Charles Ashbacher

Introdução à Lógica Neutrosófica

Editora Brasileira Neutrosófica

Rua Heitor Villa Lobos, 5488
Bairro Flodoaldo Pontes Pinto
Porto Velho, Rondônia
CEP: 76820-628, Brazil

Introdução à Lógica Neutrosófica

Charles Ashbacher

Tradução por

Angelo de Oliveira
Departamento Acadêmico de Ciência da Computação - DACC
Universidade Federal de Rondônia (UNIR)
Porto Velho - Rondônia - Brasil
angelo@unir.br/mrxyztplk@gmail.com/goratchim@gmail.com

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO

A819 Ashbacher, Charles

Introdução à Lógica Neutrosófica/Charles Ashbacher; tradução Angelo de Oliveira. – Porto Velho (RO), 2020.
142 p.

Tradução de: Introduction to Neutrosophic Logic
Inclui bibliografia
ISBN 978-65-000-05717-1.

1. Lógica Matemática. 2. Lógica Neutrosófica. I. Oliveira, Angelo de (tradutor). II. Título.

CDC 511.3
CDU 510.6

Lista de Tabelas

Sumário

Prefácio

Como alguém que trabalha pesadamente em matemática e computadores, eu posso verdadeiramente apreciar o papel que a lógica encena em nosso mundo moderno. Ninguém pode entender os fundamentos da matemática enquanto desconhece os fundamentos de lógica e como provas são construídas. Duas das primeiras disciplinas que eu assisti como aluno de graduação em matemática foram em fundamentos da matemática, e dificilmente passa um dia em que eu não uso algum tópico destes cursos.

Lógica também é um componente fundamental de classes avançadas em computação. Eu estou atualmente lecionando cursos avançados sobre programação em linguagem assembly e organização de computadores. Constante referência é feita sobre como as regras da lógica são incorporadas nos circuitos fundamentais de um computador. A lógica usada nestas classes é conhecida como clássica ou Lógica Booleana.

A lógica neutrosófica é uma extensão da lógica clássica, mas como você verá neste livro, existem dois passos intermediários entre elas. A lógica neutrosófica é outra ideia gerada por Florentin Smarandache, que parece ser uma máquina perpétua de ideias. Tal como a lógica clássica, ela pode ser usada de muitas formas, desde estatística a mecânica quântica.

Contudo, neutrosofia é mais do que uma forma de lógica. Existem diferentes definições, estendendo-se em muitos campos diferentes. Para os nossos propósitos, nos concentraremos quase exclusivamente em lógica devido ao objetivo primário deste livro ser contrastá-la com outras formas de lógica. Existe um jornal devotado à publicação de artigos derivados das ideias de Smarandache chamado Smarandache Notion Journal e que é editado pelo Dr. Minh Perez. Informação a respeito deste e outros avanços podem ser encontrados no web site devotado à postagem dos últimos resultados a respeito das ideias de Smarandache: Smarandache Notions Journal.

Para ver algums dos últimos resultados na área de Neutrosofia, vá para os seguintes sites: Neutrosophic Proceedings e Neutrosophics.

A Primeira Conferência Internacional em Neutrosofia, Lógica Neutrosófica, Conjuntos Neutrosóficos, Probabilidade e Estatística Neutrosófica, foi realizada na Universidade de New Mexico, Gallup, 1-3 de dezembro de 2001 (veja: First International Conference on Neutrosophy).

O capítulo 1 deste livro é devotado a uma introdução aos fundamentos da lógica clássica, ao comportamento do conectivos e aos princípios de raciocínio formal. O próximo capítulo é um exame de uma extensão da lógica clássica chamada lógica trivalente. Com valores que podem ser interpretados como "sim", "não" e "alguma outra coisa", então é possível evitar as restrições forçadas pela lei do terceiro excluído. O capítulo três descreve as regras da lógica fuzzy, onde os valores das variáveis podem ser quaisquer valores no contínuo de zero a um. Finalmente, o capítulo quatro faz uma introdução à lógica neutrosófica, onde o relacionamento entre ela e as outras lógicas ficará claro.

Como sempre, a criação de um livro é um evento complexo, desde o primeiro germe de uma ideia até o último ponto de cola na encadernação. Primeiro e antecipadamente, eu gostaria de agradecer a Florentin Smarandache por todas as suas ideias que eu usei em meu trabalho. Isto inclui três livros, mais de vinte artigos de pesquisa e vários problemas que foram colocados em jornais matemáticos. Ele é verdadeiramente um homem renascentista. Eu tenho livros de sua arte, poesia, filosofia e matemática em minha coleção. Sim, o plural pode ser aplicado a cada categoria várias vezes.

Eu gostaria também de agradecer às pessoas da American Research Press por sua diligência em trazer este livro para uma forma física. Apesar de sua ajuda, todos os erros que permanecem no livro são, é claro, somente de minha responsabilidade. Elogios especiais necessitam serem dados a Minh Perez, que manteve as ideias movendo-se em ambas direções. Lamarr Widmer do Messiah College leu o rascunho inicial deste manuscrito e forneceu muitas sugestões úteis sobre como ele poderia ser melhorado. Obrigado Lamarr! Ele e eu somos parte do conselho editorial de Smarandache Notions Journal e ele também é membro do conselho editorial do Journal of Recreational Mathematics, que eu co-edito.

Eu também gostaria de agradecer mais um vez aqueles que tiveram uma mão em minha educação. Meu mentor, amigo e companheiro de debate Leo Lim está se aposentando no final deste ano acadêmico após mais de trinta anos ensinando química em Mount Mercy College. Eles acharão difícil substituí-lo, e não apenas porque sua marca está impressa na mobília. Finalmente, eu gostaria de mencionar minha família. Eu tenho três das mais trabalhadoras crianças das redondezas. Com idades de 10 e 11, elas tem seu próprio negócio de jardinagem e remoção de neve. Elas são acessórios permanentes na vizinhança, empurrando ou puxando algo. Minha esposa Patti também faz muito para me ajudar em meus empreendimentos profissionais. Na maioria dos casos, isto não é mais do que me deixar sozinho de modo que eu posso movimentar meus dedos sobre o teclado. Ela também fez a arte da capa para este livro[1].

Junho de 2002.

Charles Ashbacher
Charles Ashbacher Technologies
Box 294, 118 Chaffee Drive
Hiawatha, IA 52233, USA.

[1]N. do T.: A capa original do livro.

Prefácio do Tradutor

O tradutor teve seu primeiro contato com a lógica neutrosófica, tal como acontece muitas vezes na vida, por acaso, quando procurava por alguma lógica (não clássica) que lidasse com indeterminação e incerteza (o leitor mais atento notará que estas palavras indeterminação e incerteza, embora próximas e quase sinônimas, possuem significados diferentes em sentido estrito). O presente livro (agora traduzido para o português) foi encontrado no site: Digital Library of Science. Em particular, a versão original do livro pode ser acessada diretamente no seguinte link: Introduction to Neutrosophic Logic.

Dentre os fatores que motivaram o trabalho de tradução, um deles foi a sábia escolha do conteúdo aliada a escrita de fácil leitura provida pelo autor. Os três primeiros capítulos, os quais tratam, respectivamente, da lógica clássica, lógica trivalente (Lukasiewicz, Kleene e Bochvar) e lógica fuzzy, servem adequadamente para o objetivo proposto, o qual é contrastar estas lógicas com a lógica neutrosófica (capítulo 04). Outro fator considerado foi a constatação de que a língua portuguesa é a 5ª (ou 6ª) língua mais falada no mundo. Assim, a presente tradução pode ser útil no processo de expandir o conhecimento da lógica neutrosófica para os ledores em português.

Em relação ao processo de tradução, tal como o autor assevera em seu prefácio que "a criação de um livro é um evento complexo", o processo de tradução de uma obra também não é isenta de erros e omissões. A linguagem inglesa de cunho técnico não se constitui um problema de maior monta, mas considerando que toda linguagem possui suas idiossincrasias, geralmente não se é possível fazer uma tradução literal do conteúdo não técnico. Neste caso, o tradutor procurou aproximar a linguagem original do autor para expressões mais reconhecíveis dos leitores da língua portuguesa, e embora tenha sido tomado o cuidado de tentar se expurgar quaisquer erros, os mesmos ainda podem existir.

Por fim, almeja-se que esta tradução venha por fim alcançar o seu objetivo, o qual é espandir o conhecimento da lógica neutrosófica para o público leitor em língua portuguesa. Quaisquer comentários e susgestões serão muito bem vindos.

Março de 2020.

Prof. Angelo de Oliveira
Departamento Acadêmico de Ciência da Computação
Universidade Federal de Rondônia (Unir)
Campus José Ribeiro Filho - BR 364, Km 9,5
CEP: 76801-059 - Porto Velho - RO - Brasil
e-mail: angelo@unir.br, mrxyztplk@gmail.com e goratchim@gmail.com

Capítulo 1

Lógica Clássica

1.1 Proposições

Na lógica clássica uma variável lógica está restrita aos valores de verdade (T) e falsidade (F). Os conectivos lógicos **and** ($\wedge$), **or** ($\vee$) e **not** ($\neg$) em lógica clássica têm seus comportamentos sumarizados na tabela 1.1.

Outros nomes para os conectivos acima definidos são **conjunção** ($\wedge$), **disjunção** ($\vee$) e **negação** ($\neg$).

> **Nota:** Alguns autores usam letras minúsculas, p, q, r, e assim por diante, para representar variáveis que podem ser verdadeiras ou falsas. Outros usam letras maiúsculas, A, B, C, e assim por diante. Neste livro ambas serão usadas.

Posto que cada variável na lógica clássica está restrita a estes dois valores, se uma expressão tem n diferentes variáveis, a tabela verdade terá 2^n linhas. A coluna resultado de uma tabela-verdade define uma **função booleana**, cujo nome deriva de George Boole, o matemático que foi o primeiro a descrever muitas das regras da lógica. Portanto, uma função booleana é uma função que associa valores de verdade e falsidade a um conjunto de variáveis e retorna um valor verdadeiro ou falso.

p	q	$p \wedge q$	$p \vee q$	$\neg p$
T	T	T	T	F
T	F	F	T	F
F	T	F	T	T
F	F	F	F	T

Tabela 1.1: Tabela verdade para os conectivos and, or e not.

Definição 1.1.1. *Expressões construídas a partir de variáveis na lógica clássica usando estes conectivos são conhecidas como* ***proposições****.*

Exemplo: Cada um das seguintes expressões são proposições.

$(p \wedge q) \vee ((r \wedge \neg s) \vee t)$
$(p \vee q \vee r \vee s)$

A avaliação de proposições é feita aplicando o seguinte conjunto de regras simples:

1. Operações em parêntesis são feitas em primeiro lugar com a mais interna tomando precedência.
2. O conectivo unário not ($\neg$) é aplicado antes de $\wedge$ e $\vee$.
3. Os conectivos $\wedge$ e $\vee$ são considerados como estando no mesmo nível.
4. Se os conectivos estão no mesmo nível, ausentes de parêntesis, eles são avaliados na ordem encontrada quando se move da esquerda para a direita.

Exemplo: Dada a proposição $p \wedge q \vee (r \wedge \neg s) \vee p$, a ordem de avaliação é dada pelos números marcados abaixo dos conectivos, como mostrado abaixo.

p ∧ q ∨ (r ∧ ¬ s) ∨ p
3 4 2 1 **5**

Consequentemente, o valor da proposição é demonstrado na tabela 1.2, onde a coluna resultado é aquela acima do número **5**.

p	q	r	s	p	$\wedge$	q	$\vee$	$(r$	$\wedge$	$\neg$s)	$\vee$	p
T	T	T	T		T		T		F	F	**T**	T
T	T	T	F		T		T		T	T	**T**	T
T	T	F	T		T		T		F	F	**T**	T
T	T	F	F		T		T		F	T	**T**	T
T	F	T	T		F		F		F	F	**T**	T
T	F	T	F		F		T		T	T	**T**	T
T	F	F	T		F		F		F	F	**T**	T
T	F	F	F		F		F		F	T	**T**	T
F	T	T	T		F		F		F	F	**F**	F
F	T	T	F		F		T		T	T	**T**	T
F	T	F	T		F		F		F	F	**F**	F
F	T	F	F		F		F		F	T	**F**	F
F	F	T	T		F		F		F	F	**F**	F
F	F	T	F		F		T		T	T	**T**	T
F	F	F	T		F		F		F	F	**F**	F
F	F	F	F		F		F		F	T	**F**	F
					3		4		2	1	**5**	

Tabela 1.2: Tabela verdade para a proposição $p \wedge q \vee (r \wedge \neg s) \vee p$.

Posto que existem dois possíveis valores para cada entrada na coluna resultante, existem 2^n diferentes funções booleanas para n variáveis lógicas.

Dado os valores verdade na coluna acima do **5**, a proposição que define a função booleana é: $p \wedge q \vee (r \wedge \neg s) \vee p$.

Claramente, dada qualquer proposição, é possível determinar os valores verdade da função booleana que ela define.

> **Nota:** Para qualquer função booleana definida por uma tabela-verdade, existem muitas diferentes proposições que podem ser utilizadas para defini-la.

Dada qualquer função booleana definida por uma tabela-verdade, sempre é possível construir uma proposição cujos valores casam com aqueles constantes na tabela-verdade, e isto é o tópico do teorema seguinte.

Teorema 1.1.1. *Dada qualquer função booleana definida por uma tabela-verdade, é possível construir uma proposição usando os conectivos* $\{\wedge, \vee, \neg\}$ *com os valores verdade casando com os da tabela-verdade.*

Demonstração. Nós a dividimos em dois casos.

Caso 1: A coluna resultado contém somente F.

Neste caso nós simplesmente usamos a expressão F, que tem o valor F para qualquer escolha de valores para qualquer conjunto de variáveis.

Caso 2: A coluna resultado contém ao menos um T.

Identifique todas as linhas onde o resultado é T. Para cada uma destas linhas construa uma expressão examinando os valores das variáveis. Se a variável tem o valor T, então nós a usamos como ela é, e se a variável tem o valor F, colocamos o operador not na frente dela. A expressão é então construída colocando o conectivo $\wedge$ entre as variáveis. $\square$

Exemplo: Dada a linha da tabela-verdade

p_1	p_2	p_3	p_4
T	F	F	T

a expressão para esta linha seria $p1 \wedge \neg p2 \wedge \neg p3 \wedge p4$.

Após um momento de pensamento, deveria estar claro que esta expressão somente é verdadeira para este conjunto específico de valores para as variáveis.

Repetindo o processo para cada linha onde o resultado é T, teríamos um conjunto de expressões onde cada uma delas é verdadeira somente para um caso. Para completar a criação de uma expressão total, cada uma dessas linhas deve ser juntadas usando o conectivo $\vee$. A expressão construída terá então o valor de T aonde quer que o resultado seja T e F em qualquer outro lugar.

Exemplo: Dados os valores verdade na tabela 1.3, a expressão correspondente seria $(p \wedge q \wedge \neg r) \vee (p \wedge \neg q \wedge r) \vee (\neg p \wedge q \wedge r) \vee (\neg p \wedge \neg q \wedge r)$.

p	q	r	Resultado
T	T	T	F
T	T	F	T
T	F	T	T
T	F	F	F
F	T	T	T
F	T	F	F
F	F	T	T
F	F	F	F

Tabela 1.3: Tabela-verdade para a expressão $(p\wedge q\wedge\neg r)\vee(p\wedge\neg q\wedge r)\vee(\neg p\wedge q\wedge r)\vee(\neg p\wedge\neg q\wedge r)$.

Definição 1.1.2. *A expressão que é criada usando as técnicas do teorema 1.1.1 é chamada de **forma disjuntiva normal (FDN)**.*

Definição 1.1.3. *Dada qualquer função booleana definida por uma tabela-verdade, se um conjunto de conectivos pode ser usado para construir uma proposição cujos valores casam com os valores verdade daquela tabela, então o conjunto de conectivos é dito ser **completo**. Do teorema 1.1.1 segue que $\{\wedge, \vee, \neg\}$ é um conjunto completo de conectivos.*

Existem muitos outros conectivos binários que podem ser definidos. É claro que dada a completude do conjunto CS $= \{\wedge, \vee, \neg\}$, todos eles podem ser construídos usando estes três operadores. Consequentemente, cada um destes conectivos adicionais será considerado uma abreviação para uma expressão construída a partir dos operadores em CS.

Outros conjuntos completos também existem. A prova que um dado conjunto de conectivos é completo geralmente é feita mostrando que é possível criar expressões cujo comportamento é equivalente a cada um dos conectivos $\wedge$, $\vee$ e $\neg$.

A definição do conectivo condicional ou implicação ($\rightarrow$) é dada na tabela 1.4, que por sua vez é equivalente à tabela-verdade de $\neg p \vee q$, como pode ser verificado examinando a tabela-verdade para $\neg p \vee q$.

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

Tabela 1.4: Tabela-verdade para o operador condicional.

Definição 1.1.4. *Na expressão $p \to q$, p é conhecido como* ***antecedente*** *e q como* ***consequente****. A implicação é frequentemente descrita como o conectivo se-então.*

O conectivo bicondicional ($\leftrightarrow$) tem os valores verdade da tabela 1.5, sendo que $p \leftrightarrow q$ é equivalente à expressão $(p \wedge q) \vee (\neg p \wedge \neg q)$. O conectivo $\leftrightarrow$ pode ser também considerado como a igualdade lógica.

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Tabela 1.5: Tabela-verdade para o operador bicondicional.

O OR Exclusivo ($\veebar$) pode ser considerado como a desigualdade lógica e tem os valores verdade dados pela tabela 1.6, sendo que $p \veebar q$ tem a mesma tabela-verdade que a expressão $(p \wedge \neg q) \vee (\neg p \wedge q)$.

p	q	$p \veebar q$
T	T	F
T	F	T
F	T	T
F	F	F

Tabela 1.6: Tabela-verdade para o operador OR Exclusivo.

Dois conectivos adicionais que são comumente usados são a negação conjunta ($\downarrow$) e a negação alternativa ($|$). As ações destes conectivos estão resumidas na tabela 1.7.

p	q	$p \downarrow q$	$p|q$
T	T	F	F
T	F	F	T
F	T	F	T
F	F	T	T

Tabela 1.7: Tabela-verdade para os operadores $\downarrow$ e $|$.

