

# Implementação de uma biblioteca da Lógica de Inconsistência Formal LFI1 em Coq

Helena Vargas Tannuri

Universidade do Estado de Santa Catarina  
helenavargastannuri@gmail.com

Orientadora: Dra Karina Girardi Roggia

Coorientador: Miguel Alfredo Nunes

22/11/2024

# Sumário

- 1 Introdução
- 2 Objetivos
- 3 Trabalhos Relacionados
- 4 Definições básicas
- 5 **LFI1**
- 6 Metateoremas
- 7 Conclusões
- 8 Referências

- Lógicas paraconsistentes são sistemas não-clássicos que separam a trivialidade da contradição;
  - Usualmente, lógicas ortodoxas assumem que toda teoria contraditória é uma teoria trivial, ou seja, uma teoria com todas as fórmulas.
- Lógica não-clássica é qualquer lógica que quebra algum dos princípios da lógica clássica;
  - As lógicas paraconsistentes quebram o princípio da explosão (definido como  $\alpha \rightarrow (\neg\alpha \rightarrow \beta)$ ) (CARNIELLI; CONIGLIO; MARCOS, 2007).
- Lógicas de inconsistência formal (**LFIs**) são lógicas paraconsistentes que resgatam de maneira *controlada* o princípio da explosão (CARNIELLI; CONIGLIO, 2016).

- Lógicas paraconsistentes, sobretudo **LFI**s, possuem diversas aplicações em diferentes campos do conhecimento;
  - Ciências naturais;
  - Linguística;
  - Computação (bancos de dados com inconsistências).
- **LFI1**: Uma **LFI** que internaliza o conceito de inconsistência dentro de sua linguagem (representado por  $\circ$ ).
  - Resgata a explosividade com o princípio da explosão *gentil* (definido como  $\circ\alpha \rightarrow (\alpha \rightarrow (\neg\alpha \rightarrow \beta))$ ) (CARNIELLI; MARCOS; AMO, 2000).

- Assistentes de provas são ferramentas de *softwares* utilizadas no desenvolvimento de provas formais (CHLIPALA, 2019);
- O Coq é um assistente de provas amplamente utilizado para a verificação formal de *software* e para o desenvolvimento de teoremas, baseando-se no cálculo de construções indutivas (SILVA, 2019);
- A implementação de uma biblioteca da **LFI1** em Coq permite provar metateoremas sobre esta lógica de maneira confiável, pois independe de um avaliador para garantir a correção destas provas;
- Análogo ao que foi desenvolvido em (SILVEIRA, 2020).

Implementar uma biblioteca de **LFI1** em Coq, assim como desenvolver os metateoremas da correção, da completude e da dedução dentro da biblioteca.

# Objetivos Específicos

- 1 Estudar conceitos relevantes sobre lógicas paraconsistentes, em especial a **LFI1**;
- 2 Estudar e revisar as provas manuais para completude, correção e metateorema da dedução da **LFI1**;
- 3 Realizar um levantamento do estado da arte do desenvolvimento de lógicas paraconsistentes em assistentes de provas;
- 4 Desenvolver uma biblioteca da **LFI1** em Coq, baseada na semântica e sintaxe previamente definidas;
- 5 Desenvolver e verificar formalmente as provas para completude, correção e metateorema da dedução em Coq.

- 1 Villadsen e Schlichtkrull (2017) implementam uma biblioteca de uma lógica paraconsistente com infinitos valores-verdade no assistente de provas Isabelle. Algumas metapropriedades são provadas dentro da biblioteca;
- 2 Amo e Pais (2007) especificam uma linguagem de consulta a banco de dados baseada na lógica de inconsistência formal **LFI1**, chamada P-Datalog;
- 3 Ávila, Abe e Prado (1997) descrevem uma linguagem de programação lógica chamada ParaLog\_e, que propõe mesclar conceitos de programação lógica clássica com conceitos de inconsistência, utilizando como base a lógica evidencial.



