

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

“Different conclusions are reached when one fact is viewed from two separate points of view. When that happens, there is no immediate way to judge which point of view is the correct one. There is no way to conclude one’s own conclusion is the correct one. But for that exact reason, it is also premature to decide one’s own conclusion is wrong.”

(Senjougahara Hitagi - Bakemonogatari, [2009])

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	12
1.4	METODOLOGIA	12
1.5	ESTRUTURA DO TRABALHO	12
2	LÓGICAS DE INCONSISTÊNCIA FORMAL	13
2.1	PARACONSISTÊNCIA	14
2.2	INCONSISTÊNCIA	16
3	A LÓGICA DE INCONSISTÊNCIA FORMAL LFI1	19
3.1	LINGUAGEM	19
3.2	AXIOMATIZAÇÃO	20
3.3	SEMÂNTICA	22
3.3.1	Matriz Lógica e valorações	23
3.4	METATEOREMAS	24
	REFERÊNCIAS	25

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 $\alpha \rightarrow (\neg\alpha \rightarrow \beta)$ (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 paraconsistente¹. No campo da linguística, inconsistências não-triviais também são possíveis, como a preservação da noção espacial da palavra “Próximo” mesmo tratando-se de objetos impossíveis² (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, é possível resgatar a trivialidade através do Princípio da Explosão Gentil, definido, no caso da **LFI1** $_{\circ}$, como $\circ\alpha \rightarrow (\alpha \rightarrow (\neg\alpha \rightarrow \beta))$ (CARNIELLI; CONIGLIO; MARCOS, 2007). Este princípio diz que a

¹ De acordo com a teoria, um elétron orbita o núcleo 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 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 **LFIs**.

Concomitante aos estudos das lógicas paraconsistentes, avanços nas áreas da computação e da matemática - como a definição de teoria de tipos por Russell (1903, 1908), a formulação desta teoria com base na sintaxe do Cálculo- λ por Church (1940) e o descobrimento da Correspondência de Curry-Howard por Curry e Feys; Howard (1958, 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 confiável (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).

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, propõe-se que sejam provados metateoremas relevantes para a **LFII** 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 e 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

No estudo de lógicas clássicas uma contradição é considerada inseparável da trivialidade, ou seja, se uma teoria possuir um subconjunto $\{\alpha, \neg\alpha\}$ de fórmulas, pode-se derivar qualquer sentença. Esta propriedade é chamada de *explosividade*. Desta forma, as lógicas clássicas (e certas lógicas não-clássicas, como a lógica intuicionista), expressam sua *explosividade* como representada pela seguinte equação: **MIGS: Esse parágrafo ficou muito bom**

$$\text{Contradições} = \text{Trivialidade}$$

As *Lógicas de Inconsistência Formal* são lógicas paraconsistentes que se propõem a questionar a noção apresentada anteriormente sem abrir mão completamente da trivialidade. Isto é feito estabelecendo uma nova propriedade, chamada de *explosividade gentil*, que resgata a trivialidade introduzindo o conceito de consistência na sua linguagem (CARNIELLI; CONIGLIO; MARCOS, 2007). A consistência é expressa na *explosividade gentil* da seguinte forma:

$$\text{Contradições} + \text{Consistência} = \text{Trivialidade}$$

Definir uma lógica que consiga superar o tabu da *explosividade* e, ao mesmo tempo, representar uma ferramenta legítima capaz de formalizar o raciocínio e separar inferências aceitáveis de inferências equivocadas é um dos objetivos dos lógicos que se descrevem como *paraconsistentistas*. As *Lógicas de Inconsistência Formal* cumprem este objetivo de maneira elegante, servindo um propósito importante no estudo de lógicas não-clássicas.

Neste capítulo são apresentadas algumas definições necessárias para caracterizar as *Lógicas de Inconsistência Formal*, baseadas em Carnielli e Coniglio (2016) e em Carnielli, Coniglio e Marcos (2007). Antes de definir as **LFI**s é preciso apresentar alguns conceitos básicos acerca de sistemas lógicos paraconsistentes. Nas definições que seguem, utiliza-se a seguinte representação: **MIGS: Isso deveria ir para a lista de símbolos**

- Letras minúsculas do alfabeto latim p, q, r, \dots para representar fórmulas atômicas.
- Letras maiúsculas do alfabeto latim A, B, C, \dots para representar conjuntos quaisquer.
- Letra minúsculas do alfabeto grego $\alpha, \beta, \gamma, \dots$ para representar fórmulas quaisquer.
- As letras Γ, Δ para representar teorias (conjuntos de fórmulas).
- As letras Σ, Θ para representar a assinatura de uma linguagem.
- O operador \vdash para representar uma relação de consequência sintática.
- O operador \models para representar uma relação de consequência semântica.
- O operador \Vdash para representar uma relação de consequência qualquer (sintática ou semântica).

Ademais, o presente trabalho segue o mesmo caminho de Carnielli e Coniglio (2016), baseando-se na teoria geral de relações de consequências para definir *lógicas tarskianas* **MIGS:** Não gostei muito desse “para definir lógicas tarskianas”, mas não sei direito como mudar preservando o sentido. Neste sentido, como a lógica **LFI1** se trata de uma *lógica tarskiana*, o presente trabalho se restringe a trabalhar somente neste escopo.