A negação conjunta é também chamada de conectivo NOR, um acrônimo para Not OR. Se você examinar os valores verdade, é fácil ver que os resultados são as negações do OR. A negação alternativa é também conhecida como conectivo NAND, um acrônimo para Not AND, significando a negação de AND. Visto que $p \downarrow q$ tem a mesma tabela-verdade que $\neg p \wedge \neg q$ e $p|q$ tem a mesma tabela-verdade que $\neg p \vee \neg q$, estes operadores podem ser considerados abreviações para expressões mais complexas. Em qualquer caso, nós somos levados de volta para o princípio básico que na lógica clássica de proposições, qualquer coisa que você deseje expressar pode ser feita usando os elementos de CS.

Teorema 1.1.2.
1) $(\downarrow)$ *é um conjunto completo de conectivos.*
2) $(|)$ *é um conjunto completo de conectivos.*

Demonstração. Para cada conectivo, necessitamos de um modo de representar o comportamento de cada um dos conectivos em $\wedge$, $\vee$, $\neg$.

O comportamento do conectivo $\neg$ para ambos é exibido na tabela 1.8.

p	$p \downarrow p$	$p\|p$
T	F	F
F	T	T

Tabela 1.8: Representação do comportamento do conectivo $\neg$ usando os conectivos $\downarrow$ e $|$.

O comportamento do conectivo $\wedge$ para ambos e exibido na tabela 1.9.

p	q	$(p \downarrow p) \downarrow (q \downarrow q)$	$(p\|q)\|(p\|q)$
T	T	T	T
T	F	F	F
F	T	F	F
F	F	F	F

Tabela 1.9: Representação do comportamento do conectivo $\wedge$ usando os conectivos $\downarrow$ e $|$.

O comportamento do conectivo $\vee$ para ambos é exibido na tabela 1.10.

p	q	$(p \downarrow q) \downarrow (p \downarrow q)$	$(p\|p)\|(q\|q)$
T	T	T	T
T	F	T	T
F	T	T	T
F	F	F	F

Tabela 1.10: Representação do comportamento do conectivo $\vee$ usando os conectivos $\downarrow$ e $|$.

Posto que o comportamento de cada um dos conectivos $\downarrow$ e $|$ pode ser singularmente usado para criar expressões tendo o mesmo comportamento dos conectivos $\neg$, $\wedge$ e $\vee$, então cada um deles é individualmente um conjunto completo de conectivos. □

Definição 1.1.5. *Uma proposição que é verdadeira para todos os valores das variáveis contidas na expressão é conhecida como **tautologia**. Se uma proposição é falsa para todos os valores das variáveis contidas na expressão, então ela é conhecida como **contradição**. Note que uma proposição p é uma tautologia se, e somente se, $\neg p$ é uma contradição. Se uma proposição não é nem uma tautologia ou uma contradição, então ela é chamada de **contingência**.*

Exemplos: As expressões seguintes são tautologias.

$A \vee \neg A$
$(A \wedge B) \vee T$
$(A \wedge B) \to A$
$A \to (A \vee B)$
$\neg A \to (A \to B)$
$(A \wedge B) \to (A \to B)$
$\neg(A \to B) \to A$
$\neg(A \to B) \to \neg B$

e as expressões seguintes são contradições:

$A \wedge \neg A$
$A \wedge F$

bem como as negações de todas as tautologias.

1.2 A Lei do Terceiro Excluído

Enquanto proposições podem ser usadas em muitas diferentes circunstâncias, existe uma limitação fundamental em seu uso. Posto que a cada expressão deve ser assinalada um valor que é verdadeiro ou falso, as opções são limitadas. Isto é conhecido como a **Lei do Terceiro Excluído**, significando que não existe meio termo entre os dois valores "extremos" de verdade e falsidade.

Uma consequência desta lei é o conceito de **prova por contradição**. O que isto significa é que se não é possível provar que uma expressão válida tem um valor, então ela deve ter o outro. Ao se usar expressões, se a expressão não pode ser provada falsa, então ela é considerada verdadeira. (Muito parecido com a vida, na qual se você não pode provar que uma pessoa está mentindo, então você é forçado a considerá-la como dizendo a verdade.)

A prova por contradição aparece nos assinalamentos de valores do conectivo $\to$. Ela é interpretada como uma declaração na qual se o antecedente é verdadeiro, então a consequência também o é. A declaração é então falsa se o antecedente é verdadeiro e a consequência é falsa. Com esta noção, se não é possível provar que a declaração é falsa, então pela lei do terceiro excluído ela deve ser verdadeira. Consequentemente, as duas últimas linhas da tabela-verdade onde o antecedente é falso tem o valor verdadeiro.

1.3 Equivalência Lógica

Definição 1.3.1. *As proposições p e q são ditas serem logicamente equivalentes se $p \leftrightarrow q$ é uma tautologia. A notação para este relacionamento é $p \Leftrightarrow q$. É possível mostrar que duas expressões são logicamente equivalentes comparando as entradas na tabela-verdade.*

Exemplos: É fácil verificar cada uma das seguintes equivalências lógicas.

$(p \rightarrow q) \Leftrightarrow (\neg p \vee q)$
$(p \leftrightarrow q) \Leftrightarrow ((p \wedge q) \vee (\neg p \wedge \neg q))$
$(p \veebar q) \Leftrightarrow ((p \wedge \neg q) \vee (\neg p \wedge q))$
$(p \downarrow q) \Leftrightarrow (\neg p \wedge \neg q)$
$(p|q) \Leftrightarrow (\neg p \vee \neg q)$

O sinal de igualdade (=) é frequentemente usado como uma notação alternativa para equivalência lógica.

As declarações do teorema 1.3.1 são todas propriedades algébricas de proposições, onde o sinal de igualdade é usado em lugar da flecha dupla bidirecional.

Teorema 1.3.1. *Se A, B e C são proposições:*
a) $A \vee B = B \vee A$ *(comutatividade de $\vee$).*
b) $A \wedge B = B \wedge A$ *(comutatividade de $\wedge$).*
c) $(A \wedge B) \wedge C = A \wedge (B \wedge C)$ *(associatividade de $\wedge$).*
d) $(A \vee B) \vee C = A \vee (B \vee C)$ *(associatividade de $\vee$).*
e) $\neg(A \vee B) = \neg A \wedge \neg B$ *(lei de De Morgan).*
f) $\neg(A \wedge B) = \neg A \vee \neg B$ *(lei de De Morgan).*
g) $\neg\neg A = A$ *(dupla negação).*
h) $A \wedge A = A$ *(idempotência).*
i) $A \vee A = A$ *(idempotência).*
j) $A \wedge \neg A = F$ *(Lei da Contradição).*
k) $A \vee \neg A = T$ *(Lei do Terceiro Excluído).*
l) $A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$ *(propriedade distributiva).*
m) $A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$ *(propriedade distributiva).*
n) $(A \vee B) \wedge A = A$ *(lei de absorção).*
o) $(A \wedge B) \vee A = A$ *(lei de absorção).*
p) $A \vee F = A$ *(lei de identidade).*
q) $A \wedge T = A$ *(lei de identidade).*
r) $A \wedge F = F$ *(lei de dominação).*
s) $A \vee T = T$ *(lei de dominação).*

Demonstração. Deixada como um conjunto de exercícios. □

Todos os conectivos são associativos à esquerda, significando que os conectivos são avaliados da esquerda para a direita. Por exemplo, as expressões

$A \wedge B \wedge C \wedge D$ e $A \rightarrow B \rightarrow C \rightarrow D$

são avaliadas como

$(((A \wedge B) \wedge C) \wedge D)$ e $(((A \rightarrow B) \rightarrow C) \rightarrow D)$.

Para avaliar expressões contendo vários conectivos diferentes, primeiro converta todas as instâncias de $\rightarrow$, $\Leftrightarrow$, $\underline{\vee}$, $\downarrow$ e $|$ para seus equivalentes lógicos em termos de $\wedge$, $\vee$ e $\neg$. Os conectivos $\wedge$ e $\vee$ são considerados como tendo o mesmo nível hierárquico, de modo que é necessário usar parêntesis se a ordem de avaliação é para ser diferente da usual (da esquerda para a direita).

1.4 Fórmulas bem formadas

Definição 1.4.1. *Uma expressão em lógica clássica é dita ser* ***bem-formada*** *ou uma* ***fórmula bem-formada (fbf)*** *se ela pode ser construída usando o seguinte conjunto de regras:*
a) T e F são bem-formadas.
b) Se p_1, p_2,..., p_k,... são variáveis lógicas restritas aos valores de T e F, então todas p_i são bem-formadas.
c) Se A é bem-formada, então também o é (A).
d) Se A é bem-formada, então também o é $\neg A$.
e) Se A e B são bem-formadas, então também o são $A \wedge B$ e $A \vee B$.
f) Somente expressões que podem ser formadas usando as propriedades (a) – (e) são bem-formadas.

O conjunto de proposições é formalmente definido como sendo todas as expressões que podem ser formadas usando as regras da definição de uma fbf.

Entenda que se os valores de todas as variavéis são conhecidos, então a avaliação dos conectivos lógicos será reduzida a expressão T ou F.

Nota: Os conectivos $\rightarrow$, $\leftrightarrow$, $\underline{\vee}$, $\downarrow$ e $|$ não aparecem na definição acima de uma fbf. Consequentemente, a assunção é que estes conectivos são substituidos por suas fórmulas lógicas equivalentes usando $\neg$, $\wedge$ e $\vee$. Esta é a definição padrão de fbfs, apesar de que nada seria alterado se a linha (e) fosse mudada para: *Se A e B são bem-formadas, então tambem o é $A \wedge B$, $A \vee B$, $A \rightarrow B$, $A \leftrightarrow B$, $A \downarrow B$, $A|B$ e $A \underline{\vee} B$.*

1.5 Um Sistema Axiomático para Proposições

Enquanto tabelas-verdade são inestimáveis para trabalho através das muitas características da lógica clássica, elas possuem limitações naturais. Quando sistemas de interpretação são construídos, é impraticável ou impossível usar tabelas-verdade para executar as computações. Para isso necessitamos de uma **teoria formal**.

Definição 1.5.1. *Uma* ***teoria formal*** *é um sistema S construído a partir das seguintes partes:*
a) Um conjunto de símbolos válidos é dado como símbolos de S. Este conjunto pode ser finito ou infinito.
b) Existe um conjunto de regras definindo expressões bem-formadas que podem ser construídas usando os símbolos de S.
c) Um conjunto de fbfs é separado do conjunto completo de fbfs e são chamados de axiomas de S. Estas expressões são tomadas como sendo verdadeiras por assunção.
d) Existe um conjunto finito de relações R_1, R_2,..., R_n entre conjuntos de fbfs em S chamado ***regras de inferência****. Estas regras são usadas para construir provas, onde as declarações são: dado que este conjunto de fbfs é verdadeiro, podemos concluir que outro fbf p também é verdadeiro.*

Em uma prova, o conjunto de fbfs dado é conhecido como hipóteses e p como a consequência. A completa estrutura das hipóteses, quaisquer conclusões intermediárias e o final é conhecido como um **teorema** no sistema S.

Exemplo: O seguinte é uma teoria axiomática formal L.

a) O conjunto de símbolos em L é $\{\neg, \rightarrow, (,), T, F, p_1, p_2, \ldots\}$, onde p_i são variáveis lógicas.
b)
   (1) Todos os símbolos em $\{T, F, p_1, p_2, \ldots\}$ são fbfs em L.
   (2) Se A e B são fbfs em L, então também o são (A), $(\neg A)$ e $A \rightarrow B$.
   (3) Somente expressões que podem ser formadas usando as regras (1) e (2) são fbfs em L.
c) Se A e B são fbfs em L, então o que segue são os axiomas de L.
   (A1) $(A \rightarrow (B \rightarrow A))$
   (A2) $((A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C)))$
   (A3) $((\neg B \rightarrow \neg A) \rightarrow ((\neg B \rightarrow A) \rightarrow B))$
d) A única regra de inferência em L é ***modus ponens***, onde B é uma consequência das duas hipóteses, $A \rightarrow B$ e A. Em outras palavras, se $A \rightarrow B$ e A são verdadeiras, podemos inferir que B também é verdadeira. Esta regra é comumente abreviada para MP e a notação usada é:

$$\mathrm{A} \rightarrow \mathrm{B}, \mathrm{A} \vdash \mathrm{B}$$

Definição 1.5.2. *A teoria axiomática formal L no exemplo anterior é conhecida como* ***cálculo proposicional****.*

Uma vez que uma teoria formal esteja definida, ela pode ser usada para raciocinar a partir de expressões já provadas serem verdadeiras para então concluir que expressões adicionais também são verdadeiras. Tais inferências são chamadas provas na teoria.

Exemplo: O seguinte é uma prova em L.

Hipótese: A.

Prova:
1) Dado A
2) $(A \to (B \to A))$ Axioma 1

Conclusão: por modus ponens $(B \to A)$.

Nota: Uma prova em um esquema formal como o cálculo proposicional é muito diferente da noção de prova em outras áreas da matemática. Em geral, provas formais são mais precisas e sequenciais, com muito menos autonomia nas técnicas que você pode usar.

1.6 Regras Adicionais de Inferência

Modus ponens não é somente a única regra de inferência que pode ser usada em provas formais. O que segue é uma lista de regras adicionais que podem ser aplicadas.

Eliminação – And: A partir de uma conjunção, qualquer uma das proposições na conjunção pode ser inferida.

$$p_1 \wedge p_2 \wedge \ldots p_r \vdash p_i$$

Introdução – And: A partir de uma lista de proposições, uma conjunção pode ser formada.

$$p_1, p_2, \ldots p_r \vdash p_1 \wedge p_2 \wedge \ldots p_r$$

Introdução – Or: Se uma proposição é verdadeira, então a disjunção com ela e qualquer outra proposição pode ser inferida.

$$p_i \vdash p_1 \vee p_2 \vee \ldots \vee p_r$$

Resolução unitária: Se uma disjunção é verdadeira e um de seus elementos é falso, então a outra deve ser verdadeira.

$$p \vee q, \neg q \vdash p$$

Modus tollens: Dada uma implicação e a negação da consequência, podemos inferir a negação do antecedente.

$$p \to q, \neg q \vdash \neg p$$

Resolução: A implicação é transitiva.

$$p \to q, q \to r \vdash p \to r$$

1.7 Raciocínio Formal

Definição 1.7.1. *Faça* $\Gamma = \{a_1, a_2, \ldots, a_k\}$ *representar um conjunto de fbfs. Uma fbf* p *é dita ser uma* ***consequência de*** Γ *se é possível construir uma prova de* p *que começa com as fbfs de* Γ*. Os elementos de* Γ *são chamados de hipóteses ou premissas e* p *é a consequência. Isto pode ser considerado como a operação lógica de começar com um conjunto de declarações conhecidas como sendo verdadeiras e, usando as regras de inferência do sistema, concluir que outra declaração também é verdadeira. A notação para tal inferência é* $\Gamma \vdash p$*. De forma geral, escrevemos* $a_1, a_2, \ldots, a_k \vdash p$ *ao invés de* $\{a_1, a_2, \ldots, a_k\} \vdash p$*.*

O que segue são consequências diretas da definição de consequência.
a) $a_1, a_2, \ldots, a_k \vdash a_i$.
b) $a_1, a_2, \ldots, a_k \vdash p$ então $a_1, a_2, \ldots, a_k, b_1, b_2, \ldots, b_j \vdash p$, onde b_i também são fbfs.

Em outras palavras, se nós podemos provar uma fbf p usando as fbfs em um conjunto, adicionar fbfs para a hipótese não altera a capacidade de derivar p.

1.8 Teoria da Quantificação

Enquanto proposições nos permitem fazer um grande trabalho, elas são limitadas no uso geral porque não se é possível as usar para representar muitas expressões. Considere as seguintes sentenças:

i) Qualquer amigo seu é um amigo meu.
ii) O inimigo do meu inimigo é meu amigo.
iii) Todos os homens são mortais.

Estas expressões contêm dentro delas características onde o valor verdade de uma expressão é dependente de variáveis internas e dos relacionamentos entre as características. Por exemplo, na primeira expressão poderiam existir muitas diferentes pessoas consideradas como sendo suas amigas. Para determinar se esta expressão é verdadeira teríamos que tomar cada e toda instância de seus amigos e determinar se eles também são amigos meus. Isto também requer um entendimento da definição da palavra amigo.

Definição 1.8.1. *Palavras tais como todo, qualquer, qualquer que seja e algum são usadas para inserir restrições no uso de uma variável. O termo formal para isto é* ***quantificador****, e o sistema lógico é chamado* ***teoria da quantificação****.*

Exemplo: Na declaração "Todos os homens são mortais.", a palavra "Todos" claramente significa que a expressão é verdadeira para qualquer valor da variável homem. O termo matemático formal é "para todos", sendo o mesmo conhecido como **quantificador universal**. Se $P(x)$ é a asserção que x tem a propriedade P, então $\forall x P(x)$ significa que para todo x, $P(x)$ é verdadeiro. Se nenhum elemento satisfaz a propriedade, então a expressão poderia ser $\forall x \neg P(x)$, e se no mínimo um elemento não satisfaz a propriedade, a expressão é $\neg \forall x P(x)$.

No caso onde no mínimo um elemento satisfaz a propriedade, o quantificador existencial pode ser usado. A palavra equivalente é "existe" e a expressão é $\exists x P(x)$, que significa que existe um elemento x tal que $P(x)$ é verdadeira.

Nota: Somente um quantificador, ou o existencial ou o universal, precisa ser definido. Por exemplo, se temos o quantificador existencial, podemos escrever a expressão $\neg(\exists x \neg P(x))$, que signigica que "não é o caso que existe algum x tal que $P(x)$ é falsa". Visto que não existe x que torna $P(x)$ falso, a expressão deve ser verdadeira para todo x, o que é equivalente a $\forall x P(x)$.

Similarmente, se nós temos o quantificador universal, escrevemos a expressão $\neg(\forall x P(x))$ que significa "não é o caso que para todo x, $P(x)$ é falsa". Visto que $P(x)$ não é falsa para todo x, então ser verdadeira para algum x.

Definição 1.8.2. *Quando se usa quantificadores, o conjunto de todos os valores que as variáveis podem ter é conhecido como* ***universo de discurso****.*

Definição 1.8.3. *Uma expressão que contém um quantificador é chamada de* ***predicado****.*

Definição 1.8.4. *Se uma variável é afetada por um quantificador, ela é dita ser* ***limitada****. Uma variável que não é afetada por um quantificador é dita ser* ***livre****.*

Exemplo: Na expressão $(\exists x)(x > y)$, x é uma variável limitada e y é uma variável livre. Posto que y é livre, ela pode ser substituída por qualquer outra variável sem alterar o significado da expressão. Portanto, a expressão $(\exists x)(x > z)$ é matematicamente equivalente.

