, H., Bueno-Soler, J., and Carnielli, W. (2023). Paraconsistence, evidence and abduction. In Magnani, L., Editor, *Handbook of Abductive Cognition*, pages 313–350. Springer.

- Presents a summary of work on logics of evidence and truth.
- Presents a new Logic of Evidence and Truth LET_K and its first-order extension QLET_K, and an application of these logics to the problem of abduction.

A new (Constructive) Logic of Evidence and Truth LET_C is here proposed.

- LET_C is based on N4* (an extension of N4 with $\bot$, $\top$ and $\prec$).
- In LET_C, explosiveness of contradictions and excluded middle can be independently recovered.
- In order to explicitly formalise evidence, λ-terms are added to LET_C obtaining the type system $\mathsf{LET}_\mathsf{C}^\lambda$.
- A realisabiliy interpretation is provided for LET_C.
- Some constructive features of LET_C are highlighted.

UNIVERSIDAD DE ANTIOQUIA 1803

Natural deduction system for LET_C I

Natural deduction system for N4*: rules of propositional intuitionistic logic plus the following, where $\sim$ is Nelson's constructive negation and $\prec$ is co-implication ($B \prec A$ reads A co-implies B).

$$\frac{}{\top}$$

$$\frac{\sim A \qquad B}{B \prec A} \qquad\qquad \frac{B \prec A}{\sim A} \qquad \frac{B \prec A}{B}$$

$$\frac{}{\sim \bot} \qquad\qquad \frac{\sim \top}{A}$$

$$\frac{\sim A}{\sim(A \wedge B)} \qquad \frac{\sim B}{\sim(A \wedge B)} \qquad\qquad \frac{\sim(A \wedge B) \qquad \begin{matrix}[\sim A] \\ \vdots \\ C\end{matrix} \qquad \begin{matrix}[\sim B] \\ \vdots \\ C\end{matrix}}{C}$$

Natural deduction system for LET_C

$$\frac{\sim A \qquad \sim B}{\sim(A \vee B)} \qquad \frac{\sim(A \vee B)}{\sim A} \qquad \frac{\sim(A \vee B)}{\sim B}$$

$$\frac{A \qquad \sim B}{\sim(A \to B)} \qquad \frac{\sim(A \to B)}{A} \qquad \frac{\sim(A \to B)}{\sim B}$$

$$\frac{\begin{array}{c}[\sim A] \\ \vdots \\ \sim B\end{array}}{\sim(B \prec A)} \qquad \frac{\sim A \qquad \sim(B \prec A)}{\sim B}$$

$$\frac{A}{\sim\sim A} \qquad \frac{\sim\sim A}{A}$$

Natural deduction system for LET_C

In $N4^{\star}$, intuitionistic negation $\neg$ is defined by $\neg A \stackrel{\text{def}}{=} A \to \bot$ and co-negation $\sim$ is defined by $\sim A \stackrel{\text{def}}{=} \top \prec A$.

UNIVERSIDAD DE ANTIOQUIA 1803

Natural deduction system for LET_C

In N4*, intuitionistic negation $\neg$ is defined by $\neg A \stackrel{\text{def}}{=} A \to \bot$ and co-negation $\neg\!\!\!\!-$ is defined by $\neg\!\!\!\!- A \stackrel{\text{def}}{=} \top \prec A$.

An equivalence operator $\leftrightarrow$ is defined in N4* as usual, but substitution by equivalent formulae is not valid.

Natural deduction system for LET_C

In $\text{N4}^\star$, intuitionistic negation $\neg$ is defined by $\neg A \stackrel{\text{def}}{=} A \to \bot$ and co-negation $\lnot$ is defined by $\lnot A \stackrel{\text{def}}{=} \top \prec A$.

An equivalence operator $\leftrightarrow$ is defined in $\text{N4}^\star$ as usual, but substitution by equivalent formulae is not valid.

A strong equivalence operator $\Leftrightarrow$ is defined by $A \Leftrightarrow B \stackrel{\text{def}}{=} (A \leftrightarrow B) \wedge (\sim A \leftrightarrow \sim B)$, and substitution by strongly equivalent formulae is valid.