Nas Seções 2.1 e 2.2 serão apresentadas definições necessárias para identificar formalmente as lógicas paraconsistentes e lógicas de inconsistência formal. Estas definições se aplicam tanto a relações de consequência semântica quanto a relações de consequência sintática, denotadas genericamente pelo operador \Vdash . Este trabalho não se aprofunda em detalhes sobre conceitos envolvendo relações e operações de consequência, suas propriedades e as diferenças entre abordagens prova-teóricas modelo-teóricas, o leitor interessado nesses assuntos poderia consultar os trabalhos de Wójcicki (1984, 1988a, 1988b) e Carnielli et al. (2008).

2.1 PARACONSISTÊNCIA

Um sistema lógico que se atreve **MIGS:** atreve **HELENA:** eu gostei de escrever esse atreve porque é mól quebrando o tabu tlg a romper com o Princípio da Explosão - o qual afirma que a partir de uma teoria contraditória, qualquer conclusão segue - é dito paraconsistente. As justificativas para questionar tal princípio existem em diversos campos do conhecimento, como na linguística (MCGINNIS, 2013), na computação (CARNIELLI; MARCOS; AMO, 2000) e até mesmo nas ciências naturais (BROWN; PRIEST, 2015). Outra justificativa para o desenvolvimento de sistemas lógicos paraconsistentes é um mero descontentamento com o caráter explosivo das lógicas ortodoxas. Por exemplo, um descontentamento com a conclusão de que “Um triângulo tem quatro lados.” a partir da evidência de que “Choveu na tarde de ontem.” e da evidência de que “Não choveu na tarde de ontem.”¹. Nesta seção, a paraconsistência será definida formalmente, partindo de definições básicas sobre lógica e classificando diferentes sistemas de acordo com propriedades acerca de sua *relação de consequência*.

Uma lógica \mathcal{L} será representada como uma dupla $\mathcal{L} = \langle \mathcal{L}, \Vdash \rangle$, onde \mathcal{L} é sua linguagem (seu conjunto de fórmulas) e \Vdash é uma relação de consequência de conclusão única, definida como $\Vdash \subseteq \wp(\mathcal{L}) \times \mathcal{L}$. Em uma consequência do tipo $A \Vdash \alpha$ (lida como “ α é uma consequência de A .”) diz-se que o conjunto A é o conjunto de premissas e α é a conclusão. A fim de facilitar a escrita e leitura, **MIGS:** uma notação que resume o conjunto de premissas pode ser estabelecida usaremos a seguinte notação no restante do trabalho.

Notação 1. Sejam Γ, Δ teorias e φ, ψ fórmulas, então $\Gamma, \Delta, \varphi \Vdash \psi$ denota $\Gamma \cup \Delta \cup \{\varphi\} \Vdash \psi$.

Definição 1 (Assinatura proposicional). Uma assinatura proposicional Θ é um conjunto de conectivos lógicos **MIGS:** com a informação acerca da aridade de cada um destes cada um contendo também informação sobre sua aridade. ■

¹ Esta forma de explosividade é um exemplo de uma contradição estudada pelas lógicas de relevância, que tratam da conexão entre as premissas e a conclusão de uma inferência (MARES, 2024).

Por exemplo, a assinatura proposicional para a lógica proposicional clássica pode ser definida como $\Theta_{LPC} = \{\wedge^2, \vee^2, \neg^1, \rightarrow^2\}$, onde o operador \wedge^2 representa uma conjunção, \vee^2 representa uma disjunção, \neg^1 representa uma negação e \rightarrow^2 representa uma implicação. No restante do texto, as aridades destes conectivos será omitida e a aridade de novos conectivos será apresentada somente na sua definição.

Uma assinatura proposicional juntamente com um conjunto enumerável de átomos são base para a definição de uma linguagem proposicional, que por sua vez é utilizada para definir uma lógica proposicional, como é o caso da **LF11**.

Definição 2 (Lógica proposicional). Um sistema lógico \mathcal{L} , definido sobre uma linguagem \mathcal{L}_Θ é dito proposicional caso \mathcal{L}_Θ seja definida a partir de um conjunto enumerável de átomos $\mathcal{P} = \{p_i \mid i \in \mathbb{N}\}$ e uma assinatura proposicional Θ . A linguagem \mathcal{L} é chamada de linguagem proposicional. ■

No desenvolvimento de metateoremas sobre propriedades de uma determinada lógica, a indução na complexidade de uma fórmula é um método comum de prova. Para isso, dada uma lógica \mathcal{L} sobre uma linguagem \mathcal{L} , define-se uma função recursiva $C(\varphi) : \mathcal{L} \rightarrow \mathbb{N}$ que retorna, para uma dada fórmula, um número natural representando sua complexidade, baseada na quantidade de operadores e átomos:

Definição 3 (Complexidade de fórmulas para a lógica proposicional clássica). Dada uma fórmula $\varphi \in \mathcal{L}_{LPC}$, a complexidade $C(\varphi)$ é definida recursivamente da seguinte forma:

1. Se $\varphi = p$, onde $p \in \mathcal{P}$, então $C(\varphi) = 1$;
2. Se $\varphi = \neg\psi$, então $C(\varphi) = C(\psi) + 1$;
3. Se $\varphi = \psi \otimes \gamma$, onde $\otimes \in \{\wedge, \vee, \rightarrow\}$, então $C(\varphi) = C(\psi) + C(\gamma) + 1$. ■

Definição 4 (Substituição). Uma substituição σ de todas as ocorrências de uma variável p_i por uma fórmula ψ em uma fórmula φ , é denotada por $\sigma(\varphi) = \varphi\{p_i \mapsto \psi\}$ (SILVA; FINGER; MELO, 2006). A substituição $\varphi\{p_i \mapsto \psi\}$ é definida indutivamente como (considerando Δ , \otimes conectivos quaisquer de aridade 1 e 2 respectivamente): **MIGS: Se você quisesse ser muito pedante, poderia estipular apenas um conectivo qualquer \otimes de aridade n e fazer a definição mais genérica possível**

1. Se $\varphi = p_i$ então, $\varphi\{p_i \mapsto \psi\} = \psi$;
2. Se $\varphi = p_j$ e $j \neq i$ então, $\varphi\{p_i \mapsto \psi\} = \varphi$;
3. Se $\varphi = \Delta\gamma$ então, $\varphi\{p_i \mapsto \psi\} = \Delta(\gamma\{p_i \mapsto \psi\})$;
4. Se $\varphi = \varphi_0 \otimes \varphi_1$ então, $\varphi\{p_i \mapsto \psi\} = \varphi_0\{p_i \mapsto \psi\} \otimes \varphi_1\{p_i \mapsto \psi\}$.

Uma fórmula α é dita *instância de substituição* de uma fórmula β caso exista uma substituição σ tal que $\alpha = \sigma(\beta)$. Ademais, a aplicação de uma substituição σ sobre todos os elementos de uma teoria Γ é definida como $\sigma[\Gamma] = \{\sigma(\varphi) \mid \varphi \in \Gamma\}$. ■

Definição 5 (Lógica Tarskiana). Uma lógica \mathcal{L} , definida sobre uma linguagem \mathcal{L} e munida com uma relação de consequência \Vdash é dita *Tarskiana* caso satisfaça as seguintes propriedades para todo $\Gamma \cup \Delta \cup \{\alpha\} \subseteq \mathcal{L}$:

- (i) Se $\alpha \in \Gamma$ então $\Gamma \Vdash \alpha$; (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)

Com a noção de substituição para lógicas proposicionais apresentada na Definição 4, é possível descrever as lógicas estruturais como sendo lógicas nas quais todas as inferências são fechadas para a substituição:

Uma lógica proposicional \mathcal{L} definida sobre uma linguagem proposicional \mathcal{L} é dita *estrutural* caso respeite a seguinte condição para todo $\Gamma \cup \Delta \cup \{\alpha\} \subseteq \mathcal{L}$:

- (iv) Se $\Gamma \Vdash \alpha$ então $\sigma[\Gamma] \Vdash \sigma(\alpha)$, para toda substituição σ de variável por fórmula.

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

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

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

Com isto, é possível definir formalmente o conceito de *paraconsistência* para lógicas Tarskianas.

Definição 6 (Lógica Tarskiana paraconsistente). Uma lógica Tarskiana \mathcal{L} , definida sobre uma linguagem \mathcal{L} , é dita *paraconsistente* se ela possuir uma negação² \neg tal que existem fórmulas $\alpha, \beta \in \mathcal{L}$ de modo que $\alpha, \neg\alpha \not\Vdash \beta$. ■

Caso a linguagem de \mathcal{L} possua uma implicação \rightarrow que respeite o metateorema da dedução³, então \mathcal{L} é paraconsistente se e somente se a fórmula $\alpha \rightarrow (\neg\alpha \rightarrow \beta)$ não for válida. Ou seja, se o Princípio da Explosão é inválido (em relação a \neg), \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. Ao internalizar o conceito de consistência, as **LFI**s propõem a noção de que uma contradição que é reconhecida inconsistentemente numa dada teoria é inofensiva e é somente fruto do excesso de informação.

² Esta negação pode ser primitiva (pertencente à assinatura da linguagem) ou definida a partir de outras fórmulas.

³ Definido como $\Gamma, \alpha \Vdash \beta \iff \Gamma \Vdash \alpha \rightarrow \beta$.

O resgate *controlado* da explosividade é feito definindo um conjunto $\bigcirc(p)$ de fórmulas dependentes somente em uma variável proposicional p . Caso uma lógica \mathcal{L} seja explosiva ao unir-se um conjunto $\bigcirc(\alpha)$ com uma contradição $\{\alpha, \neg\alpha\}$, ou seja, se $\bigcirc(\alpha), \alpha, \neg\alpha \vdash \beta$ para todo α e β pertencentes à sua linguagem, e ainda $\bigcirc(\alpha), \alpha \not\vdash \beta$ e $\bigcirc(\alpha), \neg\alpha \not\vdash \beta$, então dizemos que \mathcal{L} é *gentilmente explosiva*.