Uma vez que os valores dos elementos no universo de discurso sejam conhecidos, é possível determinar se o predicado é verdadeiro ou falso se ele não contém variáveis livres.

Exemplo: Se o universo de discurso é o conjunto dos números $U = \{1, 2, 3, 4, 5\}$, então a expressão $(\exists x)(x > 0)$ é verdadeira. Adicionalmente, a expressão $(\forall x)(x > 0)$ também é verdadeira.

A teoria da quantificação é frequentemente utilizada com conjuntos de discurso infinitos. Nestes casos, é necessário usar a teoria para determinar se a expressão é verdadeira ou falsa.

Exemplo: Se o universo de discurso é o conjunto de todos os inteiros, então:
(a) $\forall n,\ n^2 > 0$ é falso;
(b) $\forall n,\ n^2 \geq n$ é verdadeiro;
(c) $\exists n,\ n^2 = 2$ é falso.

O equivalente em português destes predicados é:
(a) O quadrado de qualquer inteiro é maior do que zero;
(b) O quadrado de qualquer inteiro n é maior ou igual a n;
(c) Existe um inteiro que é a raiz quadrada de dois.

Exemplo: Se o universo de discurso é o conjunto de todos os números reais x onde $0 \leq x \leq 1$, então:
(a) $\forall x,\ x^2 \leq 1.0$ é verdadeiro;
(b) $\forall x,\ x^2 \leq \mathrm{x}$ é verdadeiro;
(c) $\exists x,\ x^2 < 0$ é falso.

Os equivalentes em português destes predicados são:
(a) O quadrado de qualquer número real entre zero e um (inclusive) é menor ou igual a 1;
(b) O quadrado de qualquer número real entre zero e um (inclusive) é menor ou igual ao próprio ao número;
(c) Existe um número real entre zero e um (inclusive) cujo quadrado é menor do que zero.

Os conectivos de proposições também podem ser usados em combinação com predicados.

Exemplo: Se o universo de discurso é o conjunto de todos os numeros reais x onde $0 \leq x \leq 1$, então:
(a) $(\forall x,\ x^2 \leq 1.0) \wedge (\forall x,\ x^2 \leq x))$ é verdadeiro;
(b) $(\forall x, x^2 < 0) \vee (\forall x, x^2 \leq 1)$ é verdadeiro;
(c) $(\forall x, x^2 < 0) \vee \neg(\forall x, x^2 \leq 1)$ é falso.

1.9 Usos da Lógica em Programação de Computadores

Lógica é um componente fundamental da programação de computadores. Linguagens tais como C++ e Java contêm operadores lógicos correspondentes à conjunção, disjunção e negação. Eles são o duplo ampersand (&&), barra dupla vertical (||) e ponto de exclamação (!). Estes operadores aceitam valores que são booleanos e retornam um booleano[1]. O comportamento das operações envolvendo and, or e not são similares a aquelas da lógica clássica.

Java e outras linguagens de programação contêm operadores adicionais que executam operações de bitwise. Em computadores os dados são expressos na forma binária, ou como uma sequência de zeros e uns. Uma operação de bitwise casa cada uma das posições de duas strings binárias e executa a operação uma posição de cada vez. As operações são similares a aquelas da lógica clássica, com verdadeiro e falso substituídos por um e zero, respectivamente.

Exemplo: As ações dos operadores de bitwise são demonstradas na tabela 1.11.

m	0	1	1	0	1	1	1	0	0	0	1	1	0	0	1	1
n	1	1	1	0	0	0	1	1	0	1	1	0	1	1	1	0
$m \wedge n$	0	1	1	0	0	0	1	0	0	0	1	0	0	0	1	0
$m \vee n$	1	1	1	0	1	1	1	1	0	1	1	1	1	1	1	1
$\neg m$	1	0	0	1	0	0	0	1	1	1	0	0	1	1	0	0

Tabela 1.11: Exemplo da ação dos operadores de bitwise.

[1] N. do T.: A palavra aqui se refere ao tipo de dado booleano, que em C++ e Java pode armazenar os valores `true` e `false`.

1

Capítulo 2

Lógica Trivalente

2.1 Lógica Trivalente de Łukasiewicz

Com somente dois possíveis valores para as variáveis, a lógica clássica usa o que é conhecido como **Lei do Terceiro Excluído**. Isto simplesmente significa que existem somente os dois extremos de verdade e falsidade, com nenhum valor intermediário possível. Embora tenha havido alguns ruídos a respeito de possíveis diferentes tipos de lógica por vários séculos, o primeiro a trabalhar um sistema com mais do que dois valores foi Jan Łukasiewicz, um lógico polonês. Neste sistema existem três valores possíveis, 1, 1/2 e 0.

O operador de negação na lógica trivalente de Łukasiewicz é definido na tabela 2.1.

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

Tabela 2.1: Tabela-verdade para o operador de negação na lógica trivalente de Łukasiewicz.

Nesta lógica, o valor 1/2 pode ser considerado como um valor intermediário de meia verdade e meia falsidade.

Nota: Outro modo que a negação pode ser definida é $\neg p = 1 - p$.

As definições dos conectivos $\wedge$ e $\vee$ na lógica trivalente de Łukasiewicz são dadas na tabela 2.2.

Nota: Observe que $p \vee q$ poderia ser definido como $max\{p, q\}$ e $p \wedge q$ como $min\{p, q\}$.

p	q	$p \vee q$	$p \wedge q$
1	1	1	1
1	1/2	1	1/2
1	0	1	0
1/2	1	1	1/2
1/2	1/2	1/2	1/2
1/2	0	1/2	0
0	1	1	0
0	1/2	1/2	0
0	0	0	0

Tabela 2.2: Tabela-verdade para os operadores $\vee$ e $\wedge$ na lógica trivalente de Łukasiewicz.

Usando as fórmulas de equivalência $(p \rightarrow q) \Leftrightarrow (\neg p \vee q)$, $((p \wedge q) \vee (\neg p \wedge \neg q)) \Leftrightarrow (p \leftrightarrow q)$, $((\neg p \wedge q) \vee (p \wedge \neg q)) \Leftrightarrow (p \veebar q)$, $(\neg p \wedge \neg q) \Leftrightarrow (p \downarrow q)$ e $(\neg p \vee \neg q) \Leftrightarrow (p|q)$, definições para estes conectivos adicionais na lógica trivalente de Łukasiewicz são dados na tabela 2.3.

p	q	$p \rightarrow q$	$p \leftrightarrow q$	$p \veebar q$	$p \downarrow q$	$p|q$
1	1	1	1	0	0	0
1	1/2	1/2	1/2	1/2	0	1/2
1	0	0	0	1	0	1
1/2	1	1	1/2	1/2	0	1/2
1/2	1/2	1/2	1/2	1/2	1/2	1/2
1/2	0	1/2	1/2	1/2	0	1/2
0	1	1	0	1	0	1
0	1/2	1	1/2	1/2	1/2	1/2
0	0	1	1	0	1	1

Tabela 2.3: Tabela-verdade para os operadores $\rightarrow$, $\leftrightarrow$, $\veebar$, $\downarrow$ e $|$ na lógica trivalente de Łukasiewicz.

Note que se 1 é considerado como a representação de verdadeiro e 0 é considerado como a representação de falso, então as entradas das linhas para p e q para zero e um são os mesmos da lógica clássica. Portanto, esta lógica trivalente é uma extensão da lógica clássica. Na maioria dos casos, novas estruturas matemáticas são definidas de modo que elas são consistentes com aquelas previamente definidas.

As definições dos conectivos na tabela 2.3 são consistentes com as fórmulas equivalentes em lógica clássica. Não é necessário manter esta consistência para todas as estruturas lógicas que são extensões da lógica clássica. Por exemplo, a definição do conectivo $\rightarrow$ pode ser mudada para aquela da tabela 2.4.

> **Nota:** Esta definição é ainda consistente com a lógica clássica, como pode ser visto examinando todas as linhas onde os valores de p e q são zero ou um.

p	q	$p \to q$
1	1	1
1	1/2	1/2
1	0	0
1/2	1	1
1/2	1/2	1/2
1/2	0	0
0	1	1
0	1/2	1
0	0	1

Tabela 2.4: Tabela-verdade alternativa para o operador $\to$ na lógica trivalente de Łukasiewicz.

Uma terceira definição do conectivo $\to$ pode ser usada, onde a regra adicional é que se $p > q$, então $p \to q = 0$. A tabela-verdade para esta definição do conectivo $\to$ é dada na tabela 2.5.

p	q	$p \to q$
1	1	1
1	1/2	0
1	0	0
1/2	1	1
1/2	1/2	1/2
1/2	0	0
0	1	1
0	1/2	1
0	0	1

Tabela 2.5: Tabela-verdade alternativa para o operador $\to$ na lógica trivalente de Łukasiewicz.

Nota: Esta definição é também consistente com a definição clássica do conectivo de implicação.

Para completar o exame todos os conectivos, Łukasiewicz definiu os conectivos de implicação e bicondicional do modo ilustrado na tabela 2.6.

p	q	$p \to q$	$p \leftrightarrow q$
1	1	1	1
1	1/2	1/2	1/2
1	0	0	0
1/2	1	1	1/2
1/2	1/2	1	1
1/2	0	1/2	1/2
0	1	1	0
0	1/2	1	1/2
0	0	1	1

Tabela 2.6: Tabela-verdade para os operadores de implicação e equivalência na lógica trivalente de Łukasiewicz (conforme definição de Łukasiewicz).

2.2 A Lógica Trivalente Forte de Kleene

Outra forma de lógica trivalente[1] foi desenvolvida por Stephen Cole Kleene. O terceiro valor nesta lógica é U, que é interpretado como indefinido ou desconhecido. As tabelas verdade dos conectivos $\vee$, $\wedge$, $\to$, $\leftrightarrow$ e $\neg$ são demonstrados nas tabelas 2.7 e 2.8.

p	q	$p \vee q$	$p \wedge q$	$p \to q$	$p \leftrightarrow q$
T	T	T	T	T	T
T	F	T	F	F	F
T	U	T	U	U	U
F	T	T	F	T	F
F	F	F	F	T	T
F	U	U	F	T	U
U	T	T	U	T	U
U	F	U	F	U	U
U	U	U	U	U	U

Tabela 2.7: Tabela-verdade para os operadores $\vee$, $\wedge$, $\to$ e $\leftrightarrow$ na lógica trivalente forte de Kleene.

p	$\neg p$
T	F
F	T
U	U

Tabela 2.8: Tabela-verdade para o operador de negação na lógica trivalente forte de Kleene.

As tabelas verdade da lógica trivalente forte de Kleene podem ser reescritos usando um I para indeterminação ao invés de U para desconhecido.

[1] N. do T.: Também chamada de Lógica de Indeterminação Forte.

Nota: A definição dos conectivos na lógica trivalente forte de Kleene constituem-se também numa extensão da lógica clássica.

Nota: Posto que os valores nesta lógica são não numéricos, as definições dos conectivos não pode envolver quaisquer operações aritméticas tais como *max* e *min*. Isto pode ser uma limitação no modo como os conectivos são usados.

Nota: Os conectivos restantes para a lógica trivalente forte de Kleene podem ser definidos usando as equivalências lógicas da lógica clássica.

É possível definir os quantificadores universal e existencial na lógica trivalente forte de Kleene. Isto é feito tratando a quantificação universal como uma conjunção infinita e o quantificador existencial como uma disjunção infinita.

Definição 2.2.1. *Faça*

$$\bigwedge_{i \in U} p_i$$

ser uma conjunção infinita de variáveis na lógica trivalente forte de Kleene. Ela tem o valor T se todos p_i são T, F se alguns p_i são F e U caso contrário.

Definição 2.2.2. *Faça*

$$\bigvee_{i \in U} p_i$$

ser uma disjunção infinita de variáveis na lógica trivalente forte de Kleene. Ela tem o valor T se alguns p_i são T, F se todos p_i são F e U caso contrário.

2.3 Lógica Trivalente de Bochvar

Uma lógica trivalente adicional foi criada por D. Bochvar e foi inspirada pelo exame de paradoxos semânticos. Por exemplo, existe a declaração clássica: "Esta expressão é falsa". Se ela é verdadeira, então ela deve ser falsa, e se ela é falsa, então ela deve ser verdadeira, o que é um paradoxo. A solução de Bochvar para tais declarações é introduzir o valor adicional M, que representa sem significado ou paradoxal. O comportamento dos conectivos é simples, sendo que se qualquer variável na expressão tem o valor M, então a expressão tem o valor M. Isto é demonstrado nas tabelas 2.9 e 2.10.

p	$\neg p$
T	F
F	T
M	M

Tabela 2.9: Tabela-verdade para o operador $\neg$ na lógica trivalente de Bochvar.

p	q	$p \vee q$	$p \wedge q$	$p \rightarrow q$	$p \leftrightarrow q$	$p \veebar q$	$p \downarrow q$	$p \vert q$
T	T	T	T	T	T	F	F	F
T	F	T	F	F	F	T	F	T
T	M	M	M	M	M	M	M	M
F	T	T	F	T	F	T	F	T
F	F	F	F	T	T	F	T	T
F	M	M	M	M	M	M	M	M
M	T	M	M	M	M	M	M	M
M	F	M	M	M	M	M	M	M
M	M	M	M	M	M	M	M	M

Tabela 2.10: Tabela-verdade para os operadores $\vee$, $\wedge$, $\rightarrow$, $\leftrightarrow$, $\veebar$, $\downarrow$ e $|$ na lógica trivalente de Bochvar.

Nota: Mais uma vez, os conectivos definidos para a lógica de Bochvar são extensões de suas contrapartes clássicas.

Nota: A lógica trivalente de Bochvar pode ser considerada uma na qual M é dominante.

É possível definir os quantificadores universal e existencial na lógica de Bochvar. Mais uma vez, conjunções e disjunções infinitas são usadas.

Definição 2.3.1. *Faça*

$$\bigwedge_{i \in U} p_i$$

ser uma conjunção infinita de variáveis na lógica de Bochvar. Ela tem o valor T se todos p_i são T, F se algum p_i é F e nenhuma variável é M e M caso contrário.

Definição 2.3.2. *Faça*

$$\bigvee_{i \in U} p_i$$

ser uma disjunção infinita de variáveis na lógica de Bochvar. Ela tem o valor T se algum p_i é T e nenhuma variável é M, F se cada p_i é F e M caso contrário.

2.4 Lógica Trivalente em Programação de Computadores

Em linguagens de programação tais como C, C++ e Java, os conectivos and (&&) e or (||) são avaliados de um modo condicional. Na expressão conjuntiva `E1 && E2`, a expressão `E1` é avaliada em primeiro lugar, e se ela é falsa, `E2` não é avaliada e a expressão retorna um `F`. Portanto, neste caso, não faz nenhuma diferença se `E2` é indefinida, pois `E2` é avaliada somente quando `E1` é `T`. Se `E1` é verdadeira, então o resultado é o valor de `E2`.

Se a expressão é uma disjução `E1 || E2`, a expressão `E1` é novamente avaliada em primeiro lugar, e se ela é verdadeira, `E2` não é avaliada e a expressão retorna um `T`. `E2` é avaliada somente quando `E1` é falsa. O valor da expressão é então `E2` quando `E1` é falsa. O comportamento das avaliações condicionais `||` (or) e `&&` (and) em computação é resumido na tabela 2.11.

p	q	p && q	$p \parallel q$
T	T	T	T
T	F	F	T
T	U	U	T
F	T	F	T
F	F	F	F
F	U	F	U
U	T	U	U
U	F	U	U
U	U	U	U

Tabela 2.11: Tabela-verdade para o operadores `||` e `&&` em linguagens de programação.

2.5 Lógica Trivalente com um valor Indeterminado

Outra forma de lógica trivalente usa o valor I para indeterminação. Nesta lógica, se qualquer variável tem o valor I, então o resultado é I. As tabelas-verdade poderiam ser a mesma daquelas para a lógica de Bochvar, aonde o M é substituído por um I.

A definição de lógica trivalente que usa o I tem aplicações no ramo da física conhecido como mecânica quântica. Mecânica quântica é a física do muito pequeno, onde eventos podem simultaneamente estar em mais do que um estado e o estado corrente não é conhecido até que uma medição seja feita.

Talvez a mais famosa das descrições da mecânica quântica seja o paradoxo do gato de Schrödinger. Um gato vivo é colocado em um container fechado com um átomo de um elemento radioativo. Se o átomo decai, ele será detectado no container e causará a quebra de uma bula de veneno que matará o gato.

De acordo com a mecânica quântica, a probabilidade que o átomo venha a decair durante o intervalo de sua meia vida é 1/2. Consequentemente, se o gato é colocado no container e ele este é fechado pelo intervalo de meia vida, de acordo com as interpretações clássicas, a probabilidade que o gato esteja vivo após o intervalo é 1/2.

Contudo, no estranho mundo da mecânica quântica, o átomo pode decair, mas até a caixa ser aberta e examinada, o gato estará em um estado intermediário de estar nem vivo nem morto. Ou, para o otimista que vive em você, ele pode ser considerado simultaneamente vivo e morto. Portanto, até que um exame tenha lugar, é impreciso assinalar um valor de verdade ou falsidade para a condição do gato estar vivo. Enquanto a lógica clássica é de pouco valor aqui, a lógica trivalente pode representar esta situação. Assinalando a condição do gato antes do exame o valor de I, o estado do gato é precisamente descrito.

2.6 Propriedades da Lógica Trivalente

Existem várias propriedades na lógica trivalente de Łukasiewicz, e algumas são listadas e provadas abaixo. Muitas delas também são válidas para outras estruturas lógicas trivalentes, mas as representações numéricas usadas na lógica de Łukasiewicz a tornam melhor adaptada como preparação para as estruturas lógicas que examinaremos nos capítulos 3 e 4.

Teorema 2.6.1. *Se A é uma variável na lógica trivalente de Łukasiewicz, então as seguintes fórmulas são satisfeitas:*
i) $A \wedge 1 = A$.
ii) $A \wedge 0 = 0$.
iii) $A \wedge 1/2 \leq 1/2$.
iv) $A \vee 1 = 1$.
v) $A \vee 0 = A$.
vi) $A \vee 1/2 \geq 1/2$.
vii) $A \wedge A = A$.
viii) $A \vee A = A$.
ix) $\neg\neg A = A$.