Natural deduction system for LET_C

Theorem 2.1

1. $\vdash \neg(A \wedge \sim A) \leftrightarrow (A \vee \sim A)$ *(but* $\nvdash \neg(A \wedge \sim A) \Leftrightarrow (A \vee \sim A)$*).*
2. $\vdash \neg(A \vee \sim A) \leftrightarrow (A \wedge \sim A)$ *(but* $\nvdash \neg(A \vee \sim A) \Leftrightarrow (A \wedge \sim A)$*).*

UNIVERSIDAD DE ANTIOQUIA 1803

Natural deduction system for LET$_C$

Theorem 2.1

1. $\vdash \neg\!\!\!\!-(A\wedge \sim A) \leftrightarrow (A\vee \sim A)$ *(but* $\nvdash \neg\!\!\!\!-(A\wedge \sim A) \Leftrightarrow (A\vee \sim A)$*).*
2. $\vdash \neg\!\!\!\!-(A\vee \sim A) \leftrightarrow (A\wedge \sim A)$ *(but* $\nvdash \neg\!\!\!\!-(A\vee \sim A) \Leftrightarrow (A\wedge \sim A)$*).*

Definition 2.2

The logic LET$_C$ is the result of adding to N4* the following defined connectives.

$$\circ A \stackrel{\text{def}}{=} \neg(A\wedge \sim A),$$

$$\star A \stackrel{\text{def}}{=} \neg\!\!\!\!-(A\wedge \sim A) \text{ (equivalently, } \star A \stackrel{\text{def}}{=} \sim\neg(A\vee \sim A)),$$

$$\bullet A \stackrel{\text{def}}{=} \neg\!\!\!\!-(A\vee \sim A) \text{ (equivalently, } \bullet A \stackrel{\text{def}}{=} \sim\circ A),$$

$$\bigstar A \stackrel{\text{def}}{=} \neg(A\vee \sim A) \text{ (equivalently, } \bigstar A \stackrel{\text{def}}{=} \sim\star A).$$

Adding λ-terms to $\mathsf{LET_C}$

A BHK-style interpretation for $\mathsf{LET_C}$ is defined by the following clauses.

- Evidence for accepting $A \wedge B$ is a pair (e_1, e_2) where e_1 is evidence for accepting A and e_2 is evidence for accepting B.
- Evidence for accepting $A \vee B$ is a pair (e_1, e_2) where $e_1 = 0$ and e_2 is evidence for accepting A, or $e_1 = 1$ and e_2 is evidence for accepting B.
- Evidence for accepting $A \to B$ is a method that converts evidence for accepting A into evidence for accepting B.
- Evidence for accepting $B \prec A$ is a pair (e_1, e_2) where e_1 is evidence for rejecting A and e_2 is evidence for accepting B.
- Evidence for accepting $\sim A$ is evidence for rejecting A.
- There is no evidence for accepting $\bot$.
- $\varnothing$ is evidence for accepting $\top$.

Adding λ-terms to LET_C

- Evidence for rejecting $A \wedge B$ is a pair (e_1, e_2) where $e_1 = 0$ and e_2 is evidence for rejecting A, or $e_1 = 1$ and e_2 is evidence for rejecting B.
- Evidence for rejecting $A \vee B$ is a pair (e_1, e_2) where e_1 is evidence for rejecting A and e_2 is evidence for rejecting B.
- Evidence for rejecting $A \to B$ is a pair (e_1, e_2) where e_1 is evidence for accepting A and e_2 is evidence for rejecting B.
- Evidence for rejecting $B \prec A$ is a method that converts evidence for rejecting A into evidence for rejecting B.
- Evidence for rejecting $\sim A$ is evidence for accepting A.
- $\varnothing$ is evidence for rejecting $\bot$.
- There is no evidence for rejecting $\top$.

Adding λ-terms to $\mathsf{LET_C}$

The Type system $\mathsf{LET_C^\lambda}$ is defined by the following rules:

$$\frac{}{\varnothing : \top}$$

$$\frac{t : \bot}{\mathcal{E}(t) : A}$$

$$\frac{r : A \qquad s : B}{(r, s) : A \wedge B}$$

$$\frac{t : A \wedge B}{\pi_1(t) : A} \qquad \frac{t : A \wedge B}{\pi_2(t) : B}$$

$$\frac{t : A}{\mathsf{in}_1(t) : A \vee B} \qquad \frac{t : B}{\mathsf{in}_2(t) : A \vee B}$$