Notação 2. Dado um átomo p , define-se $\bigcirc(p)$ como um conjunto não-vazio de fórmulas dependentes somente em p . Com base neste conjunto, define-se a notação $\bigcirc(\varphi)$ para representar o conjunto obtido pela substituição de todas as ocorrências de p por φ em todos os elementos de $\bigcirc(p)$, ou seja, para uma fórmula φ qualquer, $\bigcirc(\varphi) = \{\psi\{p \mapsto \varphi\} \mid \psi \in \bigcirc(p)\}$.

A definição a seguir foi proposta por Carnielli e Coniglio (2016) e se encontra no meio de outras duas definições, uma mais generalista (apresentada na Definição 8) e outra mais restrita (apresentada na Definição 9). Ela é utilizada para definir o que informalmente descrito como **LFI** no início desta seção.

Definição 7 (Lógica de Inconsistência Formal). Seja $\mathcal{L} = \langle \mathcal{L}_\Theta, \vdash \rangle$ uma lógica padrão, de forma que sua assinatura proposicional Θ possua uma negação \neg . Seja $\bigcirc(p)$ um conjunto não-vazio de fórmulas dependentes somente na variável proposicional p . Então \mathcal{L} será uma *Lógica de Inconsistência Formal* (**LFI**) (em relação a $\bigcirc(p)$ e \neg) caso ela respeite as seguintes condições:

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

A condição (i) diz que toda **LFI** é *não-explosiva* (em relação a \neg) e as condições (ii) e (iii) dizem que toda **LFI** é *gentilmente explosiva* (em relação a $\bigcirc p$ e \neg).

MIGS: Sinto que você ainda não terminou essa seção, mas é importante você falar que o conceito de consistência é internalizado na linguagem por meio de um conectivo e aqui (ou logo depois das outras definições) seria o lugar ideal para você fazer isso. **HELENA:** Isso só serve pra **LFI1** e talvez pra **LFI2**. Outros exemplos de **LFIs** como a J3 e outras maluquices lá sequer têm um operador primitivo pra consistência. Acho que vou adicionar essa explicação em outro momento no texto

Na literatura existem outras definições para as Lógicas de Inconsistência Formal, que relaxam a condição (ii) para obter uma definição mais uniforme. Carnielli e Coniglio (2016), definem **LFIs fracas** da seguinte forma:

Definição 8 (**LFI** fraca). Seja $\mathcal{L} = \langle \mathcal{L}_\Theta, \vdash \rangle$ uma lógica padrão, de forma que sua assinatura proposicional Θ possua uma negação \neg . Seja $\bigcirc(p)$ um conjunto não-vazio de fórmulas dependentes somente na variável proposicional p . Então \mathcal{L} será uma **LFI fraca** (em relação a $\bigcirc(p)$ e

\neg) caso ela respeite as seguintes condições:

- (i) Existem $\varphi, \psi \in \mathcal{L}_\Theta$ de modo que $\varphi, \neg\varphi \not\models \psi$;
- (ii) Existem $\varphi, \psi \in \mathcal{L}_\Theta$ de modo que $\bigcirc(\varphi), \varphi \not\models \psi$;
- (iii) Existem $\varphi, \psi \in \mathcal{L}_\Theta$ de modo que $\bigcirc(\varphi), \neg\varphi \not\models \psi$;
- (iv) Para todo $\varphi, \psi \in \mathcal{L}_\Theta$ tem-se $\bigcirc(\varphi), \varphi, \neg\varphi \models \psi$. ■

Como é possível observar pelas duas definições acima, toda **LFI** é uma **LFI** fraca (já que a condição (ii) da Definição 7 satisfaz as condições (ii) e (iii) da Definição 8), mas o inverso não é necessariamente verdade. Ademais, é possível estabelecer outra definição (também mais uniforme do que a Definição 7) que introduz o conceito de **LFIs fortes** como feito abaixo:

Definição 9 (LFI forte). Seja $\mathcal{L} = \langle \mathcal{L}_\Theta, \models \rangle$ uma lógica padrão, de forma que sua assinatura proposicional Θ possua uma negação \neg . Seja $\bigcirc(p)$ um conjunto não-vazio de fórmulas dependentes somente na variável proposicional p . Então \mathcal{L} será uma **LFI forte** (em relação a $\bigcirc(p)$ e \neg) caso ela respeite as seguintes condições:

- (i) Existem $\alpha, \beta \in \mathcal{L}_\Theta$ de modo que:
 - (i.a) $\alpha, \neg\alpha \not\models \beta$;
 - (i.b) $\bigcirc(\alpha), \alpha \not\models \beta$;
 - (i.c) $\bigcirc(\alpha), \neg\alpha \not\models \beta$;
- (ii) Para todo $\varphi, \psi \in \mathcal{L}_\Theta$ tem-se $\bigcirc(\varphi), \varphi, \neg\varphi \models \psi$. ■

É imediato perceber que toda **LFI forte** é uma **LFI** (já que a condição (i) da Definição 9 satisfaz as condições (i) e (ii) da Definição 7), mas o inverso não é necessariamente verdade. Ademais, no escopo das lógicas proposicionais, é possível estabelecer uma forma mais simples de provar que uma dada lógica proposicional é uma **LFI forte**, tomando α e β como dois átomos p e q quaisquer nas condições (i.a), (i.b) e (i.c) da definição acima.