Demonstração.

i) Se $A = 1$, então $1 \wedge 1 = 1 = A$. Se $A = 1/2$, então $1/2 \wedge 1 = 1/2 = A$ e se $A = 0$, então $0 \wedge 1 = 0 = A$.

ii) Se $A = 1$, então $1 \wedge 0 = 0$. Se $A = 1/2$, então $1/2 \wedge 0 = 0$ e se $A = 0$, então $0 \wedge 0 = 0$.

iii) Se $A = 1$, então $1 \wedge 1/2 = 1/2 \leq 1/2$. Se $A = 1/2$, então $1/2 \wedge 1/2 = 1/2 \leq 1/2$ e se $A = 0$, então $0 \wedge 1/2 = 0 \leq 1/2$.

iv) Se $A = 1$, então $1 \vee 1 = 1$. Se $A = 1/2$, então $1/2 \vee 1 = 1$ e se $A = 0$, então $0 \vee 1 = 1$.

v) Se $A = 1$, então $1 \vee 0 = 1 = A$. Se $A = 1/2$, então $1/2 \vee 0 = 1/2 = A$ e se $A = 0$, então $0 \vee 0 = 0 = A$.

vi) Se $A = 1$, então $1 \vee 1/2 = 1 \geq 1/2$. Se $A = 1/2$, então $1/2 \vee 1/2 = 1/2 \geq 1/2$ e se $A = 0$, então $0 \vee 1/2 = 1/2 \geq 1/2$.

vii) Se $A = 1$, então $1 \wedge 1 = 1 = A$. Se $A = 1/2$, então $1/2 \wedge 1/2 = 1/2 = A$ e se $A = 0$, $0 \wedge 0 = 0 = A$.

viii) Se $A = 1$, então $1 \vee 1 = 1 = A$. Se $A = 1/2$, então $1/2 \vee 1/2 = 1/2 = A$ e se $A = 0$, $0 \vee 0 = 0 = A$.

ix) Para qualquer número x, $1 - (1 - x) = 1 - 1 + x = x$. □

Teorema 2.6.2. *Se A, B e C são variáveis na lógica trivalente de Łukasiewicz:*
i) $A \vee B = B \vee A$.
ii) $A \wedge B = B \wedge A$.
iii) $A \vee (B \vee C) = (A \vee B) \vee C$.
iv) $A \wedge (B \wedge C) = (A \wedge B) \wedge C$.

Em outras palavras, $\vee$ e $\wedge$ são associativos e comutativos na lógica trivalente.

Demonstração. Para provar estas expressões, nós confiaremos nas definições equivalentes onde $A \vee B = max\{A, B\}$ e $A \wedge B = min\{A, B\}$.

i) $A \vee B = max\{A, B\} = max\{B, A\} = B \vee A$.

ii) $A \wedge B = min\{A, B\} = min\{B, A\} = B \wedge A$.

iii) $A \vee \{B \vee C\} = max\{A, max\{B, C\}\} = max\{A, B, C\} = max\{max\{A, B\}, C\} = (A \vee B) \vee C$.

iv) $A \wedge \{B \wedge C\} = min\{A, min\{B, C\}\} = min\{A, B, C\} = min\{min\{A, B\}, C\} = (A \wedge B) \wedge C$. □

Teorema 2.6.3. *Se A, B e C são variáveis na lógica trivalente de Łukasiewicz:*
i) $(A \vee B) \wedge A = A$. *(Absorção)*
ii) $(A \wedge B) \vee A = A$. *(Absorção)*
iii) $A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$. *(Distribuição)*
iv) $A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$. *(Distribuição)*

Demonstração. Mais uma vez, estas fórmulas são facilmente verificadas usando as definições $A \vee B = max\{A, B\}$ e $A \wedge B = min\{A, B\}$.

i) Se $max\{A, B\} = A$, então a expressão é $A \wedge A$ que é igual a A. Se $max\{A, B\} = B$, então $min\{A, B\} = A$.

ii) Se $min\{A, B\} = A$, então $A \vee A = A$. Se $min\{A, B\} = B$, then $max\{A, B\} = A$.

iii) Esta prova é feita usando análise de casos.

Caso 1: $A \geq B \geq C$
$B \wedge C = C$ e então $A \vee C = A$, assim o lado esquerdo tem o valor A. $(A \vee B) = A$ e $(A \vee C) = A$, assim o lado direito é $A \wedge A = A$.

Caso 2: $A \geq C \geq B$
Este é similar ao caso 1, simplesmente reverta os papéis de B e C.

Caso 3: $B \geq A \geq C$
$(B \wedge C) = C$ e $A \vee C = A$, assim o lado esquerdo tem o valor A. $(A \vee B) = B$, $(A \vee C) = A$ e $A \wedge B = A$, assim o lado direito tem o valor A.

Caso 4: $C \geq A \geq B$
$(B \wedge C) = B$ e $A \vee B = A$, assim o lado esquerdo tem o valor A. $(A \vee B) = A$, $(A \vee C) = C$ e $A \wedge C = A$, assim o lado direito tem o valor A.

Caso 5: $B \geq C \geq A$
$(B \wedge C) = C$ and $A \vee C = C$, assim o lado esquerdo tem o valor C. $(A \vee B) = B$, $(A \vee C) = C$ e $B \wedge C = C$, assim o lado direito tem o valor C.

Caso 6: $C \geq B \geq A$
Este é similar ao caso 5, simplesmente intercambie os papéis de B e C.

iv) A prova é idêntica a da propriedade iii). □

Teorema 2.6.4. *Se A e B são variáveis na lógica trivalente de Łukasiewicz:*
i) $\neg(A \vee B) = \neg A \wedge \neg B$ *(De Morgan)*
ii) $\neg(A \wedge B) = \neg A \vee \neg B$ *(De Morgan)*

Demonstração.

i) A prova é feita por meio de análise de casos.

Caso 1: Se A ou B são iguais a 1, então o lado esquerdo é zero. Uma das duas negações à direita é zero, assim a conjunção à direita é zero.

Caso 2: Se o maior valor das variáveis é 1/2, então o lado esquerdo é 1/2. Se ambas são 1/2, então cada negação à direita é 1/2 e a conjunção é 1/2. Se uma é 1/2 e a outra é zero, então uma negação é 1/2 e a outra é 1, assim a conjunção é 1/2.

Caso 3: Tanto A quanto B são iguais a zero. Neste caso, a disjunção à esquerda é zero e a negação é 1. Ambas as negações à direita são um, assim a conjunção é 1.

ii) A prova é feita por meio de análise de casos.

Caso 1: Se um dos valores é zero, então a conjunção à esquerda é zero, assim o lado esquerdo é 1. Uma das negações no lado direito é um e a disjunção seria um.

Caso 2: Se uma das variáveis é 1/2, então a conjunção à esquerda é 1/2, assim o valor à esquerda é 1/2. Posto que uma das negações à direita é 1/2 e a outra é 1/2 ou zero, o valor à direita também é 1/2.

Caso 3: Ambas as variáveis são iguais a 1. Então a conjunção à esquerda é 1, assim o valor à esquerda é zero. Ambas as negações à direita são zero, assim a disjunção também é zero. □

2.7 Modus Ponens na Lógica Trivalente de Łukasiewicz

A regra de inferência modus ponens na lógica trivalente de Łukasiewicz pode ser interpretada da seguinte maneira. Se $p \to q$ e p são verdadeiras, então podemos concluir que q também é verdadeira. Lembre que a tabela verdade do conectivo $\to$ é

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

Tabela 2.12: Tabela-verdade para o operador condicional.

assim, MP pode ser interpretada como significando que tanto p quanto $p \to q$ são verdadeiras, o que nos restringe à primeira linha da tabela, onde q também é verdadeira.

MP pode ser modificada para situações na lógica trivalente de Łukasiewicz, mas agora nós somente temos que ser mais precisos em nossas definições de "verdade".

Definição 2.7.1. *A regra de inferência MP pode ser definida da seguinte maneira na lógica trivalente de Łukasiewicz.*

i) Se $p = 1$ *e* $p \to q$ *é 1 ou 1/2, então* q *tem o valor de* $p \to q$.
ii) Se $p = 1/2$ *e* $p \to q = 1$, *então* $q = 1$.
iii) Se $p = 1/2$ *e* $p \to q = 1/2$, *então* $q = 1/2$.

Os casos onde $p = 0$ ou $p \to q = 0$ não necessitam ser considerados, dado que modus ponens requer que ambas sejam "verdadeiras".

Note como isto coincide com as tabelas verdade para o conectivo $\to$ nas tabelas 2.4 e 2.5, mas não na tabela 2.3. Um teorema em lógica trivalente então também deve especificar os valores verdade sendo assinalados para a hipótese como uma consequência de qual definição dos conectivos está sendo usada. Teorias formais também podem ser criadas em lógica trivalente, a única complicação é adicionar material que especifica as constantes e torna precisa as regras de inferência.

Exemplo: O seguinte é uma teoria formal L_{3V} na lógica trivalente de Łukasiewicz.

a) O conjunto de símbolos em L_{3V} são $\{\neg, \vee, \wedge, \to, (,), 1, 0, 1/2, \text{A}, \text{B}, \ldots\}$.
b)
   1) Todos os símbolos em $\{1, 0, 1/2, A, B, \ldots\}$ são fbfs em L_{3V}.
   2) Se A e B são fbfs em L_{3V}, então também o são (A), $\neg A$, $A \vee B$, $A \to B$ e $A \wedge B$.
   3) Somente expressões que são formadas usando as regras (1) e (2) são fbfs em L_{3V}.
c) Se A e B são fbfs em L_{3V}, então os axiomas de L_{3V} são os seguintes:
   1) $A \to (A \vee B)$ tem um valor maior do que zero.
   2) $(A \wedge B) \to A$ tem um valor maior do que zero.
   3) $(A \wedge B) \to B$ tem um valor maior do que zero.
d) A única regra de inferência é modus ponens, onde $\to$ é definido na tabela 2.4.

O que segue são teoremas em L_{3V}.

a) De A diferente de zero podemos inferir $A \vee B$ com no mínimo o mesmo valor verdade de A.

Demonstração.

1) $(A \to (A \vee B)) > 0$. (Axioma 1 L_{3V})
2) A tem um valor maior do que zero. (Dado)

3)

i) Se $A = 1$, então aplicando MP_{3V}(modus ponens L_{3V}) em 1) e 2), podemos concluir que $(A \vee B) = 1$.

ii) Se $A = 1/2$, então aplicando MP_{3V} em 1) e 2), podemos conlcuir que $(A \vee B) = 1/2$.

□

b) De $A \wedge B$ diferente de zero podemos inferir A com o mesmo valor verdade de $A \wedge B$.

Demonstração.

1) $(A \wedge B \rightarrow A) > 0$. (Axioma 2 L_{3V})
2) A $\wedge$ B não é zero. (Dado)
3)

i) Se $A \wedge B = 1$, então nós podemos aplicar MP_{3V} em 1) e 2) para concluir que $A = 1$.

ii) ii) $SeA \wedge B = 1/2$, então nós podmeos aplicar MP_{3V} em 1) e 2) para concluir que $A = 1/2$.

□

2.8 Regras de Inferência na Lógica Trivalente de Łukasiewicz

Regras de inferência são escritas na forma

$$p_1, p_2, \ldots, p_k \vdash q$$

A regra de inferência modus ponens para a lógica clássica é escrita na forma

$$p \rightarrow q, p \vdash q$$

Contudo, esta notação deve ser alterada no reino da lógica trivalente, e cada regra de inferência terá uma qualificação diferente. Para começar, o princípio será que todas as proposições que servem de hipótese são maiores do que zero. Adicionalmente, a qualificação adicional será que a conclusão deve ter um valor maior ou igual aos valores de todas as expressões da hipótese. Isto é expresso na definição formal.

Definição 2.8.1. *A regra de inferência*

$$p_1, p_2, \ldots, p_k \vdash q$$

é válida na lógica trivalente de Łukasiewicz se sempre que cada uma das expressões $p_1, p_2, \ldots, p_k$ *é maior do que zero, a expressão* q *tem um valor que é diferente de zero. Algumas vezes também é útil clarificar o valor de* q.

Por exemplo, a partir da tabela de valores para o conectivo $\rightarrow$ na lógica trivalente, como definidos nas tabelas 2.4 e 2.5, é possível concluir que de $p \rightarrow q$ e p, podemos concluir q com um valor de no mínimo $1/2$. Este resultado não é válido para a definição contida na tabela 2.3.

Se ela é qualificada, a regra de inferência *eliminação and* também pode ser usada na lógica trivalente de Łukasiewicz.

Dado $p_1 \wedge p_2 \wedge \ldots \wedge p_k = v$, então p_i pode ser inferido para ter um valor $\geq v$.

A regra de inferência *introdução and* também tem um significado qualificado na lógica trivalente.

Dado que $p_1, p_2, \ldots, p_k$ são todos diferentes de zero, podemos inferir que $p_1 \wedge p_2 \wedge \ldots \wedge p_k$ também é diferente de zero.

A regra de inferência *introdução or* tem um significado qualificado na lógica trivalente.

Dado p, podemos inferir $p \vee q$ tendo um valor maior ou igual a p.

Nota: A regra de inferência *resolução unitária* não se aplica na lógica trivalente de Łukasiewicz, posto que se $p \vee q$ e $\neg q$ são diferentes de zero, então não é possivel concluir que p é diferente de zero. Use os valores de $p = 0$ e $q = 1/2$ para verificar isto.

Nota: A regra de inferência *modus tollens* também não se aplica na lógica trivalente para o conectivo $\rightarrow$ como definido na tabela 2.3. Se $p = 1$ e $q = 1/2$, então $p \rightarrow q = 1/2$, $\neg q = 1/2$ e $\neg p = 0$. Ela também não se aplica se a definição do conectivo $\rightarrow$ é a da tabela 2.4. Se $p = 1$ e $q = 1/2$, $p \rightarrow q = 1/2$ e $\neg p = 0$. Contudo, a regra de inferência *modus tollens* se aplica se a definição do conectivo $\rightarrow$ é o da tabela 2.5. Em todos os casos onde $p \rightarrow q$ e $\neg q$ são diferentes de zero, $\neg p$ é diferente de zero.

A regra de inferência *resolução* também se aplica na lógica trivalente de Łukasiewicz.

Se $p \rightarrow q$ e $q \rightarrow r$ são diferentes de zero, então $p \rightarrow r$ também é diferente de zero. Isto pode ser verificado pelo exame da tabela verdade para as três variáveis p, q e r e as três diferente definições do conectivo $\rightarrow$.

A definição de raciocínio formal na lógica trivalente de Łukasiewicz é similar às definições das regras de inferência.

Definição 2.8.2. *Faça $\Gamma = \{a_1, a_2, \ldots, a_k\}$ representar um conjunto de fbfs na lógica trivalente de Łukasiewicz. Uma fbf p é dita ser uma consequência de Γ se sempre for o caso que todos os elementos de Γ são diferentes de zero, então p também é diferente de zero.*

A notação para uma consequência na lógica trivalente é a mesma que a da lógica clássica.

$$a_1, a_2, \ldots, a_k \vdash p$$

2.9 Tautologias e Contradições na Lógica Trivalente de Łukasiewicz

A definição de tautologias e contradições na lógica trivalente é a natural.

Definição 2.9.1. *Uma proposição p em lógica trivalente é uma tautologia se ela tem o valor 1(T) para todos os valores das variáveis em p. Ela é uma contradição se ela tem o valor 0(F) para todos os valores das variáveis.*

Tautologias e contradições são muito menos comuns em lógica trivalente e isto é fácil de ver a partir do seguinte teorema.

Teorema 2.9.1. *Nenhuma expressão em uma lógica trivalente pode ser uma tautologia (contradição) se ela também não é uma tautologia (contradição) na lógica clássica.*

Demonstração. Posto que a lógica trivalente restrita aos valores de $1(T)$ e $0(F)$ é a lógica clássica, se uma expressão é verdadeira para todos os valores das três variáveis, ela dever também ser verdadeira para as variáveis sendo zero e um. □

Nota: Posto que $p \to p$, $p \leftrightarrow q$ e $p \vee \neg p$ são tautologias na lógica clássica mas não na lógica trivalente de Łukasiewicz, existem "menos" tautologias na lógica trivalente de Łukasiewicz. Dado que $p \vee 1$ é sempre 1 e $p \wedge 0$ é sempre zero, tautologias e contradições existem na lógica trivalente de Łukasiewicz. As outras lógicas trivalentes tem propriedades similares.

Dependendo das circunstâncias, poderemos estar interessados nos valores das expressões que estão acima de um certo valor ao invés de precisamente um.

Definição 2.9.2. *A expressão p na lógica trivalente de Łukasiewicz é dita ser uma meia tautologia se ela tem um valor maior ou igual a 1/2 para todos os valores das variáveis em p.*

Estas tautologias existem, dado que as fbfs $p \vee \neg p$, $p \to p$ e $p \leftrightarrow p$ todas tem valores que são maiores ou igual a $1/2$ para todos os valores de p.

Capítulo 3

Lógica Fuzzy

3.1 Definição dos Conectivos Básicos da Lógica Fuzzy

No final dos anos 60, Lofti A. Zadeh, um professor do Departamento de EE/CS na Universidade da Califórnia em Berkeley, formulou uma expansão das lógicas clássica e multivalorada conhecida como **lógica fuzzy**. O esquema usa a mesma ideia básica da probabilidade de ocorrência de um evento, o qual é um número entre 1.0 (certo de ocorrer) e 0.0 (certo de não ocorrer). Esta gradação de valores é aplicada à lógica, criando graus de verdade. A lógica fuzzy foi originalmente desenvolvida para ser aplicada em problemas de representação de conhecimento, sendo um sistema mais intuitivo para descrever eventos antes que eles aconteçam. Por exemplo, depois de amanhã seremos capazes de aplicar o valor de T ou F para a expressão "Nevará amanhã". Contudo, hoje nós não temos tal certeza, mas baseado na experiência e no conhecimento de padrões de tempo é possível dizer, "Existe 80% de chance que nevará amanhã".

A lógica fuzzy permite que valores entre 0.0 e 1.0 sejam aplicados às variáveis. Se o valor é 0.0, então a variável é considerada falsa, se o valor é 1.0, ela é considerada verdadeira. Para valores intermediários, tais como x = 0.67, nós usamos uma expressão como: Existe 67% de chance que x seja verdadeiro (Em aproximadamente dois terços do tempo x é verdadeiro).

