UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC
CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO – BCC

HELENA VARGAS TANNURI

IMPLEMENTAÇÃO DE UMA BIBLIOTECA DA LÓGICA DE INCONSISTÊNCIA FORMAL LFI1 EM COQ

JOINVILLE
2024

HELENA VARGAS TANNURI

IMPLEMENTAÇÃO DE UMA BIBLIOTECA DA LÓGICA DE INCONSISTÊNCIA FORMAL LFI1 EM COQ

Trabalho de conclusão de curso submetido à Universidade do Estado de Santa Catarina como parte dos requisitos para a obtenção do grau de Bacharel em Ciência da Computação

Orientadora: Karina Girardi Roggia
Coorientador: Miguel Alfredo Nunes

JOINVILLE
2024

HELENA VARGAS TANNURI

IMPLEMENTAÇÃO DE UMA BIBLIOTECA DA LÓGICA DE INCONSISTÊNCIA FORMAL LFI1 EM COQ

Trabalho de conclusão de curso submetido à Universidade do Estado de Santa Catarina como parte dos requisitos para a obtenção do grau de Bacharel em Ciência da Computação

Orientadora: Karina Girardi Roggia
Coorientador: Miguel Alfredo Nunes

BANCA EXAMINADORA:

Orientadora:

____

Dra. Karina Girardi Roggia
UDESC

Coorientador:

____

Miguel Alfredo Nunes
UNICAMP

Membros:

____

Dr. Cristiano Damiani Vasconcellos
UDESC

____

Me. Paulo Henrique Torrens
University of Kent

Joinville, Junho de 2024

AGRADECIMENTOS

“The fake is of far greater value. In its deliberate attempt to be real, it’s more real than the real thing."

(Kaiki Deishuu – Nisemonogatari, [2012])

RESUMO

Palavras-chave: Coq, Lógica paraconsistente, LFI1, Lógica de Inconsistência Formal, Lógica Trivalorada.

ABSTRACT

Keywords: teste.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

As lógicas paraconsistentes são uma família de lógicas na qual a presença de contradições não implica trivialidade, ou seja, são sistemas lógicos que possuem uma negação que não respeita o Princípio da Explosão, definido como $\alpha \to (\neg\alpha \to \beta)$ (CARNIELLI; CONIGLIO; MARCOS, 2007). Tradicionalmente, em lógicas ortodoxas, qualquer teoria que seja inconsistente - e, portanto, não respeite o Princípio da não-contradição, definido como $\neg(\alpha \wedge \neg\alpha)$ - será uma teoria trivial (uma teoria que contem todas as sentenças). Deste modo, as lógicas paraconsistentes surgem como uma ferramenta que permite tratar contradições sem trivializar o sistema lógico (CARNIELLI; CONIGLIO, 2016).

De acordo com Priest, Tanaka e Weber (2022), as motivações para o estudo de lógicas paraconsistentes podem ser observadas em diversos campos do conhecimento. Nas ciências naturais, por exemplo, teorias inconsistentes e não-triviais são comuns, como é o caso da teoria do átomo de Bohr, que, segundo Brown e Priest (2015), deve possuir um mecanismo de inferência paraconsistente[1]. No campo da linguística, inconsistências não-triviais também são possíveis, como a preservação da noção espacial da palavra "Próximo" mesmo tratando-se de objetos impossíveis[2] (MCGINNIS, 2013). Ademais, no contexto da computação, uma aplicação da paraconsistência é o uso de lógicas de inconsistência formal para a modelagem e o desenvolvimento de bancos de dados evolucionários (CARNIELLI; MARCOS; AMO, 2000).