Uma lógica  $\mathcal{L} = \langle \mathcal{L}, \Vdash \rangle$  que respeita estas propriedades é dita Tarskiana:

- (i) Se  $\alpha \in \Gamma$  então  $\Gamma \Vdash \alpha$ ; (reflexividade)
- (ii) Se  $\Delta \Vdash \alpha$  e  $\Delta \subseteq \Gamma$  então  $\Gamma \Vdash \alpha$ ; (monotonicidade)
- (iii) Se  $\Delta \Vdash \alpha$  e  $\Gamma \Vdash \delta$  para todo  $\delta \in \Delta$  então  $\Gamma \Vdash \alpha$ . (corte)



Uma lógica Tarskiana é dita *padrão* caso ela respeite as seguintes condições:

- (i) Se  $\Gamma \Vdash \alpha$ , então  $\sigma[\Gamma] \Vdash \sigma(\alpha)$ , para toda substituição  $\sigma$  de variável por fórmula.
- (ii) Se  $\Gamma \Vdash \alpha$ , então existe conjunto finito  $\Gamma_0 \subseteq \Gamma$  tal que  $\Gamma_0 \Vdash \alpha$ .

□

Uma lógica Tarskiana é dita *paraconsistente* se ela possuir uma negação  $\neg$  e existirem fórmulas quaisquer  $\alpha, \beta \in \mathcal{L}$  tal que  $\alpha, \neg\alpha \not\vdash \beta$ .



Uma lógica padrão será uma lógica de inconsistência formal (**LFI**) (em relação a  $\bigcirc(p)$  e  $\neg$ , onde  $\bigcirc(p)$  é um conjunto não-vazio de fórmulas dependentes somente da variável  $p$ ) caso respeite as seguintes condições:

- (i) Existem  $\gamma, \delta \in \mathcal{L}$ , de modo que  $\gamma, \neg\gamma \not\vdash \delta$ ;
- (ii) Existem  $\alpha, \beta \in \mathcal{L}$ , de modo que:
  - (ii.a)  $\bigcirc(\alpha), \alpha \not\vdash \beta$ ;
  - (ii.b)  $\bigcirc(\alpha), \neg\alpha \not\vdash \beta$ ;
- (iii) Para todo  $\varphi, \psi \in \mathcal{L}$ , tem-se  $\bigcirc(\varphi), \varphi, \neg\varphi \vdash \psi$ . □

Construção indutiva do menor conjunto  $\mathcal{L}_\Sigma$  que respeita:

1.  $\mathcal{P} \subseteq \mathcal{L}_\Sigma$
2. Se  $\varphi \in \mathcal{L}_\Sigma$ , então  $\Delta\varphi \in \mathcal{L}_\Sigma$ , com  $\Delta \in \{\neg, \circ\}$
3. Se  $\varphi, \psi \in \mathcal{L}_\Sigma$ , então  $\varphi \otimes \psi \in \mathcal{L}_\Sigma$ , com  $\otimes \in \{\wedge, \vee, \rightarrow\}$

□

Cálculo de Hilbert  $\vdash_{\mathbf{LFI1}}$ , com 20 axiomas:

$$\alpha \rightarrow (\beta \rightarrow \alpha) \quad (\mathbf{Ax1})$$

$$(\alpha \rightarrow (\beta \rightarrow \gamma)) \rightarrow ((\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \gamma)) \quad (\mathbf{Ax2})$$

$$\alpha \rightarrow (\beta \rightarrow (\alpha \wedge \beta)) \quad (\mathbf{Ax3})$$

$$(\alpha \wedge \beta) \rightarrow \alpha \quad (\mathbf{Ax4})$$

$$(\alpha \wedge \beta) \rightarrow \beta \quad (\mathbf{Ax5})$$

$$\alpha \rightarrow (\alpha \vee \beta) \quad (\mathbf{Ax6})$$

$$\beta \rightarrow (\alpha \vee \beta) \quad (\mathbf{Ax7})$$

$$(\alpha \rightarrow \gamma) \rightarrow ((\beta \rightarrow \gamma) \rightarrow ((\alpha \vee \beta) \rightarrow \gamma)) \quad (\mathbf{Ax8})$$