A lógica fuzzy descreve mais acuradamente o mundo em que vivemos. Frases como provável, não provável, provavelmente, improvável, a maioria, poucos, usualmente e aproximadamente, são muito mais comuns do que os absolutos verdadeiro e falso.

As operações lógicas básicas de CS = $\{\wedge, \vee, \neg\}$ têm definições simples em lógica fuzzy.

$x \wedge y$ = o menor dos valores de x e y.
$x \vee y$ = o maior dos valores de x e y.
$\neg x = 1.0 - x$.

Exemplo: Se $x = 0.45$ e $y = 0.84$, então $x \wedge y = 0.45$, $x \vee y = 0.84$ e $\neg x = 1.0 - 0.45 = 0.55$.

Deve estar claro que se os valores das variáveis são restritos a 1.0 e 0.0, os conectivos fuzzy comportam-se do mesmo modo que os operadores clássicos, com 1.0 sendo equivalente a T e 0.0 sendo a equivalente a F. Se as restrições são as três possibilidades 1.0, 0.0 e 0.5, então o comportamento é o da lógica trivalente de Łukasiewicz.

Note as similaridades em como os conectivos $\vee$ e $\wedge$ são definidos em lógica fuzzy com as definições em lógica trivalente.

A lógica fuzzy tem encontrado uso em muitas áreas, tais como sistemas de controle, sistemas especialistas, máquinas de inferência e inteligência artificial. Muitos dispositivos, tais como automóveis, eletrodomésticos, elevadores, trens e brinquedos, contêm dispositivos embarcados que aplicam as propriedades da lógica fuzzy para definir suas regras de comportamento.

Muitas das expressões que são verdadeiras em lógica trivalente também são verdadeiras em lógica fuzzy.

Teorema 3.1.1. *Se A é uma variável em uma lógica fuzzy, então as fórmulas seguintes são satisfeitas:*
i) $A \wedge 1 = A$.
ii) $A \wedge 0 = 0$.
iii) $A \vee 1 = 1$.
iv) $A \vee 0 = A$.
v) $\neg\neg A = A$.

Demonstração. As provas todas se fiam nas definições dos conectivos.

i) Visto que $A \wedge 1 = min\{A, 1\}$ e $A \leq 1$, o resultado é imediato.
ii) Visto que $A \wedge 0 = min\{A,\ 0\}$ e $A \geq 0$, o resultado é imediato.
iii) Visto que $A \vee 1 = max\{A,\ 1\}$ e $A \leq 1$, o resultado é imediato.
iv) Visto que $A \vee 0 = max\{A,\ 0\}$ e $A \geq 0$, o resultado é imediato.
v) $\neg\neg A = 1 - (1 - x) = 1 - 1 + x = x = A$. □

A lógica fuzzy pode ser considerada uma lógica infinitamente valorada. Visto que o intervalo real de zero a um é não enumerável, o número de todos os possíveis valores para as variáveis em lógica fuzzy é infinito não enumerável.

Teorema 3.1.2. *Se A, B e C são variáveis em lógica fuzzy:*
i) $A \vee B = B \vee A$.
ii) $A \wedge B = B \wedge A$.
iii) $A \vee (B \vee C) = (A \vee B) \vee C$.
iv) $A \wedge (B \wedge C) = (A \wedge B) \wedge C$.

Em outras palavras, $\vee$ e $\wedge$ são comutativos e associativos em lógica fuzzy. As provas dos elementos do teorema são similares a aquelas do teorema 2.6.2.

Teorema 3.1.3. *Se A e B são variáveis em lógica fuzzy:*
i) $(A \vee B) \wedge A = A$ *(Absorção)*
ii) $(A \wedge B) \vee A = A$ *(Absorção)*
iii) $A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$ *(Distributividade)*
iv) $A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$ *(Distributividade)*

A prova destas expressões são similares a aquelas do teorema 2.6.3.

Teorema 3.1.4. *Se A e B são variáveis em lógica fuzzy:*
i) $\neg(\mathrm{A} \vee \mathrm{B}) = \neg\mathrm{A} \wedge \neg\mathrm{B}$ *(De Morgan)*
ii) $\neg(\mathrm{A} \wedge \mathrm{B}) = \neg\mathrm{A} \vee \neg\mathrm{B}$ *(De Morgan)*

Demonstração.

i) A prova é feita por meio de análise de casos.

Caso 1: $A \geq B$
Então $A \vee B = A$ e a negação no lado esquerdo é $1 - A$. Posto que $A \geq B$, $1 - A \leq 1 - B$, portanto, o mínimo no lado direito é $1 - A$.

Caso 2: $B \geq A$
Esta prova é idêntica a aquela do caso 1 com os papéis de A e B intercambiados.

ii) A prova é feita por meio de análise de casos.

Caso 1: $A \geq B$
Então $A \wedge B = B$ e a negação à esquerda é $1 - B$. Posto que $A \geq B$, $1 - B \geq 1 - A$, portanto, o máximo no lado direito é $1 - B$.

Caso 2: $B \geq A$
Esta prova é idêntica a aquela do caso 1 com os papéis de A e B intercambiados. □

3.2 Outros Conectivos em Lógica Fuzzy

Usando as equivalências que foram definidas no capitulo sobre lógica clássica, é possível construir conectivos análogos à implicação, igualdade lógica, ou exclusivo, negação conjunta e negação alternativa na lógica fuzzy.

Definição 3.2.1. *Se x e y são variáveis em lógica fuzzy:*
i) $x \rightarrow y = \neg x \vee y = max\{1 - x, y\}$.
ii) $x \leftrightarrow y = (\neg x \vee y) \wedge (\neg y \vee x) = min\{max\{1 - x, y\}, max\{1 - y, x\}\}$.
iii) $x \veebar y = (x \wedge \neg y) \vee (\neg x \wedge y) = max\{min\{x, 1 - y\}, min\{1 - x, y\}\}$.
iv) $x \downarrow y = \neg x \wedge \neg y = min\{1 - x, 1 - y\}$.
v) $x | y = \neg x \vee \neg y = max\{1 - x, 1 - y\}$.

Enquanto estas fórmulas fornecem um meio de expressar estes conectivos, o valor de algumas delas pode ser questionado. Por exemplo, em lógica clássica, o conectivo $\leftrightarrow$ simboliza a igualdade lógica e $\veebar$ a desigualdade lógica. Quando se lida com variáveis fuzzy, igualdade de valores seria uma coisa muito rara. Adicionalmente, é forçoso arguir que $min\{max\{1-x,y\},max\{1-y,x\}\}$ é uma representação de igualdade e $max\{min\{x,1-y\},min\{1-x,y\}\}$ é uma representação de desigualdade. Portanto, se os conectivos $\leftrightarrow$ e $\veebar$ são definidos em lógica fuzzy, definições alternativas geralmente são usadas.

3.3 Tautologias e Contradições em Lógica Fuzzy

Em lógica fuzzy, 1.0 é equivalente à verdade e 0.0 é equivalente à falsidade. Enquanto é possível ter expressões que são avaliadas para estes valores, na maioria dos casos uma expressão que é avaliada para um valor próximo de 1.0 ou 0.0 poderia ser considerada significante. Executando uma análise de probabilidades das circunstâncias, existem dois diferentes níveis de significância que são globalmente usados.

Definição 3.3.1. *Na análise probabilística de eventos, se a probabilidade dos eventos ocorrendo por acaso é menor do que* 0.05*, então o resultado é dito ser* ***estatisticamente significante****. Se a probabilidade de eventos ocorrendo por acaso é menor do que* 0.01*, então o resultado é dito ser* ***altamente significante****.*

Definição 3.3.2. *Se uma expressão fuzzy é sempre avaliada para um valor maior do que* 0.95*, então a expressão é dita ser uma* ***tautologia significante*** *. Uma expressão fuzzy que sempre é avaliada para um valor maior do que* 0.99 *é dita ser uma* ***tautologia altamente significante****. Se uma expressão sempre é avaliada para um valor menor do que* 0.05*, então ela é dita ser uma* ***contradição significante*** *, e se uma expressão é sempre avaliada para ser menor do que* 0.01*, então ela é dita ser uma* ***contradição altamente significante****. Se uma expressão sempre é avaliada em* 1.0*, ela é uma* ***tautologia absoluta*** *, se ela sempre é avaliada para* 0.0*, ela é uma* ***contradição absoluta*** *.*

Tautologias absolutas e contradições absolutas existem em lógica fuzzy, por exemplo, $p \vee 1 = 1$ e $p \wedge 0 = 0$.

3.4 Implementando os conectivos fuzzy em um programa de computador

É fácil implementar os conectivos da lógica fuzzy em um programa de computador. O que segue é uma implementação da lógica fuzzy na linguagem de programação Java, uma linguagem orientada a objetos que é amplamente usada para construir grandes programas. Visto que ela é orientada a objetos, ela é baseada em estruturas conhecidas como **classes**, assim nós começaremos com uma classe fuzzy que representará as variáveis em lógica fuzzy.

```
/* This class is the definition of the elements of a fuzzy logic.
Developed by Charles Ashbacher, December 30, 2000. */

public class fuzzy
{
    // This is the data value of the class.
    private float value;

    /* This is the constructor that is called when the new operator is
       used to create a fuzzy object. */
    public fuzzy(float invalue)
    {
        /* Do not use the value if it is not within the acceptable range. */
        if((invalue < 0.0) || (invalue > 1.0))
        {
            System.out.println("The input value is not within
                                the valid range");
            value = 0.0f;
        }
        else
        {
            value = invalue;
        }
    }

    // This function allows us to modify the value of the fuzzy object.
    public void setvalue(float invalue)
    {
        /* Do not use the value if it is not within the acceptable range. */
        if((invalue < 0.0) || (invalue > 1.0))
        {
            System.out.println("The input value is not within
                                the valid range");
            return;
        }

        value = invalue;
    }

    // This function allows us to obtain the value of the fuzzy object.
    public float getvalue()
    {
        return value;
    }
}
```

Esta classe pode então ser usada em um programa que contém funções que computam os valores dos conectivos fuzzy.

```
/* This program is designed to implement the basic connectives of fuzzy logic.
It was written by Charles Ashbacher 12/30/2000. */

public class Usingfuzzy
{
    // This function performs the fuzzy AND operation.
    public static fuzzy fuzzyand(fuzzy f1, fuzzy f2)
    {
        float x1 = f1.getvalue();
        float x2 = f2.getvalue();
        float invalue;

        if(x1 >= x2)
        {
            invalue = x2;
        }
        else
        {
            invalue = x1;
        }

        fuzzy f3 = new fuzzy(invalue);
        return f3;
    }

    // This function performs the fuzzy OR operation.
    public static fuzzy fuzzyor(fuzzy f1, fuzzy f2)
    {
        float x1 = f1.getvalue();
        float x2 = f2.getvalue();
        float invalue;

        if(x1 >= x2)
        {
            invalue = x1;
        }
        else
        {
            invalue = x2;
        }

        fuzzy f3 = new fuzzy(invalue);
        return f3;
    }
```

```java
/* Fuzzy implication is defined as max{1.0 - f1.value, f2.value}. */
public static fuzzy fuzzyimplication(fuzzy f1, fuzzy f2)
{
    float x1 = 1.0f - f1.getvalue();
    float x2 = f2.getvalue();
    float invalue;

    if(x1 >= x2)
    {
        invalue = x1;
    }
    else
    {
        invalue = x2;
    }

    fuzzy f3 = new fuzzy(invalue);
    return f3;
}

/* Fuzzy altdenial is defined as max{1.0 - f1.value, 1.0 - f2.value}. */
public static fuzzy fuzzyaltdenial(fuzzy f1, fuzzy f2)
{
    float x1 = 1.0f - f1.getvalue();
    float x2 = 1.0f - f2.getvalue();
    float invalue;

    if(x1 >= x2)
    {
        invalue = x1;
    }
    else
    {
        invalue = x2;
    }

    fuzzy f3 = new fuzzy(invalue);
    return f3;
}

/* Fuzzy joint denial is defined as min{1.0 - f1.value, 1.0 - f2.value}.*/
public static fuzzy fuzzyjointdenial(fuzzy f1, fuzzy f2)
{
    float x1 = 1.0f - f1.getvalue();
    float x2 = 1.0f - f2.getvalue();
    float invalue;

    if(x1 >= x2){
        invalue = x2;
    }
```

```
        else
        {
            invalue = x1;
        }

        fuzzy f3 = new fuzzy(invalue);
        return f3;
    }

    /* Fuzzy logical equivalence is defined as
       min{max{1.0 - f1.value, f2.value}, max{f1.value,1.0 - f2.value}}. */
    public static fuzzy fuzzyequivalence(fuzzy f1, fuzzy f2)
    {
        float f1\_value = f1.getvalue();
        float f2\_value = f2.getvalue();
        float OneMinusf1\_value=1.0f - f1\_value;

        float OneMinusf2\_value=1.0f - f2\_value;
        float invalue, firstentry, secondentry;

        if(OneMinusf1\_value >= f2\_value)
        {
            firstentry = OneMinusf1\_value;
        }
        else
        {
            firstentry = f2\_value;
        }

        if(OneMinusf2\_value >= f1\_value)
        {
            secondentry = OneMinusf2\_value;
        }
        else
        {
            secondentry = f1\_value;
        }

        if(firstentry >= secondentry)
        {
            invalue = secondentry; \\
        }
        else
        {
            invalue = firstentry;
        }

        fuzzy f3 = new fuzzy(invalue);
        return f3;
    }
```

```
/* Fuzzy logical exclusive or is defined as
   max{min{1.0 - f1.value, f2.value}, min{f1.value,1.0 - f2.value}}. */
public static fuzzy fuzzyxor(fuzzy f1, fuzzy f2)
{
    float f1\_value = f1.getvalue();
    float f2\_value = f2.getvalue();
    float OneMinusf1\_value = 1.0f - f1\_value;

    float OneMinusf2\_value = 1.0f - f2\_value;
    float invalue, firstentry,secondentry;

    if(OneMinusf1\_value >= f2\_value)
    {
        firstentry = f2\_value;
    }
    else
    {
        firstentry = OneMinusf1\_value;
    }

    if(OneMinusf2\_value >= f1\_value)
    {
        secondentry = f1\_value;
    }
    else
    {
        secondentry = OneMinusf2\_value;
    }

    if(firstentry >= secondentry)
    {
        invalue = firstentry;
    }
    else
    {
        invalue = secondentry;
    }

    fuzzy f3 = new fuzzy(invalue);
    return f3;
}

// This function performs fuzzy negation.
public static fuzzy fuzzynegate(fuzzy f1)
{
    float x1 = f1.getvalue();
    fuzzy f3 = new fuzzy(1.0f - x1);
    return f3;
}
```

```
    // This is the function where execution begins. The values passed into the
    // fuzzy objects are the values that they will have. All three of the
    // functions matching the connectives are used and the output is just
    // to demonstrate them in action.
    public static void main(String args[ ])
    {
        fuzzy f1 = new fuzzy(0.65f);
        fuzzy f2 = new fuzzy(0.32f);
        fuzzy f3;

        f3 = fuzzynegate(f1);
        System.out.println("The value of f3 after fuzzy negation is " +
                           f3.getvalue());

        f3 = fuzzyand(f1,f2);
        System.out.println("The value of f3 after fuzzy ANDing is " +
                           f3.getvalue());

        f3 = fuzzyor(f1,f2); \\
        System.out.println("The value of f3 after fuzzy ORing is " +
                           f3.getvalue());

        f3 = fuzzyimplication(f1,f2);
        System.out.println("The value of f3 after fuzzy implication is " +
                           f3.getvalue());

        f3 = fuzzyequivalence(f1,f2);
        System.out.println("The value of f3 after fuzzy equivalence is " +
                           f3.getvalue());

        f3 = fuzzyxor(f1, f2);
        System.out.println("The value of f3 after fuzzy exclusive OR is " +
                           f3.getvalue());

        f3 = fuzzyaltdenial(f1, f2);
        System.out.println("The value of f3 after fuzzy alternate denial is "
                           + f3.getvalue());

        f3 = fuzzyjointdenial(f1, f2);
        System.out.println("The value of f3 after fuzzy joint denial is " +
                           f3.getvalue());
    }
}
```

Quando este programa é executado, a saída é:

```
The value of f3 after fuzzy negation is 0.35000002
The value of f3 after fuzzy ANDing is 0.32
The value of f3 after fuzzy ORing is 0.65
```

```
The value of f3 after fuzzy implication is 0.35000002
The value of f3 after fuzzy equivalence is 0.35000002
The value of f3 after fuzzy exclusive OR is 0.65
The value of f3 after fuzzy alternate denial is 0.68
The value of f3 after fuzzy joint denial is 0.35000002
```

3.5 Regras de Inferência em Lógica Fuzzy

As regras de inferência em lógica fuzzy são similares a aquelas da lógica trivalente.

Contudo, existe uma diferença chave no modo no qual um valor numérico pode ser assinalado para alguns resultados.

Definição 3.5.1. *Uma regra de inferência*

$$p_1, p_2, \ldots, p_n \vdash q$$

*é **z-válida** em lógica fuzzy se sempre que cada uma das expressões $p_1, p_2, \ldots, p_k$ é maior ou igual a z, a expressão q tem um valor maior ou igual a z. Se a única conclusão é que q é maior do que zero, então ela é **diferente de zero válida**.*

Dada a definição do conectivo $\rightarrow$ para lógica fuzzy, a regra de inferência *modus ponens* não é aplicável. É possível para $p \rightarrow q$ e p serem diferentes de zero enquanto q é zero. Como foi demonstrado no capítulo 2, que não é necessariamente a última palavra em *modus ponens*, também é possível definir o conectivo $\rightarrow$ de modo que ele é $\neg p \vee q$ se $p \leq q$ e 0 se $p > q$. Se esta definição é usada, então se $p \rightarrow q$ e p são diferentes de zero, segue que q também é diferente de zero.

Se ela está propriamente qualificada, a regra de *eliminação-and* também pode ser usada em lógica fuzzy.

Dado $p_1 \wedge p_2 \wedge \ldots \wedge p_k = v$, então p_i pode ser inferida para ter um valor $\geq v$.

A regra de inferência *introdução-and* também tem um significado qualificado em lógica fuzzy.