As lógicas de inconsistência formal (**LFI**s), são lógicas paraconsistentes que introduzem na sua linguagem os conceitos de consistência e inconsistência como formas de representar o excesso de informações (por exemplo, evidência para α e evidência para $\neg\alpha$), para resgatar a capacidade de se obter a trivialidade em alguns casos (CARNIELLI; CONIGLIO; MARCOS, 2007). Ao explicitamente representar a consistência dentro da sua linguagem, é possível estudar teorias inconsistentes sem necessariamente assumir que elas são triviais, porém possibilitando a trivialidade em situações específicas. A ideia por trás das **LFI**s é que deve-se respeitar as noções da lógica clássica o máximo possível, desviando desta somente na presença de contradições. Isto significa que, na ausência de contradições, o Princípio da Explosão deve ser tomado como válido (PRIEST; TANAKA; WEBER, 2022). Segundo Carnielli e Coniglio; Barrio e Carnielli (2016, 2019), na lógica **LFI1**, uma lógica paraconsistente e trivalorada, os conceitos de inconsistência e consistência são introduzidos à linguagem por meio do operador $\bullet$ para a inconsistência ou $\circ$ para a consistência, sendo que qualquer um destes pode ser usado para definir a linguagem da **LFI1**. Desta forma, como veremos ao longo do presente trabalho, e possível resgatar a trivialidade através do Princípio da Explosão Gentil, definido, no caso da $\mathbf{LFI1}_\circ$, como $\circ\alpha \to (\alpha \to (\neg\alpha \to \beta))$(CARNIELLI; CONIGLIO; MARCOS, 2007). Este princípio diz que a

[1] De acordo com a teoria, um elétron orbita o núcelo do átomo sem radiar energia. Porém, de acordo com as equações de Maxwell, que compõem parte da teoria de Bohr, um elétron que está acelerando em órbita deve radiar energia. Estes fatos são inconsistentes entre si, entretanto, não é possível inferir *tudo* sobre o comportamento dos elétrons a partir disso. Portanto, o mecanismo de inferência deve se tratar de um mecanismo paraconsistente.

[2] Por exemplo, na sentença "Adam está próximo de um cubo esférico", a noção espacial entre Adam e um objeto impossível é preservada.

trivialidade é obtida a partir da contradição de uma informação consistente.

Um sistema lógico capaz de lidar com informações inconsistentes é de grande interesse no campo da computação, sobretudo no gerenciamento de bancos de dados (CARNIELLI; MARCOS; AMO, 2000). Um banco de dados pode ser definido como um conjunto estruturado de relações finitas que armazena informações. Estas informações precisam satisfazer condições conhecidas como restrições de integridade antes de serem inseridas no banco (CODD, 1970). As restrições são definidas pelo projetista do banco de dados no momento da implementação e podem ser formalizadas como sentenças de primiera ordem fixas (CARNIELLI; MARCOS; AMO, 2000). Conforme o banco de dados evloui, é preciso atualizar as informações contidas para refletir a realidade, contudo, como informações contraditórias não são permitidas pelas restrições de integridade, isso torna o processo de atualização difícil e trabalhoso. Ademais, a existência de bancos de dados que possam alterar suas restrições de integridade com o passar do tempo (conhecidos como bancos de dados evolucionários) é outro ponto de interesse que pode ser explorado com o uso das **LFI**s.

Concomitante aos estudos das lógicas paraconsistentes, avanços nas áreas da computação e da matemática - como a definição de teoria de tipos por Russell (RUSSELL, 1903; RUSSELL, 1908), a formulação desta teoria com base na sintaxe do Cálculo-λ por Church (1940) e o descobrimento da Correspondência de Curry-Howard (CURRY; FEYS, 1958; HOWARD, 1980) - possibilitaram o desenvolvimento de assistentes de provas (HARRISON; URBAN; WIEDIJK, 2014). Assistentes de provas são ferramentas da área de verificação formal, que buscam garantir que um programa está correto de acordo com uma especificação formal. Isto é feito a partir de provas desenvolvidas utilizando métodos matemáticos para a correção de propriedades de um *software* (CHLIPALA, 2019). Tradicionalmente, a verificação da validade de provas é feita manualmente por avaliadores, que seguem o raciocínio do autor e dão um veredito baseado no quão convincente a prova é. Os assistentes de provas surgem como alternativas à verificação manual, possibilitando ao matemático - ou programador - verificar provas na medida em que elas são desenvolvidas, tornando este processo mais fácil e seguro (PAULIN-MOHRING, 2015).