$$\frac{t : A \vee B \qquad \begin{matrix}[x : A] \\ \vdots \\ r : C\end{matrix} \qquad \begin{matrix}[y : B] \\ \vdots \\ s : C\end{matrix}}{\mathsf{case}\ t\ \mathsf{of}\ [x]r\ \mathsf{or}\ [y]s : C}$$

$$\frac{\begin{matrix}[x : A] \\ \vdots \\ t : B\end{matrix}}{\lambda x.t : A \to B}$$

$$\frac{r : A \to B \qquad s : A}{\mathsf{ap}(r, s) : B}$$

$$\frac{r : \sim A \qquad s : B}{\langle r, s \rangle : B \prec A}$$

$$\frac{t : B \prec A}{\pi_1^*(t) : \sim A} \qquad \frac{t : B \prec A}{\pi_2^*(t) : B}$$

Adding λ-terms to LET_C

$$\frac{}{\varnothing : {\sim}\bot}$$

$$\frac{t : {\sim}\top}{\mathcal{E}(t) : A}$$

$$\frac{t : {\sim}A}{\mathrm{in}_1(t) : {\sim}(A \wedge B)} \qquad \frac{t : {\sim}B}{\mathrm{in}_2(t) : {\sim}(A \wedge B)}$$

$$\frac{t : {\sim}(A \wedge B) \qquad \begin{matrix}[x : {\sim}A] \\ \vdots \\ r : C\end{matrix} \qquad \begin{matrix}[y : {\sim}B] \\ \vdots \\ s : C\end{matrix}}{\text{case } t \text{ of } [x]r \text{ or } [y]s : C}$$

$$\frac{r : {\sim}A \qquad s : {\sim}B}{(r, s) : {\sim}(A \vee B)}$$

$$\frac{t : {\sim}(A \vee B)}{\pi_1(t) : {\sim}A} \qquad \frac{t : {\sim}(A \vee B)}{\pi_2(t) : {\sim}B}$$

$$\frac{r : A \qquad s : {\sim}B}{\langle r, s \rangle : {\sim}(A \to B)}$$

$$\frac{t : {\sim}(A \to B)}{\pi_1^*(t) : A} \qquad \frac{t : {\sim}(A \to B)}{\pi_2^*(t) : {\sim}B}$$

$$\frac{\begin{matrix}[x : {\sim}A] \\ \vdots \\ t : {\sim}B\end{matrix}}{\lambda x.t : {\sim}(B \prec A)}$$

$$\frac{r : {\sim}(B \prec A) \qquad s : {\sim}A}{\mathrm{ap}(r, s) : {\sim}B}$$

$$\frac{t : A}{\mathrm{id}(t) : {\sim}{\sim}A}$$

$$\frac{t : {\sim}{\sim}A}{\mathrm{id}^{-1}(t) : A}$$

Adding λ-terms to LET_C

The β-reduction relation on λ-terms is the less compatible relation satisfying:

$$
\begin{aligned}
\mathsf{ap}(\lambda x.r, s) &\to_\beta r[x := s] \\
\pi_1((r, s)) &\to_\beta r \\
\pi_2((r, s)) &\to_\beta s \\
\pi_1^*(\langle r, s\rangle) &\to_\beta r \\
\pi_2^*(\langle r, s\rangle) &\to_\beta s \\
\mathsf{case}\ \mathsf{in}_1(t)\ \mathsf{of}\ [x]r\ \mathsf{or}\ [y]s &\to_\beta r[x := t] \\
\mathsf{case}\ \mathsf{in}_2(t)\ \mathsf{of}\ [x]r\ \mathsf{or}\ [y]s &\to_\beta s[y := t] \\
\mathsf{id}(\mathsf{id}^{-1}(t)) &\to_\beta t \\
\mathsf{id}^{-1}(\mathsf{id}(t)) &\to_\beta t
\end{aligned}
$$

Adding λ-terms to $\mathsf{LET_C}$

Theorem 3.1

$B_1, \ldots, B_n \vdash A$ *in* $\mathsf{LET_C}$ *iff there is* $t \in \Lambda$ *such that* $x_1 : B_1, \ldots, x_n : B_n \vdash t : A$ *in* $\mathsf{LET}^{\lambda}_{\mathsf{C}}$.