Dado que p_1, p_2, ..., p_k são todos diferentes de zero ou maiores ou iguais a v, podemos inferir que $p_1 \wedge p_2 \wedge \ldots \wedge p_k$ também é maior ou igual a v.

A regra de inferência *introdução-or* também tem um significado qualificado em lógica fuzzy.

Dado p, podemos inferir $p \vee q$ tendo um valor maior ou igual a p.

A regra de inferência *resolução unitária* não se aplica em lógica fuzzy.

Dado que se $p \vee q$ e $\neg q$ são diferentes de zero, então não é possível concluir que p é diferente de zero. Use os valores de $p = 0$ e $q = 1/2$ para verificar isto.

A regra de inferência *modus tollens* também não se aplica na lógica fuzzy se o conectivo $\rightarrow$ é definido como $\neg p \vee q$.

Contudo, se a definição é $\neg p \vee q$ se $p \leq q$ e 0 se $p > q$, então, se $p \rightarrow q$ e $\neg q$ são diferentes de zero, então segue que p também é diferente de zero.

A regra de inferência resolução também se aplica em lógica fuzzy.

Se p → q e q → r são diferentes de zero, então p → r também é diferente de zero. Isto pode ser verificado pelo exame da tabela verdade para as três variáveis p, q e r.

A definição de raciocínio formal em lógica fuzzy é similar às definições para as regras de inferência.

Definição 3.5.2. *Faça* $\Gamma = \{a_1, a_2, \ldots, a_k\}$ *representar um conjunto de fbfs em lógica fuzzy. Uma fbf* p *é dita ser uma* ***consequência de*** $\mathbf{\Gamma}$ *se sempre que todos os elementos de* Γ *são diferentes de zero, então* p *também é diferente de zero.*

Definição 3.5.3. *Faça* $\Gamma = \{a_1, a_2, \ldots, a_k\}$ *representar um conjunto de fbfs em lógica fuzzy. Uma fbf* p *é dita ser uma* ***consequência aumentada de*** $\mathbf{\Gamma}$ *se* p *é uma consequência de* Γ *e o valor de* p *é maior ou igual a todos os elementos de* Γ*.*

A notação para uma consequência em lógica fuzzy é a mesma da lógica clássica:

$$a_1, a_2, \ldots, a_k \vdash p.$$

A notação para uma consequência aumentada em lógica fuzzy é:

$$a_1, a_2, \ldots, a_k \vdash^* p.$$

Exemplo: Dadas as regras de inferência na lógica fuzzy, as seguintes sentenças são todas inferências válidas.

$p, q \vdash p \wedge q.$
$p \wedge q \vdash p.$
$p \wedge q \vdash^* p.$
$p \vdash p \vee q.$

3.6 Lógica Modal com Variáveis Fuzzy

Lógica modal é uma forma de lógica onde as noções de necessidade e possibilidade estão inclusas. Uma verdade necessária é uma que é sempre verdadeira e uma verdade possível é uma que pode ser verdadeira. A diferença é que agora existe a inclusão de diferentes mundos possíveis. Um mundo é um conjunto de parâmetros descrevendo as circunstâncias das variáveis lógicas. As definições dos parâmetros são chamadas de uma **interpretação**. Consequentemente, uma verdade necessária é um que é verdadeira para todas as possíveis interpretações e uma verdade contingente é uma que é verdadeira no mínimo em uma interpretação.

Exemplo: Faça o universo de discurso ser os inteiros maiores do que zero. Suponha que definamos a operação de "adição mais k ($+k$)" como sendo o conjunto infinito de operações

$$m + kn = m + n + k \text{ para } m \text{ e } n \text{ inteiros e } k \geq 1.$$

Em outras palavras, a operação de adição mais k é a soma de dois inteiros mais k.

A declaração "A adição mais k é sempre comutativa" é um verdade necessária, posto que $m + kn = m + n + k = n + m + k = n + km$ para todos os valores de k.

Contudo, a declaração "$m + k0 = m$" é uma verdade contingente, posto que ela somente é verdade quando $k = 0$.

Definição 3.6.1. *Se A é uma expressão, então LA é lido "A é uma verdade necessária" e MA é lido "A é uma verdade contingente".*[1] *Claramente, para qualquer expressão A, se LA então MA.*

Às variáveis em lógica fuzzy podem ser assinaladas diferentes valores com base em diferentes circunstâncias, onde os diferentes conjuntos de circunstâncias fazem um mundo.

Definição 3.6.2. *Uma **estrutura modal** em lógica fuzzy é uma estrutura com as seguintes características:*
i) Um conjunto de constantes 0 *e* 1*, onde* 0 *é sempre* 0.0 *e* 1 *é* 1.0*.*
ii) Um conjunto de variáveis $V = \{x, y, z, \ldots\}$ que pode ter os valores no intervalo $[0.0, 1.0]$. Este conjunto pode ser finito ou infinito.
iii) $W = \{w1, w2, \ldots\}$ é um conjunto não vazio de mundos possíveis ou interpretações das variáveis e operadores. Este conjunto também pode ser finito ou infinito.
iv) Um conjunto de funções que aceita um elemento de V (a variável) e um elemento de W (o mundo) e associa um número entre 0.0 *e* 1.0 *(inclusive). Esta função associa valores para as variáveis neste mundo.*
v) Um conjunto de operadores lógicos que pode ser usado nas variáveis.
vi) Um conjunto de relações que aceita um elemento de W e um elemento do conjunto de operadores lógicos e associa uma função que define a interpretação daquele operador naquele mundo.
viii) Um conjunto de regras que define o conjunto de fbfs na estrutura modal.

[1] N. do T.: Simbolicamente, tem-sem $\Box A$ e $\Diamond A$ como representações para LA e MA, respectivamente.

Exemplo: A liga maior dos times de basebol nos Estados Unidos está dividida em duas ligas, a nacional e a americana. No fim de cada temporada um time de cada liga se junta na série mundial para decidir o campeonato.

O que segue é uma definição de uma estrutura modal em lógica fuzzy.

a) O conjunto de variáveis é $V = \{Cubs, Braves, \ldots, Yankees\}$*, um para cada liga maior de basebol. A cada variável é associada um valor que representa a probabilidade que ela aparecerá na série mundial.*
b) O conjunto de operadores lógicos é $\{\wedge, \vee, \neg\}$*.*
c) O conjunto de fbfs é definido do seguinte modo:
i) Cada elemento de V *ou* $\{0, 1\}$ *é uma fbf.*
ii) Se A *e* B *são fbfs, então* (A), $A \vee B$, $A \wedge B$ *e* $\neg A$ *são fbfs.*
iii) Somente expressões que podem ser formadas usando (i) e (ii) são fbfs.
d) Existe somente um mundo e ele é o conjunto de times na liga maior de basebol.
e) Se x *é um elemento de* V*, então* $\neg x = 1 - x$*.*
f) Se x *e* y *são elementos de* V*, então* $x \wedge y = 0.0$ *se os times estão na mesma liga e* $x \wedge y = min\{x, y\}$ *se* x *e* y *são times em diferentes ligas.*
g) Se x *e* y *são elementos de* V*, então* $x \vee y = x + y$ *se os times estão na mesma liga e* $x \vee y = max\{x, y\}$ *se os times estão em ligas diferentes.*

Um exemplo de uma função que associa valores às variáveis neste mundo é como segue:

Liga Nacional		Liga Americana	
Time	Valor	Time	Valor
Cubs	0.06	Yankees	0.16
Braves	0.09	Red Sox	0.10
Expos	0.04	Orioles	0.05
Mets	0.09	Devil Rays	0.02
Marlins	0.03	Blue Jays	0.03
Phillies	0.07	White Sox	0.09
Reds	0.06	Twins	0.06
Pirates	0.05	Indians	0.13
Cardinals	0.09	Tigers	0.02
Astros	0.10	Royals	0.03
Brewers	0.02	Mariners	0.12
DiamondBacks	0.12	Athletics	0.07
Dodgers	0.05	Angels	0.07
Giants	0.09	Rangers	0.05
Padres	0.06		
Rockies	0.04		

Onde a soma dos valores é um para cada liga. Isto é consistente com as regras da probabilidade, onde evento certo tem probabilidade um.

3.7 Lógica Temporal

Claramente, conforme a temporada de basebol progride, as probabilidades associadas aos times irão se alterar conforme suas performances. **Lógica temporal** é uma lógica onde os valores podem mudar conforme o tempo. Alterar a lógica modal dada na secção anterior de modo que ela seja temporal requer somente uma modificação da função que associa os valores às variáveis no mundo.

iv) Um conjunto de funções que aceita um elemento de V (a variável), um elemento de W (o mundo) e um parâmetro temporal t que associa um número entre 0.0 e 1.0 (inclusive). Esta função associa valores para as variáveis neste mundo.

Para o exemplo anterior, a função que associa as probabilidades deve ter um rótulo que lista o tempo quando as probabilidades são assinaladas. Operadores adicionais usados em lógica temporal refletem a passagem do tempo.

Definição 3.7.1. *Faça A ser uma expressão em uma lógica temporal.*
i) FA – A será verdadeira em algum tempo futuro.
ii) PA – A foi verdadeira em algum tempo passado.
iii) GA – A será verdadeira em todos os tempos futuros.
iv) HA – A foi sempre verdadeira no passado.

Capítulo 4

Lógica Neutrosófica

4.1 Definição da Lógica Neutrosófica

A **Lógica Neutrosófica** foi criada por Florentin Smarandache (1995) e é uma extensão/combinação de lógica fuzzy, lógica intuicionista, lógica paraconsistente e as lógicas trivalentes que usam um valor indeterminado. Em lógica neutrosófica, de um modo fácil, qualquer variável lógica x é descrita por uma tripla ordenada

$$x = (t, i, f)$$

onde t é o grau de verdade, f é o grau de falsidade e i é o nível de indeterminação.

Nota:

A) Para manter consistência com as lógicas clássica e fuzzy, existe o caso especial onde $t + i + f = 1$.

B) Para se referir à lógica intuicionista, que significa informação incompleta sobre uma variável, proposição ou evento, tem-se $t + i + f < 1$.

C) Para se referir à lógica paraconsistente, que significa fontes de informação contraditórias a respeito de uma mesma variável lógica, proposição ou evento, tem-se $t + i + f > 1$.

Uma definição alternativa usada por Smarandache é ter os três valores na tripla ordenada como inteiros maiores ou iguais a zero e $t + i + f$ < ou = ou > 100. Esta definição, é consistente, é claro, com porcentagens. Para nossos propósitos, o uso de números reais cuja soma vai até 1.0 é preferível, dado que isso é mais consistente com outras lógicas.

Uma **definição mais geral** desenvolvida à posteriori por Smarandache (1999-2002)[1] é: Faça T, I, F serem subconjuntos reais standard (padrão) ou non-standard (não padrão) do intervalo

[1] Smarandache, Florentin (2007). A Unifying Field in Logics: Neutrosophic Logic. Neutrosophy, Neutroosphic Set, Neutrosophic Probability and Statistics - 6th Edition, InfoLearnQuest (Link: http://fs.unm.edu/eBook-Neutrosophics2.pdf).

unitário non-standard $]^{-}0, 1^{+}[$[2], com

$$sup T = t\_sup, inf T = t\_inf,$$
$$sup I = i\_sup, inf I = i\_inf,$$
$$sup F = f\_sup, inf F = f\_inf,$$

e

$$n\_sup = t\_sup + i\_sup + f\_sup,$$
$$n\_inf = t\_inf + i\_inf + f\_inf.$$

Os conjuntos T, I, F não são necessariamente intervalos, mas podem ser quaisquer subconjuntos sub-unitários reais: discretos ou contínuos; simples elementos, finitos ou infinitos (enumeráveis ou não enumeráveis); união ou intersecção de vários subconjuntos, etc.

Eles também podem se sobrepor. O subconjuntos reais podem representar os erros relativos na determinação de t, i, f (neste caso os subconjuntos T, I, F são reduzidos a pontos).

> Estaticamente, T, I, F são subconjuntos. Mas dinamicamente, olhando para outra perspectiva, os componentes T, I, F são em cada instância dependentes de muitos parâmetros e, portanto, eles podem ser considerados funções vetoriais valoradas de conjuntos ou mesmo operadores.

Os parâmetros poder ser: tempo, espaço, etc. (alguns dos quais são parâmetros escondidos/ desconhecidos): T(t, s, ...), I(t, s, ...), F(t, s, ...), onde t = tempo, s = espaço, etc., o que permite à lógica neutrosófica ser usada em física quântica. O cálculo dinâmico neutrosófico pode também ser usado em psicologia e a lógica neutrosófica tenta refletir a dinâmica de coisas e ideias.

Por exemplo: A proposição "Amanhã estará chovendo" não significa uma estrutura de componentes de valores fixos; esta proposição pode ser dita 40% verdadeira, 50% indeterminada e 45% falsa no tempo time t_1; mas no tempo t_2 ela pode ter sido modificada para 50% verdadeira, 49% indeterminada e 30% falsa (de acordo com novas evidências, fontes, etc.); e amanhã, no tempo t_{145}, a mesma proposição pode ser 100% verdadeira, 0% indeterminada e 0% falsa (se amanhã realmente irá chover). Esta é a dinâmica: os valores verdade mudam de um tempo para outro.

Em outro exemplo: o valor verdade de uma proposição pode mudar de um lugar para outro. Por exemplo: a proposição "Está chovendo" é 0% verdadeira, 0% indeterminada e 100% falsa em Albuquerque (New Mexico), mas movendo-se para Las Cruces (New Mexico) os valores

[2]Existem duas notações usadas por diferentes autores (F. Smarandache, J. Dezert, Charles T. Le, A. Buller, J. Allen, M. Khoshnevisan, S. Singh, S. Bhattacharya, F. Liu, Gh. C. Dinulescu-Campina, C. Lucas, C. Gershenson) para o intervalo unitário não padrão: $]^{-}0, 1^{+}[$ e $\Vdash^{-} 0, 1^{+} \dashv\!|$; ambas estão corretas.

mudam e podem ser (1, 0, 0).

Também, os valores verdade dependem/mudam com relação ao observador (subjetividade é outro parâmetro das funções/operadores T, I, F). Por exemplo: "John is smart" pode ser (.35, .67, .60) de acordo com seu chefe, mas (.80, .25, .10) de acordo consigo mesmo, ou (.50, .20, .30) de acordo com sua secretária, etc.

T, I e F são chamados componentes neutrosóficos, representando respectivamente os valores de verdade, indeterminação e falsidade referentes à neutrosofia, lógica neutrosófica, conjunto neutrosófico, probabilidade neutrosófica, estatística neutrosófica.

Esta representação está próxima do raciocínio da mente humana. Ela caracteriza/pega a imprecisão no conhecimento ou inexatidões linguísticas percebidas por vários observadores (é por isto que T, I, F são subconjuntos - não necessariamente elementos únicos), incerteza devida a conhecimento incompleto ou aquisição de erros ou aleatoriedade (é por isso que o subconjunto I existe), e vagueza devida a falta de contorno nítidos ou limites (é por isto que T, I, F são subconjuntos e I existe).

Alguém tem de especificar os limites superior (x_sup) e inferior (x_inf) dos subconjuntos porque em muitos problemas surge a necessidade de os computar.

Definição 4.1.1. ***Lógica Neutrosófica*[3]*:*** *Uma lógica na qual cada proposição é estimada ter a porcentagem de verdade em um subconjunto T, a porcentagem de indeterminação em um subconjunto I, e a porcentagem de falsidade em um subconjunto F, é chamada Lógica Neutrosófica.*

Nós usamos um subconjunto para representar a verdade (ou a indeterminação e a falsidade) ao invés de um número somente porque em muitos casos não somos capazes de determinar exatamente as porcentagens de verdade e de falsidade, mas somente aproximações. Por exemplo, uma proposição pode ser 30-40% (de trinta a quarenta porcento) verdadeira e ser 60-70% (de sessenta a setenta porcento) falsa, e ainda pior: ser 30-40% (de trinta a quarenta porcento) ou 45-50% (de quarenta a cinquenta porcento) verdadeira (de acordo com vários analistas) e ser 60% ou ser 66-70% (de sessenta e seis a setenta porcento) falsa.

Os subconjuntos não são necessariamente intervalos, mas quaisquer conjuntos (discretos, contínuos, abertos ou fechados, intervalos semi-abertos ou semi-fechados, intersecções ou uniões dos conjuntos anteriores, etc.) em relação com a proposição dada. Um subconjunto poderia ter um elemento somente em casos especiais desta lógica.

Constantes: (T, I, F) valores-verdade, onde T, I, F são subconjuntos standard (padrão) ou non-standard (não padrão) do intervalo não padrão $]^-0, 1^+[$, onde $n_{inf} = inf\, T + inf\, I + inf\, F$

[3]Também chamada **Lógica Smarandache** - Dictionary of Computing (Dennis Howe, England).

$\geq {}^{-}0$, e $n_{sup} = sup\,T + sup\,I + sup\,F \leq 3^{+}$.
Fórmulas atômicas: a, b, c, ...
Fórmulas arbitrárias: A, B, C, ...

> "A lógica neutrosófica é um quadro formal tentando mensurar a verdade, indeterminação e falsidade". (F. Smarandache)

A hipótese de Smarandache é que **nenhuma teoria é livre de paradoxos**, porque imprecisão na linguagem, expressão metafórica, vários níveis ou meta-níveis de entendimento (ou interpretação) podem se sobrepor.

A vantagem de usar lógica neutrosófica é que esta lógica distingue entre verdade relativa, que é uma verdade somente em um ou poucos mundos, denotada por 1, e verdade absoluta, que é uma verdade em todos os mundos, denotada por 1^{+}. E similarmente, lógica neutrosófica distingue entre falsidade relativa, denotada por 0, e falsidade absoluta, denotada por ${}^{-}0$.

> Em lógica neutrosófica a soma dos componentes não é necessariamente 1, como nas lógicas clássica e fuzzy, mas qualquer número entre ${}^{-}0$ e 3^{+}, e isto permite à lógica neutrosófica a capacidade de lidar com paradoxos, proposições que são verdadeiras e falsas ao mesmo tempo: assim $\text{NL}_{(paradoxo)} = (1, 1, 1)$; a lógica fuzzy não pode fazer isto porque a soma dos componentes deve ser 1.