Assistentes de provas como Coq, Lean e Isabelle permitem ao usuário definir e provar propriedades sobre objetos matemáticos com valor computacional (GEUVERS, 2009). No presente trabalho será utilizado o Coq, este que utiliza o Cálculo de Construções Indutivas como formalismo para o desenvolvimento de provas (TEAM, 2024). O Coq ganhou notoriedade como ferramenta de verificação formal após seu uso na prova de correção de diversos teoremas e sistemas computacionais complexos, como a prova do teorema das quatro qores (GEUVERS, 2009), a certificação de um compilador para a linguagem de programação C (LEROY, 2021) e a prova da correção do algoritmo união-busca (CONCHON; FILLIÂTRE, 2007).

A proposta deste trabalho é desenvolver uma biblioteca da lógica de inconsistência formal **LFI1** em Coq, de maneira análoga como foi feito para a lógica modal em (SILVEIRA, 2020). Após a implementação da biblioteca, serão provados metateoremas relevantes para a **LFI1** utilizando o Coq.

1.1 OBJETIVO GERAL

O objetivo geral deste trabalho é implementar uma biblioteca da **LFI1** em Coq, assim como desenvolver provas da completude, da correção e do metateorema da dedução dentro da biblioteca.

1.2 OBJETIVOS ESPECÍFICOS

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

1.3 TRABALHOS RELACIONADOS

A partir de um levantamento acerca do estado da arte do desenvolvimento de lógicas paraconsistentes em assistentes de provas na literatura, foram encontrados alguns trabalhos semelhantes ao presente trabalho. Estes são: (VILLADSEN; SCHLICHTKRULL, 2017), no qual os autores implementam uma biblioteca de uma lógica paraconsistente utilizando assistente de provas Isabelle. A lógica em questão possui uma quantidade infinita contável de valores verdades não-clássicos, sendo uma generalização da lógica trivalorada proposta por Łukasiewicz, como definida por Simons (2023). Além de implementar a biblioteca, são provados teoremas e metateoremas sobre esta lógica, como o número mínimo de valores verdades a serem analisados para determinar o valor verdade de uma fórmula e os metateoremas da redução e da dedução.

1.4 METODOLOGIA

1.5 ESTRUTURA DO TRABALHO

2 LÓGICAS DE INCONSISTÊNCIA FORMAL

Neste capítulo são apresentadas algumas definições necessárias para caracterizar as lógicas de inconsistência formal, baseadas no trabalho de Carnielli e Coniglio (2016). Antes de definir as **LFI**s, entretanto, é preciso apresentar alguns conceitos acerca de sistemas lógicos paraconsistentes.

2.1 PARACONSISTÊNCIA

Uma lógica $\mathscr{L}$ é definida como uma dupla $\mathscr{L} = \langle \mathcal{L}, \vdash \rangle$, onde $\mathcal{L}$ é sua linguagem (seu conjunto de fórmulas) e $\vdash$ é uma relação de consequência de conclusão única, definida como $\vdash \subseteq \wp(\mathcal{S}) \times \mathcal{S}$.

Definição 1 (Lógica proposicional)**.** Um sistema lógico $\mathscr{L}$, definido sobre uma linguagem $\mathcal{L}$ é dito proposicional caso $\mathcal{L}$ seja definida a partir de um conjunto enumerável de átomos $\mathcal{P} = \{p_i \mid i \in \omega\}$ e uma assinatura proposicional Θ, consistindo em um conjunto de conectivos lógicos com a informação acerca da aridade de cada um destes. Uma linguagem $\mathcal{L}$ definida sobre uma assinatura proposicional é chamada de linguagem proposicional. ■