3 A LÓGICA DE INCONSISTÊNCIA FORMAL LFI1

HELENA: coisas bancos de dados. A *Lógica De Inconsistência Formal LFI1* introduz na sua assinatura o operador \circ para representar a consistência. Uma informação é dita consistente caso ela e sua negação não sejam simultaneamente verdadeiras, ou seja, dada uma informação α , sua consistência $\circ\alpha$ será equivalente a fórmula $\neg(\alpha \wedge \neg\alpha)$. Com a introdução deste novo operador, é possível lidar com a inconsistência de informações sem que trivialidade ocorra, já que, caso uma informação seja conhecidamente *inconsistente*, ou seja, $\neg\circ\alpha$, então ela se trata de uma contradição inofensiva, fruto do excesso de informações numa dada teoria.

No trabalho de Carnielli, Marcos e Amo (2000) a lógica **LFI1*** é definida como uma extensão de primeira ordem da lógica proposicional **LFI1**. A motivação para definir-se uma *Lógica de Inconsistência Formal* de primeira ordem vem da natureza das informações contidas em bancos de dados, estas que podem ser compreendidas como sentenças de primeira ordem fixas (CODD, 1970), entretanto, o presente trabalho trata somente da lógica proposicional **LFI1**. Ademais, Carnielli, Marcos e Amo (2000) tomam o operador de *inconsistência* (denotado por \bullet) como primitivo. Isto foi feito pois o foco era explorar a **LFI1** como uma ferramenta para lidar com inconsistências em bancos de dados, portanto tomar a inconsistência como primitiva era de grande interesse. Entretanto, no presente trabalho, será utilizada a definição apresentada em Carnielli e Coniglio (2016), onde a linguagem é definida utilizando o operador \circ como primitivo. Isto foi feito pois salienta algumas propriedades interessantes da negação \neg , como a presença das leis de De Morgan, axiomatizadas na Seção 3.2.

Este capítulo é dividido da seguinte forma: na Seção 3.1 é apresentada a linguagem da lógica proposicional **LFI1** bem como definições necessárias para desenvolver as provas de metateoremas. Na Seção 3.2 contém uma breve explicação sobre sistemas de prova sintáticos e uma Axiomatização de Hilbert para a **LFI1** é definida. Na Seção 3.3 a semântica da **LFI1** é definida a partir de matrizes lógicas e de uma semântica de valorações não determinística.

3.1 LINGUAGEM

A lógica proposicional **LFI1** aqui apresentada é definida com base em Carnielli e Coniglio (2016) sobre a linguagem \mathcal{L}_Σ , que por sua vez é definida sobre um conjunto enumerável de átomos $\mathcal{P} = \{p_n \mid n \in \mathbb{N}\}$ e uma assinatura proposicional $\Sigma = \{\wedge^2, \vee^2, \rightarrow^2, \neg^1, \circ^1\}$. Como de costume, o conectivo \wedge^2 representa uma conjunção, \vee^2 representa uma disjunção, \rightarrow^2 representa uma implicação, \neg^1 representa uma negação e \circ^1 é o conectivo de consistência, definido de forma primitiva. No restante do texto a aridade destes conectivos será omitida. A linguagem \mathcal{L}_Σ da **LFI1** é definida da seguinte forma:

Definição 10 (Linguagem da **LFI1**). A linguagem \mathcal{L}_Σ da **LFI1** é definida indutivamente como o

menor conjunto a que respeita as seguintes regras:

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 $\otimes \in \{\wedge, \vee, \rightarrow\}$ ■

A precedência dos conectivos é dada de maneira costumeira, com a adição do operador \circ de consistência, seguindo a ordem (da maior precedência para a menor): $\circ, \neg, \wedge, \vee, \rightarrow$. Os conectivos binários \wedge e \vee são associativos à esquerda, ou seja, uma expressão do tipo $\alpha \wedge \beta \wedge \gamma$ é lida como $((\alpha \wedge \beta) \wedge \gamma)$, e o conectivo \rightarrow é associativo à direita, ou seja, uma expressão do tipo $\alpha \rightarrow \beta \rightarrow \gamma$ é lida como $(\alpha \rightarrow (\beta \rightarrow (\gamma)))$.

A linguagem da **LFII** pode ser definida de maneira equivalente utilizando-se o operador de inconsistência (representado por \bullet), definido como $\bullet\alpha \stackrel{\text{def}}{=} \neg \circ \alpha$, como feito por Carnielli, Marcos e Amo (2000).

Na Definição 3 a função C da complexidade de uma fórmula na lógica proposicional clássica foi recursivamente definida. É possível estender esta definição para identificar a complexidade de uma fórmula na **LFII** adicionando-se uma condição para o operador \circ :

Definição 11 (Complexidade de uma fórmula na **LFII**). Dada uma fórmula $\varphi \in \mathcal{L}_\Sigma$, a complexidade $C(\varphi)$ é definida adicionando-se a seguinte condição à Definição 3

Se $\varphi = \circ\psi$, então $C(\varphi) = C(\psi) + 2$. ■

Note que a complexidade de uma fórmula do tipo $\circ\alpha$ é estritamente maior que a complexidade de α e $\neg\alpha$. Isto se dá pois, como evidenciado pela semântica de valorações na Definição 17, existe uma dependência de $\circ\alpha$ em $\{\alpha, \neg\alpha\}$.