Observações: Neste livro estudaremos somente o caso especial onde os componentes T, I, F são subconjuntos reduzidos a um elemento cada, respectivamente t, i e f; também será ignorada a distinção entre verdade/falsidade/indeterminação relativa e absoluta. Este caso é então dividido em três subcasos:
a) Quando a soma dos componentes é $t + i + f = 1$ (lógicas clássica e fuzzy);
b) Quando a soma dos componentes é $t + i + f < 1$ (lógica intuicionista);
c) Quando a soma dos componentes é $t + i + f \geq 1$ (lógica paraconsistente).

Definição 4.1.2. *Um elemento de uma **Lógica Intuicionista Neutrosófica** (INL - Intuitionistic Neutrosophic Logic) é uma quádrupla (t, i, f, u) onde $t + i + f + u = 1.0$ e $u \geq 0.0$, t é o grau de verdade, i o valor de indeterminação, f o grau de falsidade e u é o grau para o qual as circunstâncias são desconhecidas.*

Definição 4.1.3. *Um elemento de uma **Lógica Paraconsistente Neutrosófica** (PNL - Paraconsistent Neutrosophic Logic) é uma tripla (t, i, f) onde $t + i + f \geq 1.0$.*

> **Nota:** É possível definir uma forma geral de lógica como um conjunto de quádruplas onde os valores são permitidos variar sobre grandes faixas. Contudo, para propósitos de simplificar o raciocínio automático em lógica neutrosófica, as três definições separadas são usadas.

Em nosso próximo livro dedicado à lógica neutrosófica estudaremos também o caso geral (quando T, I, F são subconjuntos em análise non-standard de $]^{-}0, 1^{+}[$), e faremos distinção entre verdade/falsidade/inderterminação relativa $\{1 \, \text{ou} \, 0\}$ e absoluta $\{1^{+} \, \text{ou} \, ^{-}0\}$, enquando a soma dos componentes inferior/superior satisfaz as seguinte desigualdade: $^{-}0 \leq inf\, T + inf\, I + inf\, F \leq sup\, T + sup\, I + sup\, F \leq 3^{+}$.

Exemplo: Se i é sempre zero e t e f devem ser zero ou um, então as variáveis são restritas às formas $(1,0,0)$ e $(0,0,1)$. O comportamento dos conectivos em CS para estes valores pode ser definido usando as seguintes tabelas verdade.

p	q	p ∧ q	p ∨ q	¬p
(1, 0, 0)	(1, 0, 0)	(1, 0, 0)	(1, 0, 0)	(0, 0, 1)
(1, 0, 0)	(0, 0, 1)	(0, 0, 1)	(1, 0, 0)	(0, 0, 1)
(0, 0, 1)	(1, 0, 0)	(0, 0, 1)	(1, 0, 0)	(1, 0, 0)
(0, 0, 1)	(0, 0, 1)	(0, 0, 1)	(0, 0, 1)	(1, 0, 0)

Tabela 4.1: Tabela verdade para os conectivos and, or e not.

Se adotarmos as abreviações $T = (1,0,0)$ e $F = (0,0,1)$, então está claro que este modelo é equivalente à lógica clássica.

Exemplo: Se os valores das variáveis em NL estão restritas a $(1,0,0)$, $(0,0,1)$ e $(1/2,0,1/2)$ e o comportamento dos conectivos é definido para ser o da lógica trivalente de Łukasiewicz, então o sistema é isomórfico com a lógica trivalente de Łukasiewicz.

Exemplo: Se i é sempre zero e t e f tem somente as restrições de estar entre zero e um, então os conectivos $\neg$, $\wedge$ e $\vee$ podem ser definidos do seguinte modo:

$\neg(t,0,f) = (f,0,t)$
$(t_1,0,f_1) \wedge (t_2,0,f_2) = (\min\{t_1,t_2\},0,\max\{f_1,f_2\})$
$(t_1,0,f_1) \vee (t_2,0,f_2) = (\max\{t_1,t_2\},0,\min\{f_1,f_2\})$

Teorema 4.1.1. *Se x é uma variável em uma lógica fuzzy, então o mapeamento $C : x \leftrightarrow (x,0,1-x)$ usando os operadores lógicos definidos no exemplo anterior é um isomorfismo entre a lógica fuzzy e uma subclasse de lógica neutrosófica onde i é sempre zero e os três elementos somam 1.*

Demonstração.

Se x e y são variáveis em uma lógica fuzzy, então eles podem ser mapeados para $(x,0,1-x)$ e $(y,0,1-y)$, respectivamente.

$C(\neg x) = C(1-x) = (1-x,0,x)$ e $\neg C(x) = \neg(x,0,1-x) = (1-x,0,x)$, logo $C(\neg x) = \neg C(x)$.

$C(x \wedge y) = C(\min\{x,y\}) = (\min\{x,y\}, 0, 1 - \min\{x,y\})$.
$C(x) \wedge C(y) = (x, 0, 1 - x) \wedge (y, 0, 1 - y) = (\min\{x,y\}, 0, \max\{1 - x, 1 - y\})$.

Dado que $1 = \min\{x,y\} + \max\{1 - x, 1 - y\} = \min\{x,y\} + (1 - \min\{x,y\})$, então:
$\max\{1 - x, 1 - y\} = (1 - \min\{x,y\})$ e $C(x \wedge y) = C(x) \wedge C(y)$.

$C(x \vee y) = C(\max\{x,y\}) = (\max\{x,y\}, 0, 1 - \max\{x,y\})$
$C(x) \vee C(y) = (x, 0, 1 - x) \vee (y, 0, 1 - y) = (\max\{x,y\}, 0, \min\{x,y\})$

Mais uma vez, dado que $1 = \max\{x,y\} + (1 - \max\{x,y\}) = \max\{x,y\} + \min\{x,y\}$, segue que: $1 - \max\{x,y\} = \min\{x,y\}$ e $C(x \vee y) = C(x) \vee C(y)$.

Consequentemente, o mapeamento C é um homomorfismo.

Se $(x, 0, 1 - x) = (x_1, 0, 1 - x_1)$, então pela definição de tripla ordenada, $x = x_1$. Portanto, C é um mapeamento de um para um e é um isomorfismo. $\square$

Dos exemplos anteriores, deve estar claro que a lógica neutrosófica é uma generalização das lógicas clássica, trivalente e fuzzy. A progressão destas lógicas é resumida na Figura 4.1.

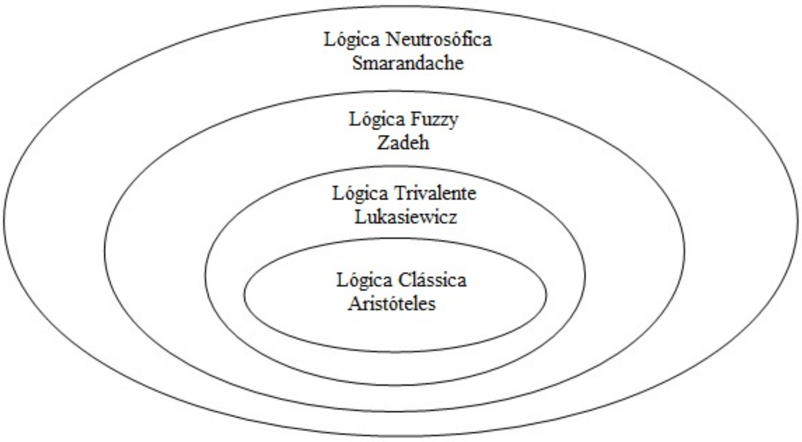

Figura 4.1: Relação entre as lógicas neutrosófica, fuzzy, trivalente e clássica.

De fato, a lógica neutrosófica pode ser considerada uma representação mais acurada do mundo em que estamos. Poucas coisas são absolutas. Na verdade, a matemática é uma das poucas áreas onde alguma coisa é conhecida com certeza. A entrada indeterminada permite reconhecer que os valores dados para as entradas verdadeiro e falso sejam comumente não conhecidas com certeza.

Existem outras definições similares de variáveis em NL que podem representar mais precisamente circunstâncias mais especializadas. Considere a tomada de uma amostragem estatística tal como a pesquisa de opinião pública onde duas opiniões são possíveis. Como uma consequência das leis da estatística, tais pesquisas tem erros de amostragem. O resultado de tais amostras são geralmente apresentados na forma: $x\%$ primeira resposta, $y\%$ segunda resposta, com uma margem de erro de $\pm k\%$.

Geralmente, $x + y = 100$. O que isto significa é que a primeira resposta poderia ser tão alta quanto x+k%, ao passo que a segunda resposta teria o valor correspondente de y − k%, ou que a primeira resposta poderia ser tão baixa quanto x − k% com a segunda sendo y + k%.

Exemplo: Um grupo de pessoas é consultado em relação às suas preferência para o candidato A versus o candidato B. Suponha que as respostas são 55% favorável ao candidato A e 45% ao candidato B, e que a pesquisa tenha um erro de mais ou menos 5%. Isto significa que os reais valores são algo entre 60% e 50% para A e entre 40% e 50% para B.

Tais situações podem ser manipuladas com a seguinte modificação da definição dos itens em uma NL.

Definição 4.1.4. *Uma Lógica Neutrosófica onde a indeterminação é uma margem de erro (NLE) é uma tripla* (t, i, f) *onde* $t + f = 1.0$*,* $i \leq \min\{t, f\}$*,* $t + i \leq 1.0$ *e* $f + i \leq 1.0$*. Isto é interpretado para significar que os valores para a verdade podem estar na faixa [t − i, t + i] e os valores para a falsidade devem estar em [f + i, f − i]. As limitações sobre os valores são necessários de modo que não existe probabilidade que seja menor do que zero ou maior do que um.*

Assim alguém pode obter $t + i + f > 1$.

Exemplo: Se uma pesquisa é feita e os resultados são que 45% das pessoas pesquisadas aprovam o trabalho que o presidente está fazendo com um erro de mais ou menos 5%, a expressão NLE correspondente seria (.45, .05, .55), o que significa que esta pesquisa indica que o percentual atual de pessoas com um índice de aprovação favorável é algo na faixa de 0.40 a 0.50.

Se os valores das variáveis podem mudar de acordo com as circunstâncias, então as triplas de NL podem ser usadas para representar circunstâncias adicionais. Por exemplo, se nós estamos dando uma expressão da forma $x > 0$, então não é possível assinalar a ela um valor verdade por causa do valor desconhecido de x.

Contudo, o valor $(0, 1, 0)$ pode ser assinalado em NL, dado que o valor de verdade é completamente indeterminado. Contudo, uma vez que seja assinalado um valor a x, então o valor será $(1, 0, 0)$ ou $(0, 0, 1)$. Adicionalmente, se conhecemos algo sobre o conjunto universo de possíveis valores para x, então os valores podem ser assinalados para t e f. Por exemplo, se o conjunto de discurso é o conjunto de todos os números reais e a escolha de x é feita aleatoriamente, podemos arguir que o valor é $(0.50, 0.0, 0.50)$. Isto poderia ser interpretado que as chances são meio a meio.

Esta situação ocorre em ciência da computação. Se uma variável é declarada sem ter um valor assinalado (por exemplo: `int m`), então o conteúdo da variável é considerado lixo, significando que o valor é apenas o que foi deixado da operações anteriores. O atual status da variável poderia ser descrito pelo valor $(0.0, 1.0, 0.0)$.

Em mecânica quântica é possível para partículas, tais como elétrons, possuir um entre dois possíveis estados, "up" e "down". Os estados estão colocados entre aspas porque a propriedade de spin de um elétron é diferente daquela estamos familiarizados. Contudo, até o spin do elétron ser examinado, ele está no estado de incerteza, sendo considerado 1/2 up e 1/2 down ou indeterminado. Se usarmos a interpretação meio a meio, então a representação NL seria $(1/2, 0, 1/2)$. Contudo, se o considerarmos indeterminado, então a representação NL seria $(0, 1, 0)$. Uma vez que o estado do spin seja examinado, então a representação seria (1, 0, 0) ou (0, 0, 1), onde um é mapeado para um spin "up" e outro para um spin "down".

4.2 Conectivos Lógicos em Lógica Neutrosófica

Definição 4.2.1. *Faça (t_1, i_1, f_1) e (t_2, i_2, f_2) serem elementos de NL onde a soma dos elementos da tripla é 1.0. Os conectivos lógicos $\{\neg, \wedge, \vee\}$ podem ser definidos do seguinte modo:*

$$\neg(t_1, i_1, f_1) = (f_1, i_1, t_1)$$
$$(t_1, i_1, f_1) \wedge (t_2, i_2, f_2) = (t = \min\{t_1, t_2\}, i = 1 - (t + f), f = \max\{f_1, f_2\})$$
$$(t_1, i_1, f_1) \vee (t_2, i_2, f_2) = (t = \max\{t_1, t_2\}, i = 1 - (t + f), f = \min\{f_1, f_2\})$$

Visto que existem outros modos de definir os conectivos, nós notaremos este conjunto por NL_1.

Estas definições são consistentes com nossas noções comuns de indeterminação. Se a um evento é assinalado um valor indeterminado de i, então a negação não pode ter um valor indeterminado que é diferente. Se fosse menos indeterminado, então os valores t e f deveriam ter valores maiores, e se é mais indeterminado, então os valores de t e f seriam menores. A troca dos valores de t e f é consistente como a negação é definida em outras estruturas lógicas.

Se nós temos duas triplas em NL_1, então o valor de t para uma operação de conjunção é o menor dos dois valores de t. O valor de f para uma operação de conjunção é o máximo dos dois valores de f. Para a operação de disjunção o valor de t é o máximo dos dois valores de t e o valor de f deve ser o mínimo dos valores de f.

Esta interpretação dos conectivos $\wedge$ e $\vee$ é consistente em como verdadeiro e falso são interpretados na lógica clássica, lógica trivalente de Lukasiewicz e lógica fuzzy. O valor falso é dominante para o conectivo $\wedge$ e o valor verdadeiro é dominante para o conectivo $\vee$.

Uma definição alternativa dos conectivos $\vee$, $\wedge$ e $\neg$, que usa o valor indeterminado como valor dominante, é consistente com a dominância de M (ou valor sem significado) na lógica de Bochvar.

Definição 4.2.2. *Faça* (t_1, i_1, f_1) *e* (t_2, i_2, f_2) *serem elementos de NL. Os conectivos lógicos* $\{\neg, \wedge, \vee\}$ *podem ser definidos do seguinte modo:*

$$\begin{aligned}
\neg(t_1, i_1, f_1) &= (f_1, i_1, t_1) \\
(t_1, i_1, f_1) \wedge (t_2, i_2, f_2) &= (1 - \max\{i_1, i_2\} - \min\{1 - \max\{i_1, i_2\}, \max\{f_1, f_2\}\}, \\
&\quad \max\{i_1, i_2\}, \min\{1 - \max\{i_1, i_2\}, \max\{f_1, f_2\}\}) \\
(t_1, i_1, f_1) \vee (t_2, i_2, f_2) &= (\min\{1 - \max\{i_1, i_2\}, \max\{t_1, t_2\}\}, \\
&\quad \max\{i_1, i_2\}, 1 - \max\{i_1, i_2\} - \min\{1 - \max\{i_1, i_2\}, \max\{t_1, t_2\}\})
\end{aligned}$$

Em outras palavras, para a conjunção o valor dominante é o indeterminado, seguido pelo valor de falsidade. Para a disjunção, o valor dominante é o indeterminado, seguido pelo valor de verdade.

Os elementos de NL com estas definições dos conectivos será referido por NL_2.

Definição 4.2.3. *Faça* (t_1, i_1, f_1, u_1) *e* (t_2, i_2, f_2, u_2) *serem elementos de* INL. *Os conectivos lógicos* $\{\neg, \wedge, \vee\}$ *podem ser definidos do seguinte modo:*

$$\begin{aligned}
\neg(t_1, i_1, f_1, u_1) &= (f_1, i_1, t_1, u_1) \\
(t_1, i_1, f_1, u_1) \wedge (t_2, i_2, f_2, u_2) &= (t = \min\{t_1, t_2\}, i = \min\{i_1, i_2\}, \\
&\quad f = \max\{f_1, f_2\}, u = 1 - t - i - f) \\
(t_1, i_1, f_1, u_1) \vee (t_2, i_2, f_2, u_2) &= (t = \max\{t_1, t_2\}, i = \min\{i_1, i_2\}, \\
&\quad f = \min\{f_1, f_2\}, u = 1 - t - i - f)
\end{aligned}$$

A definição de negação é consistente com a negação em outras formas de lógica, em que ela é simplesmente a reversão dos valores dos componentes de verdade e falsidade. Também faz sentido que os elementos que representam indeterminação e incerteza não sejam alterados por uma negação.

Esta definição também é consistente com a dominação de verdadeiro e falso em lógica clássica, na qual falso domina na conjunção e verdadeiro domina na disjunção. Com esta definição os valores indeterminados são reduzidos e os valores desconhecidos aumentados.

Os elementos de INL com estas definições dos conectivos serão referenciado por INL_1.

Definição 4.2.4. *Faça* (t_1, i_1, f_1, u_1) *e* (t_2, i_2, f_2, u_2) *serem elementos de* INL. *Os conectivos lógicos* $\neg, \wedge, \vee$ *podem ser definidos do seguinte modo:*

$$\begin{aligned}
\neg(t_1, i_1, f_1, u_1) &= (f_1, i_1, t_1, u_1) \\
(t_1, i_1, f_1, u_1) \wedge (t_2, i_2, f_2, u_2) &= (t = \min\{t_1, t_2\}, i = 1 - t - f - u, f = \max\{f_1, f_2\}, u = \min\{u_1, u_2\}) \\
(t_1, i_1, f_1, u_1) \vee (t_2, i_2, f_2, u_2) &= (t = \max\{t_1, t_2\}, i = 1 - t - f - u, f = \min\{f_1, f_2\}, u = \min\{u_1, u_2\})
\end{aligned}$$

Esta definição também mantém a mesma consistência da negação com outras formas de lógica.

O valor de verdade ainda é dominante na disjunção e falso também o é na conjunção. Com esta definição os valores desconhecidos tendem a serem reduzidos com o correspondente aumento no valor indeterminado.

Os elementos de INL com estas definições dos conectivos serão referenciados por INL_2.