Definição 2 (Lógica padrão)**.** Uma lógica $\mathscr{L}$, definida sobre uma linguagem $\mathcal{L}$ é dita *Tarskiana* caso satisfaça as seguintes propriedades para todo $\Gamma \cup \Delta \cup \{\alpha\} \subseteq \mathcal{L}$:

(i) Se $\alpha \in \Gamma$ então $\Gamma \vdash \alpha$;

(ii) Se $\Delta \vdash \alpha$ e $\Delta \subseteq \Gamma$ então $\Gamma \vdash \alpha$;

(iii) Se $\Delta \vdash \alpha$ e $\Gamma \vdash \beta$ para todo $\beta \in \Delta$ então $\Gamma \vdash \alpha$.

Uma lógica $\mathscr{L}$ é dita *finitária* caso satisfaça o seguinte:

(iv) Se $\Gamma \vdash \alpha$ então existe um subconjunto finito Γ_0 de Γ tal que $\Gamma_0 \vdash \alpha$.

Uma lógica $\mathscr{L}$ definida sobre uma linguagem proposicional $\mathcal{L}$ é dita *estrutural* caso respeite a seguinte condição:

(v) Se $\Gamma \vdash \alpha$ então $\sigma[\Gamma] \vdash \sigma(\alpha)$, para toda substituição σ de fórmulas por variáveis.

Por fim, uma lógica $\mathscr{L}$ é dita *padrão* caso ela seja Tarskiana, finitária e estrutural. ■

Com isto, é possível definir formalmente a *paraconsistência* para lógicas Tarskianas.

Definição 3 (Lógica Tarskiana paraconsistente)**.** Uma lógica Tarskiana $\mathscr{L}$ é dita *paraconsistente* se ela possuir uma negação $\neg$ (primitiva ou definida) tal que $\alpha, \neg\alpha \nvdash \beta$ para algumas fórmulas α e β na sua linguagem. ■

Caso a linguagem de $\mathscr{L}$ possua uma implicação $\rightarrow$ que respeite o metateorema da dedução[1], então $\mathscr{L}$ é paraconsistente somente se a fórmula $\alpha \rightarrow (\neg\alpha \rightarrow \beta)$ não for válida. Ou seja, o Princípio da explosão, em relação a negação $\neg$, é inválido e, portanto, $\neg$ é uma negação não-explosiva.

2.2 INCONSISTÊNCIA

A motivação para o desenvolvimento das **LFI**s é possuir sistemas lógicos paraconsistentes nos quais é possível resgatar, de maneira *controlada*, o Princípio da Explosão. Isto é feito definindo um conjunto $\bigcirc(p)$ de fórmulas dependentes somente de uma variável proposicional p. Caso uma lógica $\mathscr{L}$ seja explosiva ao unir-se um conjunto $\bigcirc(\alpha)$ com uma contradição $\{\alpha, \neg\alpha\}$, para todo α na sua linguagem, mas não seja explosiva ao unir-se $\bigcirc(\alpha)$ com α nem se unindo $\bigcirc(\alpha)$ com $\neg\alpha$, dizemos que $\mathscr{L}$ é *gentilmente explosiva*.

Definição 4 (Lógica de Inconsistência Formal)**.** Seja $\mathscr{L} = \langle \mathcal{L}_\Sigma, \vdash \rangle$ uma lógica padrão, de forma que sua assinatura proposicional Σ possua uma negação $\neg$. Seja $\bigcirc(p)$ um conjunto não-vazio de fórmulas dependentes somente na variável proposicional p. Então $\mathscr{L}$ será uma *Lógica de Inconsistência Formal* (**LFI**) (em relação a $\bigcirc(p)$ e $\neg$) caso ela respeite as seguintes condições (considerando $\bigcirc(\varphi) = \{\psi(\varphi) \mid \psi(p) \in \bigcirc(p)\}$):

[1] Definido como $\Gamma, \alpha \vdash \beta \iff \Gamma \vdash \alpha \rightarrow \beta$.