UNIVERSIDAD DE ANTIOQUIA 1803

Adding λ-terms to LET_C

Theorem 3.1

$B_1, \ldots, B_n \vdash A$ in LET_C iff there is $t \in \Lambda$ such that $x_1 : B_1, \ldots, x_n : B_n \vdash t : A$ in $\text{LET}^{\lambda}_{\text{C}}$.

Lemma 3.2 (Free Variables Lemma)

1. *If $\Gamma \vdash t : A$, then $FV(t) \subseteq Dom(\Gamma)$.*
2. *If $\Gamma \vdash t : A$, then $\Gamma \upharpoonright FV(t) \vdash t : A$.*

Adding λ-terms to $\mathsf{LET_C}$

Theorem 3.1

$B_1, \ldots, B_n \vdash A$ in $\mathsf{LET_C}$ iff there is $t \in \Lambda$ such that $x_1 : B_1, \ldots, x_n : B_n \vdash t : A$ in $\mathsf{LET}_\mathsf{C}^\lambda$.

Lemma 3.2 (Free Variables Lemma)

1. *If $\Gamma \vdash t : A$, then $FV(t) \subseteq Dom(\Gamma)$.*
2. *If $\Gamma \vdash t : A$, then $\Gamma \restriction FV(t) \vdash t : A$.*

Lemma 3.3 (Uniqueness of Types)

If $\Gamma \vdash t : A$ and $\Gamma \vdash t : B$, then $\vdash A \Leftrightarrow B$.

UNIVERSIDAD DE ANTIOQUIA 1803

Adding λ-terms to LET_C

Lemma 3.4 (Subject Reduction)

If $\Gamma \vdash t : A$ *and* $t \twoheadrightarrow_\beta s$*, then* $\Gamma \vdash s : A$*.*

Adding λ-terms to LET_C

Lemma 3.4 (Subject Reduction)

If $\Gamma \vdash t : A$ *and* $t \twoheadrightarrow_\beta s$, *then* $\Gamma \vdash s : A$.

Definition 3.5

A term $t \in \Lambda$ is legal if there is a context Γ and a formula A such that $\Gamma \vdash t : A$.

Adding λ-terms to LET_C

Lemma 3.4 (Subject Reduction)

If $\Gamma \vdash t : A$ and $t \twoheadrightarrow_\beta s$, then $\Gamma \vdash s : A$.

Definition 3.5

A term $t \in \Lambda$ is legal if there is a context Γ and a formula A such that $\Gamma \vdash t : A$.

Theorem 3.6 (Normalization Theorem)

Every legal term is strongly normalising.

UNIVERSIDAD DE ANTIOQUIA 1803

Realisability interpretation for LET_C

Whether a natural l P-realises or N-realises a formula of $\mathrm{N4}^{\star}$ is defined by:

l P-realises $\top$. (1P)

l N-realises $\bot$. (1N)

l P-realises $A \wedge B$ iff $l = \mathrm{P}(m, n)$, m P-realises A and n P-realises B. (2P)

l N-realises $A \wedge B$ iff $l = \mathrm{P}(m, n)$, $m = 0$ and n N-realises A,
or $m > 0$ and n N-realises B. (2N)

l P-realises $A \vee B$ iff $l = \mathrm{P}(m, n)$, $m = 0$ and n P-realises A,
or $m > 0$ and n P-realises B. (3P)

l N-realises $A \vee B$ iff $l = \mathrm{P}(m, n)$, m N-realises A and n N-realises B. (3N)

l P-realises $A \to B$ iff, for every m that P-realises A, $\varphi_l(m)$ P-realises B. (4P)

l N-realises $A \to B$ iff $l = \mathrm{P}(m, n)$, m P-realises A and n N-realises B. (4N)

l P-realises $B \prec A$ iff $l = \mathrm{P}(m, n)$, m N-realises A and n P-realises B. (5P)

l N-realises $B \prec A$ iff, for every m that N-realises A, $\varphi_l(m)$ N-realises B. (5N)

l P-realises $\sim A$ iff l N-realises A. (6P)

l N-realises $\sim A$ iff l P-realises A. (6N)

Realisability interpretation for LET_C

Theorem 4.1

If $B_1, \ldots, B_n \vdash A$*, there is a recursive function* $h : \mathbb{N}^n \to \mathbb{N}$ *such that, if* $l_1, \ldots, l_n$ *P-realise* $B_1, \ldots, B_n$*, respectively, then* $h(l_1, \ldots, l_n)$ *P-realises* A*.*

UNIVERSIDAD DE ANTIOQUIA 1803

Realisability interpretation for LET_C

Theorem 4.1

If $B_1, \ldots, B_n \vdash A$, there is a recursive function $h : \mathbb{N}^n \rightarrow \mathbb{N}$ such that, if $l_1, \ldots, l_n$ P-realise $B_1, \ldots, B_n$, respectively, then $h(l_1, \ldots, l_n)$ P-realises A.

