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

27/11/2024

Sumário

- Lógica não-clássica: quebra algum dos princípios da lógica clássica;
- Lógicas paraconsistentes: rompem com o princípio da explosão (contradição = trivialidade);
- Lógicas de inconsistência formal (**LFI**s): lógicas paraconsistentes que resgatam de maneira *controlada* o princípio da explosão (CARNIELLI; CONIGLIO, 2016).
  - Aplicações em diferentes campos do conhecimento (ciências naturais, linguística, computação);
  - **LFI1**: $\circ\alpha \to (\alpha \to (\neg\alpha \to \beta))$ princípio da explosão *gentil* (CARNIELLI; MARCOS; AMO, 2000).

Introdução

- Assistentes de provas: ferramentas de *softwares* utilizadas no desenvolvimento de provas formais (CHLIPALA, 2019);
- Coq: um assistente de provas amplamente utilizado, baseado no cálculo de construções indutivas (CIC) (SILVA, 2019);
  - Provas independem de um avaliador para garantir sua correção, facilitando o processo de verificá-las.

Objetivo Geral

Implementar uma biblioteca de **LFI1** em Coq, assim como desenvolver os metateoremas da correção, da completude e da dedução dentro da biblioteca, análogo ao que foi desenvolvido por Silveira 2020.

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. Desenvolver uma biblioteca da **LFI1** em Coq, baseada na semântica e sintaxe previamente definidas;
4. Desenvolver e verificar formalmente as provas para completude, correção e metateorema da dedução em Coq.

Trabalhos Relacionados

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 **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 $\mathscr{L} = \langle \mathcal{L}, \Vdash \rangle$ 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)

$\square$

Definições básicas - Lógica padrão

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 σ de variável por fórmula.

(ii) Se $\Gamma \Vdash \alpha$, então existe conjunto finito $\Gamma_0 \subseteq \Gamma$ tal que $\Gamma_0 \Vdash \alpha$.

$\square$

Uma lógica Tarskiana é dita *paraconsistente* se existirem fórmulas quaisquer $\alpha, \beta \in \mathcal{L}$ tal que $\alpha, \neg\alpha \nVdash \beta$. $\square$

Uma lógica padrão será uma lógica de inconsistência formal (**LFI**) (considerando $\bigcirc(\alpha)$ é um conjunto não-vazio de fórmulas dependentes somente da fórmula α) caso respeite as seguintes condições:

(i) Existem $\alpha, \beta \in \mathcal{L}$, de modo que $\alpha, \neg\alpha \nVdash \beta$;

(ii) Existem $\alpha, \beta \in \mathcal{L}$, de modo que:

(ii.a) $\bigcirc(\alpha), \alpha \nVdash \beta$;

(ii.b) $\bigcirc(\alpha), \neg\alpha \nVdash \beta$;

(iii) Para todo $\alpha, \beta \in \mathcal{L}$, tem-se $\bigcirc(\alpha), \alpha, \neg\alpha \Vdash \beta$. $\square$

LFI1 - Linguagem

Sendo $\mathcal{P} = \{p_i \mid i \in \mathbb{N}\}$ e $\mathbf{\Sigma} = \{\neg, \circ, \wedge, \vee, \rightarrow\}$, a linguagem é dada pela construção indutiva do menor conjunto $\mathcal{L}_{\mathbf{\Sigma}}$ que respeita:

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

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

$$\alpha \to (\beta \to \alpha) \qquad \textbf{(Ax1)}$$
$$(\alpha \to (\beta \to \gamma)) \to ((\alpha \to \beta) \to (\alpha \to \gamma)) \qquad \textbf{(Ax2)}$$
$$\alpha \to (\beta \to (\alpha \wedge \beta)) \qquad \textbf{(Ax3)}$$
$$(\alpha \wedge \beta) \to \alpha \qquad \textbf{(Ax4)}$$
$$(\alpha \wedge \beta) \to \beta \qquad \textbf{(Ax5)}$$
$$\alpha \to (\alpha \vee \beta) \qquad \textbf{(Ax6)}$$
$$\beta \to (\alpha \vee \beta) \qquad \textbf{(Ax7)}$$
$$(\alpha \to \gamma) \to ((\beta \to \gamma) \to ((\alpha \vee \beta) \to \gamma)) \qquad \textbf{(Ax8)}$$
$$(\alpha \to \beta) \vee \alpha \qquad \textbf{(Ax9)}$$
$$\alpha \vee \neg\alpha \qquad \textbf{(Ax10)}$$

