

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

*“Per me si va nella città dolente,
per me si va ne l’eterno dolore,
per me si va tra la perduta gente.*

*Giustizia mosse il mio alto fattore;
fecemi la divina podestate, la somma sapienza e
‘l primo amore.*

*Dinanzi a me non fuor cose create
se non etterne, e io eterno duro.
Lasciate ogne speranza, voi ch’intrate.”*

(Dante Alighieri – La Divina Commedia,
[1320])

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	10
1.1	OBJETIVO GERAL	12
1.2	OBJETIVOS ESPECÍFICOS	12
1.3	TRABALHOS RELACIONADOS	13
1.4	METODOLOGIA	13
1.5	ESTRUTURA DO TRABALHO	13
	REFERÊNCIAS	14

1 INTRODUÇÃO

MIGS: Quando você faz citações com `\cite{}` no texto é printado como (AUTOR, YYYY) o que é um estilo de citação que fica feio no meio do texto (veja por exemplo o começo do segundo parágrafo dessa seção) e que é mais adequado para o final de sentenças ou parágrafos. Isso era algo que eu fazia e que cobraram na minha defesa, o melhor, nesses casos, é usar o comando `\citeshort{}` que está definido no `Pacotex.tex`, ele printa as citações como (Autor, YYYY), algo mais adequado para citações dentro do texto. Creio que seria melhor já fazer essa modificação agora no começo e criar o hábito de ir alternando entre os dois comandos quando for mais adequado.

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 Explosão¹ (CARNIELLI; CONIGLIO; MARCOS, 2007). Tradicionalmente, em lógicas ortodoxas, qualquer teoria que seja inconsistente - e, portanto, não respeite o Princípio da não-contradição² - será uma teoria trivial (uma teoria que possui todas as sentenças) **MIGS:** Essa frase ficou meio esquisita, seria interessante rever. 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; 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; PRIEST, 2015), deve possuir uma inferência paraconsistente **MIGS:** Seria interessante pontuar qual é a inferência, mesmo que em nota de rodapé. 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 (MCGINNIS, 2013) **MIGS:** Igualmente aqui, um exemplo mesmo que bobinho ajudaria a entender, tipo “Cebolinha e Cascão estavam próximos um do outro quando chegou Mônica”. 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 (**LFIs**), são lógicas paraconsistentes que introduzem **MIGS:** na sua linguagem os conceitos de consistência e inconsistência como formas de representar o excesso de informações (por exemplo, evidência **MIGS:** de para α e evidência **MIGS:** de 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 **LFIs** é

¹ Definido como $\alpha \rightarrow (\neg\alpha \rightarrow \beta)$.

² Definido como $\neg(\alpha \wedge \neg\alpha)$.

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; CONIGLIO, 2016; BARRIO; CARNIELLI, 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. **MIGS: -A linguagem da LFI1 pode ser definida com qualquer um destes operadores, sendo que qualquer um destes pode ser usado para definir a linguagem da LFI1.** Isto será indicado subscrevendo-se o operador utilizado, como **LFI1_o** e **LFI1_•**. Desta forma, como veremos ao longo do presente trabalho, é possível resgatar a trivialidade através do Princípio da Explosão Gentil, definido, no caso da **LFI1_o**, como $\circ\alpha \rightarrow (\alpha \rightarrow (\neg\alpha \rightarrow \beta))$ (CARNIELLI; CONIGLIO; MARCOS, 2007). Este princípio diz que a 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). **MIGS: Talvez seja interessante pegar uma referência mais recente pra isso, capaz que a banca reclame.** As restrições são definidas pelo projetista do banco de dados no momento da implementação e podem ser formalizadas como sentenças de primeira ordem fixas (CARNIELLI; MARCOS; AMO, 2000). Conforme o banco de dados evolui, 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 (CHURCH, 1940) e o estabelecimento **MIGS: Talvez “descobrimento” seja melhor** 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 provar **MIGS: provas desenvolvidas para provar** 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 cores (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) **MIGS: Olha só, uma justificativa para uso do Coq que não menciona o kernel daquele OS estranho (Nunca tinha ouvido falar dessa prova, irei conferir).**

A proposta deste trabalho é desenvolver uma biblioteca da lógica de inconsistência formal **LFII** 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 **LFII** utilizando o Coq.

1.1 OBJETIVO GERAL

O objetivo geral deste trabalho é implementar uma biblioteca da **LFII** 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 **LFII**;
- Estudar e revisar as provas manuais para completude, correção e metateorema da dedução da **LFII**;
- Realizar um levantamento do estado da arte do desenvolvimento de lógicas paraconsistentes em assistentes de provas;
- Desenvolver uma biblioteca da **LFII** 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 **MIGS: infinita contável implica que é infinito** contável de valores verdades não-clássicos, sendo uma generalização da lógica trivalorada proposta por Łukasiewicz, **MIGS: como** definida em (SIMONS, 2023) **MIGS: Já que você não está citando o trabalho original onde essa lógica foi definida, é melhor você apresentar a citação como “como foi feito por (FULANO, ANO)”, já que quem fez originalmente não foi Fulano, mas ele apresentou da forma que você está se referindo. Além de implementar a biblioteca, são provados teoremas e metateoremas sobre esta lógica MIGS: dentre eles...?**

1.4 METODOLOGIA

1.5 ESTRUTURA DO TRABALHO

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 11.
- 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 2 vezes nas páginas 10 e 11.
- CARNIELLI, Walter; CONIGLIO, Marcelo Esteban. **Paraconsistent logic: Consistency, contradiction and negation**. [S.l.]: Springer International Publishing, 2016. Citado 2 vezes nas páginas 10 e 11.
- CARNIELLI, Walter; MARCOS, João; AMO, Sandra De. Formal inconsistency and evolutionary databases. **Logic and logical philosophy**, p. 115–152, 2000. Citado 2 vezes nas páginas 10 e 11.
- 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 12.
- 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 12.
- 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 12.

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 12.

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 2 vezes nas páginas 10 e 11.

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 12.

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 13.

TEAM, The Coq Development. **The Coq Reference Manual**. France, 2024. Citado na página 12.

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 13.