3 LFI1

BLABLABLABLABLA introduz um textinho

3.1 LINGUAGEM

A lógica **LFI1**[1] aqui apresentada é definida com base nos trabalhos de Carnielli e Coniglio (2016) como uma estrutura da forma $\langle \mathcal{S}, \Vdash \rangle$, onde $\mathcal{S}$ é a sua linguagem (seu conjunto de fórmulas) e $\Vdash$ é uma relação de consequência de conclusão única definida como $\Vdash \subseteq \wp(\mathcal{S}) \times \mathcal{S}$. A linguagem[2] $\mathcal{S}$ da **LFI1** é definida sobre um conjunto enumerável de átomos $\mathcal{P} = \{p_n \mid n \in \omega\}$ e uma assinatura $\Sigma = \{\wedge, \vee, \rightarrow, \neg, \bullet\}$ da seguinte forma:

Definição 5 (Linguagem da **LFI1**)**.** A linguagem $\mathcal{S}$ da **LFI1** é definida indutivamente como o menor conjunto a partir das seguintes regras:

$\mathcal{P} \subseteq \mathcal{S}$

Se $\varphi \in \mathcal{S}$, então $\triangle\varphi \in \mathcal{S}$, com $\triangle \in \{\neg, \bullet\}$

Se $\varphi, \psi \in \mathcal{S}$, então $\varphi \otimes \psi \in \mathcal{S}$, com $\otimes \in \{\wedge, \vee, \rightarrow\}$ ■

Definição 6 (Subfórmulas)**.** O conjunto $\mathrm{Sub}(\varphi)$ de subfórmulas de uma fórmula φ é definido indutivamente por:

$\mathrm{Sub}(p_i) = \{p_i\},\ p_i \in \mathcal{P}$

$\mathrm{Sub}(\triangle\varphi) = \{\triangle\varphi\} \cup \mathrm{Sub}(\varphi),\ \triangle \in \{\neg, \bullet\}$

$\mathrm{Sub}(\varphi \otimes \psi) = \{\varphi \otimes \psi\} \cup \mathrm{Sub}(\varphi) \cup \mathrm{Sub}(\psi),\ \otimes \in \{\wedge, \vee, \rightarrow\}$ ■

[1] Por mais que existam extensões de primeira ordem da **LFI1**, como a **LFI1***, definida em (CARNIELLI; MARCOS; AMO, 2000), o presente trabalho trata somente do fragmento proposicional desta.

[2] A linguagem da **LFI1** pode ser definida de maneira equivalente utilizando-se o operador de consistência (representado por $\circ$), seguindo a definição $\circ\alpha \stackrel{\text{def}}{=} \neg \bullet \alpha$.

REFERÊNCIAS

BARRIO, Eduardo Alejandro; CARNIELLI, Walter. Volume II: New advances in Logics of Formal Inconsistency. **Logic Journal of the IGPL**, v. 28, n. 5, p. 845–850, 01 2019. ISSN 1367-0751. Disponível em: <https://doi.org/10.1093/jigpal/jzy063>. Citado na página 10.

BROWN, M. Bryson; PRIEST, Graham. Chunk and permeate ii: Bohr?s hydrogen atom. **European Journal for Philosophy of Science**, Springer Verlag, v. 5, n. 3, p. 297–314, 2015. Citado na página 10.

CARNIELLI, Walter; CONIGLIO, Marcelo; MARCOS, João. Logics of formal inconsistency. In: ____. [S.l.]: Springer, 2007. p. 1–93. ISBN 978-1-4020-6323-7. Citado na página 10.

CARNIELLI, Walter; CONIGLIO, Marcelo Esteban. **Paraconsistent logic: Consistency, contradiction and negation**. [S.l.]: Springer International Publishing, 2016. Citado 3 vezes nas páginas 10, 13 e 15.

CARNIELLI, Walter; MARCOS, João; AMO, Sandra De. Formal inconsistency and evolutionary databases. **Logic and logical philosophy**, p. 115–152, 2000. Citado 3 vezes nas páginas 10, 11 e 15.