$$(\alpha \rightarrow \beta) \vee \alpha \quad (\mathbf{Ax9})$$

$$\alpha \vee \neg \alpha \quad (\mathbf{Ax10})$$

$$\circ \alpha \rightarrow (\alpha \rightarrow (\neg \alpha \rightarrow \beta)) \quad (\mathbf{bc1})$$

$$\neg \neg \alpha \rightarrow \alpha \quad (\mathbf{cf})$$

$$\alpha \rightarrow \neg \neg \alpha \quad (\mathbf{ce})$$

$$\neg \circ \alpha \rightarrow (\alpha \wedge \neg \alpha) \quad (\mathbf{ci})$$

$$\neg(\alpha \vee \beta) \rightarrow (\neg \alpha \wedge \neg \beta) \quad (\mathbf{neg}\vee_1)$$

$$(\neg \alpha \wedge \neg \beta) \rightarrow \neg(\alpha \vee \beta) \quad (\mathbf{neg}\vee_2)$$

$$\neg(\alpha \wedge \beta) \rightarrow (\neg \alpha \vee \neg \beta) \quad (\mathbf{neg}\wedge_1)$$

$$(\neg \alpha \vee \neg \beta) \rightarrow \neg(\alpha \wedge \beta) \quad (\mathbf{neg}\wedge_2)$$

$$\neg(\alpha \rightarrow \beta) \rightarrow (\alpha \wedge \neg \beta) \quad (\mathbf{neg}\rightarrow_1)$$

$$(\alpha \wedge \neg \beta) \rightarrow \neg(\alpha \rightarrow \beta) \quad (\mathbf{neg}\rightarrow_2)$$

Regra de inferência *modus ponens*:

$$\frac{\alpha \quad \alpha \rightarrow \beta}{\beta} \text{MP}$$





A **LFI1** possui dois sistemas semânticos:

- Semântica matricial;
  - Trivalorada;
  - Algébrica.
- Semântica de valorações.
  - Bivalorada;
  - Não-determinística.

$\mathcal{M}_{\text{LFI1}} = \langle M, D, O \rangle$ , com  $M = \{1, 1/2, 0\}$  e  $D = \{1, 1/2\}$ .

$\rightarrow$	1	1/2	0
1	1	1/2	0
1/2	1	1/2	0
0	1	1	1

$\wedge$	1	1/2	0
1	1	1/2	0
1/2	1/2	1/2	0
0	0	0	0

$\vee$	1	1/2	0
1	1	1	1
1/2	1	1/2	1/2
0	1	1/2	0

	$\neg$
1	0
1/2	1/2
0	1

	$\circ$
1	1
1/2	0
0	1

$\Gamma \models_{\mathcal{M}_{\text{LFI1}}} \varphi$  sse, para toda valoração  $h : \mathcal{L}_{\Sigma} \rightarrow M$  de  $\mathcal{M}_{\text{LFI1}}$ , se  $h[\Gamma] \subseteq D$  então  $h(\varphi) \in D$ .

□

# LFI1 - Semântica de valorações

Uma função  $v : \mathcal{L}_\Sigma \rightarrow \{1, 0\}$  é uma valoração para a lógica **LFI1** caso ela satisfaça as seguintes cláusulas:

$$v(\alpha \wedge \beta) = 1 \iff v(\alpha) = 1 \text{ e } v(\beta) = 1 \quad (vAnd)$$

$$v(\alpha \vee \beta) = 1 \iff v(\alpha) = 1 \text{ ou } v(\beta) = 1 \quad (vOr)$$

$$v(\alpha \rightarrow \beta) = 1 \iff v(\alpha) = 0 \text{ ou } v(\beta) = 1 \quad (vImp)$$

$$v(\neg \alpha) = 0 \implies v(\alpha) = 1 \quad (vNeg)$$

$$v(\circ \alpha) = 1 \implies v(\alpha) = 0 \text{ ou } v(\neg \alpha) = 0 \quad (vCon)$$