Corollary 4.2

If $B_1, \ldots, B_n \vdash A$ and it is supposed that $l_1, \ldots, l_n$ P-realise $B_1, \ldots, B_n$, respectively, then there is l that P-realises A (particularly, if $\vdash A$, then there is l that P-realises A).

UNIVERSIDAD DE ANTIOQUIA 1803

Realisability interpretation for LET_C

Theorem 4.1

If $B_1, \ldots, B_n \vdash A$, there is a recursive function $h : \mathbb{N}^n \to \mathbb{N}$ such that, if $l_1, \ldots, l_n$ P-realise $B_1, \ldots, B_n$, respectively, then $h(l_1, \ldots, l_n)$ P-realises A.

Corollary 4.2

If $B_1, \ldots, B_n \vdash A$ and it is supposed that $l_1, \ldots, l_n$ P-realise $B_1, \ldots, B_n$, respectively, then there is l that P-realises A (particularly, if $\vdash A$, then there is l that P-realises A).

Theorem 4.3

The following formulas do not have P-realisers:

1. $\bot$ *and* $\sim \top$.
2. $p \vee \sim p$, *for a propositional variable* p.
3. $\sim(p \wedge \sim p)$, *for a propositional variable* p.

Constructible features of LET_C

Some constructive features of LET_C are the following:

Constructible features of LET_C

Some constructive features of LET_C are the following:

- Satisfies the disjunction property (i.e. if $\vdash A \vee B$, then $\vdash A$ or $\vdash B$).

UNIVERSIDAD DE ANTIOQUIA 1803

Constructible features of LET_C

Some constructive features of LET_C are the following:

- Satisfies the disjunction property (i.e. if $\vdash A \vee B$, then $\vdash A$ or $\vdash B$).
- Satisfies the (negation of) conjunction property (i.e. if $\vdash \sim(A \wedge B)$, then $\vdash \sim A$ or $\vdash \sim B$).

UNIVERSIDAD DE ANTIOQUIA 1803

Constructible features of LET_C

Some constructive features of LET_C are the following:

- Satisfies the disjunction property (i.e. if $\vdash A \vee B$, then $\vdash A$ or $\vdash B$).
- Satisfies the (negation of) conjunction property (i.e. if $\vdash \sim(A \wedge B)$, then $\vdash \sim A$ or $\vdash \sim B$).
- Has a BHK-style interpretation (where constructions are evidence).

UNIVERSIDAD DE ANTIOQUIA 1803

Constructible features of LET_C

Some constructive features of LET_C are the following:

- Satisfies the disjunction property (i.e. if $\vdash A \vee B$, then $\vdash A$ or $\vdash B$).
- Satisfies the (negation of) conjunction property (i.e. if $\vdash \sim(A \wedge B)$, then $\vdash \sim A$ or $\vdash \sim B$).
- Has a BHK-style interpretation (where constructions are evidence).
- Evidence has an algorithmic interpretation by means of λ-calculus terms.

UNIVERSIDAD DE ANTIOQUIA 1803

Constructible features of LET_C

Some constructive features of LET_C are the following:

- Satisfies the disjunction property (i.e. if $\vdash A \vee B$, then $\vdash A$ or $\vdash B$).
- Satisfies the (negation of) conjunction property (i.e. if $\vdash \sim(A \wedge B)$, then $\vdash \sim A$ or $\vdash \sim B$).
- Has a BHK-style interpretation (where constructions are evidence).
- Evidence has an algorithmic interpretation by means of λ-calculus terms.
- Has a realisability interpretation.

UNIVERSIDAD DE ANTIOQUIA 1803

Thank you for your attention!

UNIVERSIDAD
DE ANTIOQUIA
1803