3.2 AXIOMATIZAÇÃO

A teoria das provas é uma das abordagens para o estudo das relações de consequência, onde a validade de uma inferência é atestada caso haja uma *prova* das conclusões a partir das premissas. Uma *prova* consiste em uma sequência de passos bem definidos aplicados sobre conjuntos (ocasionalmente unitários) de proposições com base nos princípios de um determinado sistema de provas. A teoria das provas é sintática¹ por natureza, ou seja, numa inferência do tipo $A \vdash B$, é relevante apenas a estrutura das fórmulas presentes em A e B , não sua interpretação ou valor verdade, essa estrutura é manipulada a fim de obter-se uma sequência de passos que – além de atestar sua validade – serve como argumento para tal (BEALL; RESTALL; SAGI, 2024). Desta forma, pode-se definir um sistema de provas sintático para servir como relação de consequência para uma determinada lógica.

¹ Vale notar que a separação (*prova* \times *sintaxe* \times *semântica* \times *modelo*) não é tão bem definida, algo que é explorado em Prawitz (2005).

No contexto da **LFI1** existem dois sistemas de prova sintáticos estabelecidos até o momento: uma Axiomatização no estilo de Hilbert, descrita em Carnielli, Marcos e Amo; Carnielli e Coniglio (2000, 2016) e um sistema de *Tableau*, descrito em Carnielli e Marcos (2001). No presente trabalho, foi escolhida a Axiomatização de Hilbert para definir a sintaxe da **LFI1**, dada a maior facilidade desenvolver metateoremas em relação ao sistema de *Tableau*.

A Axiomatização no estilo de Hilbert (também conhecida como Sistema de Hilbert ou Cálculo de Hilbert) é um sistema composto por um conjunto de fórmulas, chamadas de *axiomas*, e um conjunto de *regras de inferência*. Uma regra de inferência é formada por uma lista de fórmulas chamadas de premissas da regra e uma fórmula chamada de conclusão da regra (RESTALL, 1999). Uma prova (também chamada de derivação) de uma dada fórmula φ consiste em uma sequência finita de fórmulas ψ_0, \dots, ψ_n , onde $\psi_n = \varphi$, e cada ψ_i ($0 \leq i \leq n$) é uma *instância de substituição* (ver Definição 4) de um axioma ou o resultado da aplicação de uma regra de inferência em fórmulas anteriores.

MIGS: Eu sugiro uma breve explicação do que é um sistema sintático de Hilbert assim como uma breve explicação de que existem diversas axiomatizações (para a mesma assinatura) e uma explicação sobre pq vc escolheu a axiomatização que escolheu

Definição 12 (LFI1). A lógica **LFI1** é definida sobre a linguagem \mathcal{L}_Σ através do seguinte cálculo de Hilbert:

(Axiomas Livrão da paraconsistência VERSÃO DO CALCULO DE HILBERT LFI₁):

$\alpha \rightarrow (\beta \rightarrow \alpha)$	(Ax1)
$(\alpha \rightarrow (\beta \rightarrow \gamma)) \rightarrow ((\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \gamma))$	(Ax2)
$\alpha \rightarrow (\beta \rightarrow (\alpha \wedge \beta))$	(Ax3)
$(\alpha \wedge \beta) \rightarrow \alpha$	(Ax4)
$(\alpha \wedge \beta) \rightarrow \beta$	(Ax5)
$\alpha \rightarrow (\alpha \vee \beta)$	(Ax6)
$\beta \rightarrow (\alpha \vee \beta)$	(Ax7)
$(\alpha \rightarrow \gamma) \rightarrow ((\beta \rightarrow \gamma) \rightarrow ((\alpha \vee \beta) \rightarrow \gamma))$	(Ax8)
$(\alpha \rightarrow \beta) \vee \alpha$	(Ax9)
$\alpha \vee \neg \alpha$	(Ax10)
$\circ \alpha \rightarrow (\alpha \rightarrow (\neg \alpha \rightarrow \beta))$	(bc1)
$\neg \neg \alpha \rightarrow \alpha$	(cf)
$\alpha \rightarrow \neg \neg \alpha$	(ce)
$\neg \circ \alpha \rightarrow (\alpha \wedge \neg \alpha)$	(ci)
$\neg(\alpha \vee \beta) \rightarrow (\neg \alpha \wedge \neg \beta)$	(neg _{∨1})
$(\neg \alpha \wedge \neg \beta) \rightarrow \neg(\alpha \vee \beta)$	(neg _{∨2})
$\neg(\alpha \wedge \beta) \rightarrow (\neg \alpha \vee \neg \beta)$	(neg _{∧1})
$(\neg \alpha \vee \neg \beta) \rightarrow \neg(\alpha \wedge \beta)$	(neg _{∧2})
$\neg(\alpha \rightarrow \beta) \rightarrow (\alpha \wedge \neg \beta)$	(neg _{→1})
$(\alpha \wedge \neg \beta) \rightarrow \neg(\alpha \rightarrow \beta)$	(neg _{→2})