$$v(\neg \circ \alpha) = 1 \implies v(\alpha) = 1 \text{ e } v(\neg \alpha) = 1 \quad (vCi)$$

$$v(\neg \neg \alpha) = 1 \iff v(\alpha) = 1 \quad (vDNE)$$


$$v(\neg(\alpha \wedge \beta)) = 1 \iff v(\neg \alpha) = 1 \text{ ou } v(\neg \beta) = 1 \quad (vDM_\wedge)$$


$$v(\neg(\alpha \vee \beta)) = 1 \iff v(\neg \alpha) = 1 \text{ e } v(\neg \beta) = 1 \quad (vDM_\vee)$$


$$v(\neg(\alpha \rightarrow \beta)) = 1 \iff v(\alpha) = 1 \text{ e } v(\neg \beta) = 1 \quad (vCip_\rightarrow)$$


$\Gamma \models_{\mathbf{LFI1}} \varphi$  sse, para toda valoração  $v$  de **LFI1**, se  $v[\Gamma] \subseteq \{1\}$  então  $v(\varphi) = 1$ . □


- Uma implementação paramétrica de fusão de sistemas sintáticos semelhante a desenvolvida nesse trabalho não foi encontrado nos trabalhos relacionados;
- Não foi possível terminar a implementação da fusão de sistemas semânticos devido a escolhas de implementação da biblioteca base;
- Não foi possível demonstrar transferência de propriedades no Coq.


 AMO, S. d.; PAIS, M. S. A paraconsistent logic programming approach for querying inconsistent databases. *International Journal of Approximate Reasoning*, v. 46, n. 2, p. 366–386, 2007. ISSN 0888-613X. Special Track on Uncertain Reasoning of the 18th International Florida Artificial Intelligence Research Symposium (FLAIRS 2005). Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0888613X06001307>>.


 ÁVILA, B. C.; ABE, J. M.; PRADO, J. P. d. A. Paralog\_e: a paraconsistent evidential logic programming language. In: *Proceedings 17th International Conference of the Chilean Computer Science Society*. [S.l.: s.n.], 1997. p. 2–8.


 CARNIELLI, W.; CONIGLIO, M.; MARCOS, J. Logics of Formal Inconsistency. In: \_\_\_\_\_. *Handbook of Philosophical Logic*. [S.l.]: Springer, 2007. p. 1–93. ISBN 978-1-4020-6323-7.


 CARNIELLI, W.; CONIGLIO, M. E. *Paraconsistent logic: Consistency, contradiction and negation*. [S.l.]: Springer International Publishing, 2016.

 CARNIELLI, W.; MARCOS, J.; AMO, S. D. Formal inconsistency and evolutionary databases. *Logic and logical philosophy*, p. 115–152, 2000.

 CHLIPALA, A. *Certified programming with dependent types: A pragmatic introduction to the coq proof assistant*. [S.l.]: The MIT Press, 2019.

 SILVA, R. C. G. *Uma certificação em Coq do algoritmo W monádico*. 2019. 78 p. Dissertação (Mestrado) — Universidade do Estado de Santa Catarina, Programa de Pós Graduação em Computação Aplicada, 2019.

 SILVEIRA, A. A. da. *Implementação de uma biblioteca de lógica modal em Coq*. Dissertação (Projeto de Diplomação) — Bacharelado em Ciência da Computação—Centro de Ciências Tecnológicas, UDESC, Joinville, 2020.

 VILLADSEN, J.; SCHLICHTKRULL, A. Formalizing a paraconsistent logic in the Isabelle proof assistant. In: \_\_\_\_\_. *Transactions on Large-Scale Data- and Knowledge-Centered Systems XXXIV: Special Issue on Consistency and Inconsistency in Data-Centric Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017. p. 92–122. ISBN 978-3-662-55947-5. Disponível em: <[https://doi.org/10.1007/978-3-662-55947-5\\_5](https://doi.org/10.1007/978-3-662-55947-5_5)>.