CHLIPALA, Adam. **Certified programming with dependent types: A pragmatic introduction to the coq proof assistant**. [S.l.]: The MIT Press, 2019. Citado na página 11.

CHURCH, Alonzo. A formulation of the simple theory of types. **The journal of symbolic logic**, Cambridge University Press, v. 5, n. 2, p. 56–68, 1940. Citado na página 11.

CODD, E. F. A relational model of data for large shared data banks. **Commun. ACM**, Association for Computing Machinery, New York, NY, USA, v. 13, n. 6, p. 377–387, jun 1970. ISSN 0001-0782. Disponível em: <https://doi.org/10.1145/362384.362685>. Citado na página 11.

CONCHON, Sylvain; FILLIÂTRE, Jean-Christophe. A persistent union-find data structure. In: **Proceedings of the 2007 Workshop on Workshop on ML**. New York, NY, USA: Association for Computing Machinery, 2007. (ML '07), p. 37–46. ISBN 9781595936769. Disponível em: <https://doi.org/10.1145/1292535.1292541>. Citado na página 11.

CURRY, Haskell Brooks; FEYS, Robert. **Combinatory logic**. Amsterdam: North-Holland Amsterdam, 1958. v. 1. Citado na página 11.

GEUVERS, Herman. Proof assistants: History, ideas and future. **Sadhana**, Springer, v. 34, p. 3–25, 2009. Citado na página 11.

HARRISON, John; URBAN, Josef; WIEDIJK, Freek. History of interactive theorem proving. In: **Computational Logic**. Amsterdam: [s.n.], 2014. v. 9, p. 135–214. Citado na página 11.

HOWARD, William Alvin. The formulae-as-types notion of construction. In: CURRY, Haskell et al. (Ed.). **To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus, and Formalism**. Chicago: Academic Press, 1980. Citado na página 11.

LEROY, Xavier. **The CompCert C verified compiler: Documentation and user's manual**. Tese (Doutorado) — Inria, 2021. Citado na página 11.

MCGINNIS, Nicholas D. The unexpected applicability of paraconsistent logic: A chomskyan route to dialetheism. **Foundations of Science**, Springer Verlag, v. 18, n. 4, p. 625–640, 2013. Citado na página 10.

PAULIN-MOHRING, Christine. Introduction to the calculus of inductive constructions. In: PALEO, Bruno Woltzenlogel; DELAHAYE, David (Ed.). **All about Proofs, Proofs for All**. College Publications, 2015, (Studies in Logic (Mathematical logic and foundations), v. 55). Disponível em: <https://inria.hal.science/hal-01094195>. Citado na página 11.

PRIEST, Graham; TANAKA, Koji; WEBER, Zach. Paraconsistent Logic. In: ZALTA, Edward N. (Ed.). **The Stanford Encyclopedia of Philosophy**. Spring 2022. [S.l.]: Metaphysics Research Lab, Stanford University, 2022. Citado na página 10.

RUSSELL, Bertrand. **Principles of Mathematics**. Cambridge: Cambridge University Press, 1903. Citado na página 11.

RUSSELL, Bertrand. Mathematical logic as based on the theory of types. **American Journal of Mathematics**, Association for Symbolic Logic, v. 30, n. 3, p. 222–262, 1908. Citado na página 11.

SILVEIRA, Ariel Agne 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. Citado na página 11.

SIMONS, Peter. Jan Łukasiewicz. In: ZALTA, Edward N.; NODELMAN, Uri (Ed.). **The Stanford Encyclopedia of Philosophy**. Spring 2023. [S.l.]: Metaphysics Research Lab, Stanford University, 2023. Citado na página 12.

TEAM, The Coq Development. **The Coq Reference Manual**. France, 2024. Citado na página 11.

VILLADSEN, Jørgen; SCHLICHTKRULL, Anders. 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>. Citado na página 12.