Regra de inferência:

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

■

3.3 SEMÂNTICA

De forma geral, a semântica é o estudo de como um sistema de símbolos (uma linguagem) internaliza informações, ou seja, é o estudo de como interpretar os símbolos de uma linguagem (BROWN, 2005). Num sistema lógico, a definição de *matrizes lógicas* é comumente usada para estabelecer o comportamento esperado dos conectivos lógicos de sua assinatura, definindo sua semântica. Outra forma de compreender a semântica de um sistema lógico é definir uma função conhecida como *semântica de valorações*. Nesta seção, a semântica da **LFI1** será definida

de dois jeitos distintos: a partir de uma *matriz lógica* e a partir de uma *semântica de valorações*. Ambas serão provadas equivalentes, a fim de facilitar a prova de metateoremas na Seção 3.4.

3.3.1 Matriz Lógica e valorações

Uma das formas de definir a semântica de uma dada lógica proposicional \mathcal{L}_Θ é definir uma *matriz lógica* (também chamada de tabela-verdade) para os conectivos de sua assinatura proposicional Θ . Para isso, é necessário definir o conceito de *álgebra* para assinaturas proposicionais:

Definição 13 (Álgebra para assinaturas proposicionais). Uma álgebra para uma assinatura proposicional Θ é uma dupla $\mathcal{A} = \langle A, O \rangle$, onde A é um conjunto não vazio (chamado de *domínio* da álgebra) e, para todo conectivo n -ário $c \in \Theta$, O é uma função que define uma operação $c^{\mathcal{A}} : A^n \rightarrow A$ em A . ■

Quando não for confuso, o mesmo símbolo é utilizado para representar um conectivo c e sua interpretação $O(c) = c^{\mathcal{A}}$. Ademais, caso Θ seja finito, a função O é substituída pela lista de conectivos de Θ . Por exemplo, uma álgebra para a assinatura Σ da **LFII** é escrita como $\mathcal{A} = \langle A, \wedge, \vee, \rightarrow, \neg, \circ \rangle$.

Definição 14 (Matriz Lógica). Seja Θ uma assinatura proposicional. Uma *matriz lógica* \mathcal{M} definida sobre Θ é uma tripla $\mathcal{M} = \langle A, D, O \rangle$, tal que o par $\langle A, O \rangle$ é uma álgebra para Θ e D é um subconjunto de A cujos elementos são ditos *designados*. ■

Com isso, uma lógica tarskiana \mathcal{L} sobre uma linguagem \mathcal{L}_Θ é definida por uma matriz lógica $\mathcal{M} = \langle A, D, O \rangle$ da seguinte forma: $\Gamma \models_M \alpha$ sse, para todo homomorfismo $h : \mathcal{L}_\Theta \rightarrow \mathcal{A}$ (onde \mathcal{A} é a álgebra por trás de \mathcal{M}), se $h[\Gamma] \subseteq D$ então $h(\alpha) \in D$. Em particular, α é uma tautologia em \mathcal{L} sse $h(\alpha) \in D$ para todo homomorfismo h .

Definição 15 (Valoração). Seja \mathcal{M} uma matriz definida a partir de uma álgebra \mathcal{A} tal que \mathcal{M} define uma lógica \mathcal{L} sobre a linguagem \mathcal{L}_Θ . Um homomorfismo $h : \mathcal{L}_\Theta \rightarrow \mathcal{A}$ é dito uma valoração sobre \mathcal{M} . ■

MIGS: Acho que seria interessante explicar o que significa o valor verdade de $\frac{1}{2}$

Definição 16 (Matriz lógica da **LFII**). A matriz lógica $\mathcal{M}_{\text{LFII}} = \langle M, D \rangle$ com domínio $M = \{1, \frac{1}{2}, 0\}$ e um conjunto de valores designados $D = \{1, \frac{1}{2}\}$ é definida da seguinte forma:

\rightarrow	$\begin{array}{c ccc} & 1 & \frac{1}{2} & 0 \\ \hline 1 & 1 & \frac{1}{2} & 0 \\ \frac{1}{2} & 1 & \frac{1}{2} & 0 \\ 0 & 1 & 1 & 1 \end{array}$	\wedge	$\begin{array}{c ccc} & 1 & \frac{1}{2} & 0 \\ \hline 1 & 1 & \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 \end{array}$	\vee	$\begin{array}{c ccc} & 1 & \frac{1}{2} & 0 \\ \hline 1 & 1 & 1 & 1 \\ \frac{1}{2} & 1 & \frac{1}{2} & \frac{1}{2} \\ 0 & 1 & \frac{1}{2} & 0 \end{array}$
---------------	--	----------	--	--------	--

	\neg
1	0
$\frac{1}{2}$	$\frac{1}{2}$
0	1

	\circ
1	1
$\frac{1}{2}$	0
0	1

■

A lógica **LFII** é, portanto, dita trivalorada.