$$\circ\,\alpha \to (\alpha \to (\neg\alpha \to \beta)) \qquad \textbf{(bc1)}$$

$$\neg\neg\alpha \to \alpha \qquad \textbf{(cf)}$$

$$\alpha \to \neg\neg\alpha \qquad \textbf{(ce)}$$

$$\neg \circ \alpha \to (\alpha \wedge \neg\alpha) \qquad \textbf{(ci)}$$

$$\neg(\alpha \vee \beta) \to (\neg\alpha \wedge \neg\beta) \qquad (\textbf{neg}\vee_1)$$

$$(\neg\alpha \wedge \neg\beta) \to \neg(\alpha \vee \beta) \qquad (\textbf{neg}\vee_2)$$

$$\neg(\alpha \wedge \beta) \to (\neg\alpha \vee \neg\beta) \qquad (\textbf{neg}\wedge_1)$$

$$(\neg\alpha \vee \neg\beta) \to \neg(\alpha \wedge \beta) \qquad (\textbf{neg}\wedge_2)$$

$$\neg(\alpha \to \beta) \to (\alpha \wedge \neg\beta) \qquad (\textbf{neg}{\to_1})$$

$$(\alpha \wedge \neg\beta) \to \neg(\alpha \to \beta) \qquad (\textbf{neg}{\to_2})$$

Regra de inferência *modus ponens*:

$$\frac{\alpha \qquad \alpha \rightarrow \beta}{\beta}\,\text{MP}$$

Uma derivação $\Gamma \vdash_{\mathbf{LFI1}} \alpha$ é uma sequência de fórmulas $\varphi_1 \ldots \varphi_n$, com $\varphi_n = \alpha$ onde toda fórmula φ_i respeita alguma das seguintes condições:

1. $\varphi_i \in \Gamma$;
2. φ_i é um axioma;
3. φ_i é resultado de MP em das fórmulas φ_j e φ_k com $k, j < i$.

$\square$

LFI1 - Semântica

A **LFI1** possui dois sistemas semânticos:

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

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

$\to$	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

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

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

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

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

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

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

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

LFI1 - Semântica de valorações

$$v(\neg \circ \alpha) = 1 \Longrightarrow v(\alpha) = 1 \text{ e } v(\neg\alpha) = 1 \qquad (vCi)$$

$$v(\neg\neg\alpha) = 1 \Longleftrightarrow v(\alpha) = 1 \qquad (vDNE)$$

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

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

$$v(\neg(\alpha \to \beta)) = 1 \Longleftrightarrow v(\alpha) = 1 \text{ e } v(\neg\beta) = 1 \qquad (vCip_\to)$$

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

Metateoremas são provas sobre propriedades de um determinado sistema lógico.

- Dedução;
- Correção;
- Completude;
- Equivalência entre sistemas semânticos.

Metateoremas - Dedução e correção

Metateorema da dedução

$\Gamma, \alpha \vdash_{\textbf{LFI1}} \beta$ se, e somente se, $\Gamma \vdash_{\textbf{LFI1}} \alpha \to \beta$.

Correção em relação a semântica matricial

Se $\Gamma \vdash_{\textbf{LFI1}} \alpha$, então $\Gamma \vDash_{\mathcal{M}_{\textbf{LFI1}}} \alpha$.

Correção em relação a semântica de bivalorações

Se $\Gamma \vdash_{\textbf{LFI1}} \alpha$, então $\Gamma \vDash_{\textbf{LFI1}} \alpha$.

Completude em relação a semântica matricial

Se $\Gamma \vDash_{\mathcal{M}_{\mathbf{LFI1}}} \alpha$, então $\Gamma \vdash_{\mathbf{LFI1}} \alpha$.

Completude em relação a semântica de bivalorações

Se $\Gamma \vDash_{\mathbf{LFI1}} \alpha$, então $\Gamma \vdash_{\mathbf{LFI1}} \alpha$.

Equivalência entre semânticas

$\Gamma \vDash_{\mathcal{M}_{\mathbf{LFI1}}} \alpha$ se, e somente se $\Gamma \vDash_{\mathbf{LFI1}} \alpha$.

Presente trabalho:

- Explana conceitos básicos para a caracterização das **LFI**s;
- Define a linguagem, sintaxe e semântica da **LFI1**;
- Revisa e desenvolve manualmente metateoremas para este sistema.

Referências

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. E. *Paraconsistent logic: Consistency, contradiction and negation*. [S.l.]: Springer International Publishing, 2016.

Referências

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>.