MIGS: Aqui acho interessante você explicar que essa é uma semântica não determinística e explicar pq você está incluindo ela, assim como eventualmente provar que ela é intertraduzível para a semântica de matrizes (isso deve ter no livro eu creio, se não fazemos juntos essa prova)

Definição 17 (Semântica de valorações para **LFII**_o). A função $v : \mathcal{L}_\Sigma \rightarrow \{1, 0\}$ é uma valoração para **LFII**_o caso ela satisfaça as seguintes cláusulas:

$v(\alpha \wedge \beta) = 1 \iff v(\alpha) = 1 \text{ e } v(\beta) = 1$	(vAnd)
$v(\alpha \vee \beta) = 1 \iff v(\alpha) = 1 \text{ ou } v(\beta) = 1$	(vOr)
$v(\alpha \rightarrow \beta) = 1 \iff v(\alpha) = 0 \text{ ou } v(\beta) = 1$	(vImp)
$v(\neg \alpha) = 0 \implies v(\alpha) = 1$	(vNeg)
$v(\circ \alpha) = 1 \implies v(\alpha) = 0 \text{ ou } v(\neg \alpha) = 0$	(vCon)
$v(\neg \circ \alpha) = 1 \implies v(\alpha) = 1 \text{ e } v(\neg \alpha) = 1$	(vCi)
$v(\neg \neg \alpha) = 1 \implies v(\alpha) = 1$	(vCf)
$v(\alpha) = 1 \implies v(\neg \neg \alpha) = 1$	(vCe)
$v(\neg(\alpha \wedge \beta)) = 1 \iff v(\neg \alpha) = 1 \text{ ou } v(\neg \beta) = 1$	(vDM _∧)
$v(\neg(\alpha \vee \beta)) = 1 \iff v(\neg \alpha) = 1 \text{ e } v(\neg \beta) = 1$	(vDM _∨)
$v(\neg(\alpha \rightarrow \beta)) = 1 \iff v(\alpha) = 1 \text{ e } v(\neg \beta) = 1$	(vCI _{p→})

3.4 METATEOREMAS

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.
- BEALL, Jc; RESTALL, Greg; SAGI, Gil. Logical Consequence. In: ZALTA, Edward N.; NODELMAN, Uri (Ed.). **The Stanford Encyclopedia of Philosophy**. Summer 2024. [S.l.]: Metaphysics Research Lab, Stanford University, 2024. Citado na página 20.
- BROWN, K. **Encyclopedia of Language and Linguistics**. Elsevier Science, 2005. ISBN 9780080547848. Disponível em: <https://books.google.com.br/books?id=cxYGQfiD_1oC>. Citado na página 22.
- 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 2 vezes nas páginas 10 e 14.
- CARNIELLI, Walter et al. **Analysis and Synthesis of Logics: How to cut and paste reasoning systems**. [S.l.]: Springer, 2008. v. 35. (Applied Logics Series, v. 35). Citado na página 14.
- 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 13.
- CARNIELLI, Walter; CONIGLIO, Marcelo Esteban. **Paraconsistent logic: Consistency, contradiction and negation**. [S.l.]: Springer International Publishing, 2016. Citado 6 vezes nas páginas 10, 13, 14, 17, 19 e 21.
- CARNIELLI, Walter; MARCOS, Joao. Tableau systems for logics of formal inconsistency. In: . [S.l.: s.n.], 2001. v. 2, p. 848–852. Citado na página 21.
- CARNIELLI, Walter; MARCOS, João; AMO, Sandra De. Formal inconsistency and evolutionary databases. **Logic and logical philosophy**, p. 115–152, 2000. Citado 6 vezes nas páginas 10, 11, 14, 19, 20 e 21.
- 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 2 vezes nas páginas 11 e 19.
- 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.

MARES, Edwin. Relevance Logic. In: ZALTA, Edward N.; NODELMAN, Uri (Ed.). **The Stanford Encyclopedia of Philosophy**. Summer 2024. [S.l.]: Metaphysics Research Lab, Stanford University, 2024. Citado na página 14.

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 2 vezes nas páginas 10 e 14.

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.

PRAWITZ, Dag. Logical consequence: A constructivist view. In: SHAPIRO, Stewart (Ed.). **Oxford Handbook of Philosophy of Mathematics and Logic**. [S.l.]: Oxford University Press, 2005. Citado na página 20.

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.

RESTALL, Greg. **An Introduction to Substructural Logics**. New York: Routledge, 1999. Citado na página 21.

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.

SILVA, Flávio Soares Corrêa da; FINGER, Marcelo; MELO, Ana Cristina Vieira de. **Lógica para Computação**. [S.l.]: Cengage Learning, 2006. v. 1. Citado na página 15.

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.

WÓJCICKI, Ryszard. **Lectures on Propositional Calculi**. Ossolineum [Poland]: Pub. House of the Polish Academy of Sciences, 1984. Citado na página 14.

WÓJCICKI, Ryszard. An axiomatic treatment of non-monotonic arguments. **Bulletin of the Section of Logic**, Department of Logic, University of Lodz, v. 17, n. 2, p. 56–61, 1988. Citado na página 14.

WÓJCICKI, Ryszard. **Theory of Logical Calculi: Basic Theory of Consequence Operations**. Dordrecht, Boston and London: Kluwer Academic Publishers, 1988. Citado na página 14.