Definição 4.2.5. *Faça* (t_1, i_1, f_1) *e* (t_2, i_2, f_2) *serem elementos de* PNL. *Os conectivos lógicos* $\{\neg, \wedge, \vee\}$ *podem ser definidos do seguinte modo:*

$$\begin{aligned} \neg(t_1, i_1, f_1) &= (f_1, i_1, t_1) \\ (t_1, i_1, f_1) \wedge (t_2, i_2, f_2) &= (t = \min\{t_1, t_2\}, i = \max\{i_1, i_2\}, f = \max\{f_1, f_2\}) \\ (t_1, i_1, f_1) \vee (t_2, i_2, f_2) &= (t = \max\{t_1, t_2\}, i = \max\{i_1, i_2\}, f = \min\{f_1, f_2\}) \end{aligned}$$

Mais uma vez, a definição de negação é consistente com a das outras lógicas. O princípio de dominação da falsidade na conjunção e da verdade na disjunção também são mantidas, com a diferença primária de outras lógicas sendo a dominação simultânea do valor indeterminado.

Os elementos de PNL com estas definições dos conectivos serão referenciados por PNL_1.

Teorema 4.2.1. *Se* (t_1, i_1, f_1) *e* (t_2, i_2, f_2) *são elementos em* NL_1, *então:*

(i) $\neg(t_1, i_1, f_1)$

(ii) $(t_1, i_1, f_1) \wedge (t_2, i_2, f_2)$

(iii) $(t_1, i_1, f_1) \vee (t_2, i_2, f_2)$

também são elementos em NL_1. *Portanto,* NL_1 *é fechada sob estas definições das operações lógicas.*

Demonstração.

(i) Se $t_1 + i_1 + f_1 = 1.00$, então $f_1 + i_1 + t_1$ também é 1.00.

(ii) Pela escolha do termo indeterminado, somente é necessário provar que $\min\{t_1, t_2\} + \max\{f_1, f_2\} \leq 1$. Assuma que $(\min\{t_1, t_2\} + \max\{f_1, f_2\}) > 1$. O valores min e max não podem ser da mesma tripla, pois se assim o fosse, a soma dos valores na tripla teria um valor maior do que 1.0. Consequentemente, o mínimo vem de uma tripla e o máximo vem de outra. Sem perda de generalidade, assuma que $\min\{t_1, t_2\} = t_1$ e $\max\{f_1, f_2\} = f_2$. Visto que $t_1 + f_2 > 1.00$, segue que $t_2 < t_1$. Contudo, isto contradiz o resultado que $t_1 \leq t_2$ e significa que a assunção $(\min\{t_1, t_2\} + \max\{f_1, f_2\}) > 1$ é incorreta. Desta feita, $(\min\{t_1, t_2\} + \max\{f_1, f_2\}) \leq 1$ e $\wedge$ é fechada.

(iii) Pela escolha do termo indeterminado, somente é necessário provar que $\max\{t_1, t_2\} + \min\{f_1, f_2\} \leq 1$. A prova é similar àquela de (ii) e é omitida. □

Teorema 4.2.2. *Se* (t_1, i_1, f_1) *e* (t_2, i_2, f_2) *são elementos em* NL_2*, então:*

(i) $\neg(t_1, i_1, f_1)$

(ii) $(t_1, i_1, f_1) \wedge (t_2, i_2, f_2)$

(iii) $(t_1, i_1, f_1) \vee (t_2, i_2, f_2)$

também são elementos em NL_2*. Consequentemente,* NL_2 *é fechada sob estas definições das operações lógicas.*

Demonstração. Similar a aquela do teorema **??**. □

Teorema 4.2.3. *Se* (t_1, i_1, f_1) *e* (t_2, i_2, f_2) *são elementos em* INL_1*,* INL_2 *e* PNL_1*, então:*

i) INL_1 *é fechada sob os conectivos* $\{\neg, \wedge, \vee\}$.

ii) INL_2 *é fechada sob os conectivos* $\{\neg, \wedge, \vee\}$.

iii) PNL_1 *é fechada sob os conectivos* $\{\neg, \wedge, \vee\}$.

Demonstração.

(i) Faça (t_1, i_1, f_1, u_1) e (t_2, i_2, f_2, u_2) serem elementos de INL_1. Se $t_1 + i_1 + f_1 + u_1 = 1.0$, então alterar a ordem não terá efeito algum. Consequentemente, INL_1 é fechada com relação a $\neg$. Visto que $\min\{t_1, t_2\} + \min\{i_1, i_2\} + \max\{f_1, f_2\} \leq 1.0$, segue da escolha de u que $t + i + f + u = 1.0$ e INL_1 é fechada com relação a $\wedge$. Um argumento similar pode ser usado para verificar que INL_1 é fechada com relação a $\vee$.

(ii) Faça (t_1, i_1, f_1, u_1) e (t_2, i_2, f_2, u_2) serem elementos de INL_2. A prova do fechamento de $\neg$ é a mesma de INL_1. As provas para o fechamento de $\wedge$ e $\vee$ são similares às de INL_1. Simplesmente troque os papéis de i e u.

(iii) Se $t_1 + i_1 + f_1 \leq 1.0$ e $t_2 + i_2 + f_2 \geq 1.0$, então escolhendo o máximo de duas das três posições deve levar à soma que também é maior ou igual a 1.0. □

Definição 4.2.6. *Dadas duas expressões* A *e* B *em* NL_1*, usamos* $\mathrm{A} =_{\mathrm{NL1}} \mathrm{B}$ *para significar que* A *e* B *sempre possuem as mesmas triplas de valores para os assinalamentos de seus componentes comuns. O símbolo (singleton) de igualdade = será usado para significar que os valores dos três elementos das triplas são iguais. A expressão* $\mathrm{A} =_{\mathrm{NL_2}} \mathrm{B}$ *significará a mesma coisa em* NL_2*, bem como* $=_{\mathrm{INL_1}}$ *em* INL_1*,* $=_{\mathrm{INL_2}}$ *em* INL_2 *e* $=_{\mathrm{PNL_1}}$ *em* PNL_1*.*

Teorema 4.2.4.
a) Faça A ser uma expressão em NL_1. Então $\neg\neg A =_{NL_1} A$.
b) Faça A ser uma expressão em NL_2. Então $\neg\neg A =_{NL_2} A$.
c) Faça A ser uma expressão em INL_1. Então $\neg\neg A =_{INL_1} A$.
d) Faça A ser uma expressão em INL_2. Então $\neg\neg A =_{INL_2} A$.
e) Faça A ser uma expressão em PNL_1. Então $\neg\neg A =_{PNL_1} A$.

Demonstração. Pelas definições, em todos os casos o operador $\neg$ intercambia os valores de t e f. Consequentemente, intercambiá-los duas vezes traz de volta os valores originais. □

Teorema 4.2.5. *Se A e B são expressões em NL_1, então:*
i) $A \wedge B =_{\mathrm{NL}_1} B \wedge A$.
ii) $A \vee B =_{\mathrm{NL}_1} B \vee A$.

Em outras palavras, $\wedge$ e $\vee$ são comutativas.

Demonstração. Faça $A = (t_1, i_1, f_1)$ e $B = (t_2, i_2, f_2)$.

i) $(t_1, i_1, f_1) \wedge (t_2, i_2, f_2) =$
$= (\min\{t_1, t_2\}, 1 - (\min\{t_1, t_2\} + \max\{f_1, f_2\}), \max\{f_1, f_2\}) =$
$= (\min\{t_2, t_1\}, 1 - (\min\{t_2, t_1\} + \max\{f_2, f_1\}), \max\{f_2, f_1\}) =$
$= (t_2, i_2, f_2) \wedge (t_1, i_1, f_1)$

Portanto, $\wedge$ é comutativa.

ii) $(t_1, i_1, f_1) \vee (t_2, i_2, f_2) =$
$= (\max\{t_1, t_2\}, 1 - (\max\{t_1, t_2\} + \min\{f_1, f_2\}), \min\{f_1, f_2\}) =$
$= (\max\{t_2, t_1\}, 1 - (\max\{t_2, t_1\} + \min\{f_2, f_1\}), \min\{f_2, f_1\}) =$
$= (t_2, i_2, f_2) \vee (t_1, i_1, f_1)$

Consequentemente, $\vee$ é comutativa. □

Teorema 4.2.6. *Se A e B são expressões em NL_2, então:*
i) $A \wedge B =_{NL_2} B \wedge A$.
ii) $A \vee B =_{NL_2} B \vee A$.

Em outras palavras, $\wedge$ e $\vee$ são comutativas em NL_2.

Demonstração. Similar ao do teorema 4.2.5. □

Teorema 4.2.7. *Se A e B são expressões em INL_1, INL_2 ou PNL_1, então:*
i) $A \wedge B =_{INL_1} B \wedge A$.
ii) $A \vee B =_{INL_1} B \vee A$.

iii) $A \wedge B =_{INL_2} B \wedge A$.
iv) $A \vee B =_{INL_2} B \vee A$.
v) $A \wedge B =_{PNL_1} B \wedge A$.
vi) $A \vee B =_{PNL_1} B \vee A$.

Em outras palavras, $\wedge$ *e* $\vee$ *são comutativas em* NL_1*,* NL_2 *e* PNL_1*.*

Demonstração. Dado que $\max\{x,y\} = \max\{y,x\}$ e $\min\{x,y\} = \min\{y,x\}$ para todo x e y, os resultados decorrem diretamente das definições dos operadores. □

Teorema 4.2.8. *Se* A*,* B *e* C *são expressões em* NL_1*, então:*
i) $(A \wedge B) \wedge C =_{NL_1} A \wedge (B \wedge C)$.
ii) $(A \vee B) \vee C =_{NL_1} A \vee (B \vee C)$.

Demonstração. Faça $A = (t_1, i_1, f_1), B = (t_2, i_2, f_2)$ e $C = (t_3, i_3, f_3)$.

$$
\begin{aligned}
&i)[(t_1,i_1,f_1) \wedge (t_2,i_2,f_2)] \wedge (t_3,i_3,f_3) = \\
&= [(\min\{t_1,t_2\}, 1-(\min\{t_1,t_2\} + \max\{t_1,t_2\}), \max\{t_1,t_2\}] \wedge (t_3,i_3,f_3) = \\
&= (\min\{\min\{t_1,t_2\},t_3\}, 1-(\min\{\min\{t_1,t_2\},t_3\} \\
&+ \max\{\max\{t_1,t_2\},t_3\}), \max\{\max\{t_1,t_2\},t_3\}) = \\
&= (\min\{t_1,\min\{t_2,t_3\}\}, 1-(\min\{t_1,\min\{t_2,t_3\}\} \\
&+ \max\{t_1,\max\{t_2,t_3\}\}), \max\{t_1,\max\{t_2,t_3\}\}) = \\
&= (t_1,i_1,f_1) \wedge [(t_2,i_2,f_2) \wedge (t_3,i_3,f_3)]
\end{aligned}
$$

Consequentemente, o conectivo $\wedge$ é associativo.

ii) A prova que $\vee$ é associativo é similar e não será feita. □

Corolário: *Se* $\{A_1, A_2, A_3, \ldots, A_n\}$ *é um conjunto de elementos em* NL_1*, então* $A_1 \vee A_2 \vee A_3 \vee \ldots \vee A_n = (t, i, f)$*, onde* $t = \max\{t_1, t_2, t_3, \ldots, t_n\}$ *e* $f = min\{f_1, f_2, f_3, \ldots, f_n\}$*.*

Demonstração. Usando os resultados do teorema 4.2.8, podemos associar de qualquer modo que desejarmos sem mudar o valor. Consequentemente, associaremos tudo à esquerda. Em outras palavras, escreveremos expressões na forma: $(\ldots,(A_1 \vee A_2) \vee A_3) \vee \ldots) \vee A_n)$.

A prova será por indução sobre $k \geq 3$.

Base: $k = 3$.

$$((A_1 \vee A_2) \vee A_3) = (t, i, f)$$

onde $t = \max\{\max\{t_1, t_2\}, t_3\} = \max\{t_1, t_2, t_3\}$ e $f = \min\{\min\{f_1, f_2\}, f_3\} = \min\{f_1, f_2, f_3\}$.

Passo indutivo: Assuma que para $k > 3$

$$(\ldots(A_1 \vee A_2) \vee A_3) \vee \ldots) \vee A_k) = (t, i, f)$$

onde $t = \max\{t_1, t_2, t_3, \cdots, t_k\}$ e $f = \min\{f_1, f_2, f_3, \cdots, f_k\}$.

Então, por definição: $(\ldots(A_1 \vee A_2) \vee A_3) \vee \ldots) \vee A_k) \vee A_{k+1}) = (t_s, i_s, f_s)$, onde $t_s = \max\{t, t_{k+1}\} = \max\{\max\{t_1, t_2, \ldots, t_k\}, t_{k+1}\} = \max\{t_1, t_2, \ldots, t_k, t_{k+1}\}$ e $f_s = \min\{f, f_{k+1}\} = \min\{\min\{f_1, f_2, \ldots, f_k\}, f_{k+1}\} = \min\{f_1, f_2, \ldots, f_k, f_{k+1}\}$.

Consequentemente, pelo princípio de indução matemática, a fórmula é verdadeira para todo n. □

Corolário: *Se* $\{A_1, A_2, A_3, \ldots, A_n\}$ *é um conjunto de elementos em* NL_1*, então* $A_1 \wedge A_2 \wedge A_3 \wedge \ldots \wedge A_n = (t, i, f)$*, onde* $t = min\{t_1, t_2, t_3, \ldots, t_n\}$ *e* $f = \max\{f_1, f_2, f_3, \ldots, f_n\}$.

Demonstração. Similar a aquela do corolário anterior e não será feita. □

Teorema 4.2.9. *Se A, B e C são expressões em* NL_2*, então:*

i) $(\mathrm{A} \wedge \mathrm{B}) \wedge \mathrm{C} =_{\mathrm{NL}_2} \mathrm{A} \wedge (\mathrm{B} \wedge \mathrm{C})$.
ii) $(\mathrm{A} \vee \mathrm{B}) \vee \mathrm{C} =_{\mathrm{NL}_2} \mathrm{A} \vee (\mathrm{B} \vee \mathrm{C})$.

Demonstração.

Faça $A = (t_1, i_1, f_1)$, $B = (t_2, i_2, f_2)$ e $C = (t_3, i_3, f_3)$.

i) Claramente, $\max\{\max\{i_1, i_2\}, i_3\} = \max\{i_1, \max\{i_2, i_3\}\}$, assim o termo do meio será o mesmo em ambos os lados. Consequentemente, se os valores dos terceiros termos podem ser provados ser iguais, a prova está completa.

No lado esquerdo, o valor da terceira entrada de $A \wedge B$ é $\min\{1 - \max\{i_1, i_2\}, \max\{f_1, f_2\}\}$, portanto, a terceira entrada de $(A \wedge B) \wedge C$ é $\min\{1 - \max\{i_1, i_2, i_3\}, \max\{\min\{1 - \max\{i_1, i_2\}, \max\{f_1, f_2\}\}, f_3\}\}$.

No lado direito, a terceira entrada de $B \wedge C$ é $\min\{1 - \max\{i_2, i_3\}, \max\{f_2, f_3\}\}$, portanto, a terceira entrada $A \wedge (B \wedge C)$ é $\min\{1 - \max\{i_1, i_2, i_3\}, \max\{\min\{1 - \max\{i_2, i_3\}, \max\{f_2, f_3\}\}, f_1\}\}$.

Se o menor valor de cada expressão é $1 - \max\{i_1, i_2, i_3\}$, então os valores são os mesmos. Consequentemente, assuma que o valor de uma das terceiras entradas não é $1 - \max\{i_1, i_2, i_3\}$, o que significa que ela é f_1, f_2, ou f_3, dado que o mínimo de $1 - \max\{i_1, i_2, i_3\}$ e $1 - \max\{i_2, i_3\}$ é $1 - \max\{i_1, i_2, i_3\}$.

Caso 1: Assuma que $f_1 = \min\{1 - \max\{i_1, i_2, i_3\}, \max\{\min\{1 - \max\{i_1, i2\}, \max\{f_1, f_2\}\}, f_3\}$.

Isto significa que $f_1 \geq f_2, f_1 \leq 1 - \max\{i_1, i_2\}, f1 \geq f_3$ e $f_1 \leq 1 - \max\{i_1, i_2, i_3\}$. Com estas restrições, f_1 é maior ou igual a qualquer valor que emerge de $min\{1 - \max\{i_2, i_3\}, \max\{f_2, f_3\}\}$, assim $f_1 = \max\{\min\{1 - \max\{i_2, i_3\}, \max\{f_2, f_3\}\}, f_1\}$. Visto que $f_1 < 1 - \max\{i_1, i_2, i_3\}$, o terceiro termo à direita também deve ser f_1.

Caso 2: Assuma que $f_2 = \min\{1 - \max\{i_1, i_2, i_3\}, \max\{\min\{1 - \max\{i_1, i_2\}, \max\{f_1, f_2\}\}, f_3\}$.

Isto significa que $f_2 \geq f_1$, $f_2 \leq 1 - \max\{i_1, i_2\}, f_2 \geq f_3$ e $f_2 \leq 1 - \max\{i_1, i_2, i_3\}$. Com estas restrições, $f_2 = \min\{1 - \max\{i_2, i_3\}, \max\{f_2, f_3\}\}$, $f_2 = \max\{f_2, f_1\}$ e $f_2 = \min\{1 - \max\{i_1, i_2, i_3\}, f_2\}$. Consequentemente, o valor do terceiro termo no lado direito também é f_2.

As provas dos casos remanescentes são todas similares e serão omitidas. Consequentemente, a conjunção é associativa em NL_2.

ii) As provas para a associatividade da disjunção são similares, o teste somente é feito no primeiro termo da tripla ao invés de ser no terceiro. □

Corolário: Se $A_1, A_2, A_3, \ldots, A_n$ é um conjunto de elementos em NL_2, então

$$A_1 \wedge A_2 \wedge A_3 \wedge \ldots \wedge A_n = (t, i, f),$$

onde $i = \max\{i_1, i_2, i_3, \ldots, i_n\}$ e $f = \min\{1 - \max\{i_1, i_2, i_3, \ldots, i_n\}, \max\{f_1, f_2, f_3, \ldots, f_n\}\}$.

Corolário: Se $\{A_1, A_2, A_3, \ldots, A_n\}$ é um conjunto em NL_2, então $A_1 \vee A_2 \vee A_3 \vee \ldots \vee A_n = (t, i, f)$, onde $i = \max\{i_1, i_2, i_3, \ldots, i_n\}$ e $t = \min\{1 - i, \max\{t_1, t_2, t_3, \ldots, t_n\}\}$.

Os quatro corolários antecedentes permitem a definição dos quantificadores universal e existencial em NL_1 e NL_2.

Definição 4.2.7. *Faça*

$$\bigwedge_{i \in U} (t_i, i_i, f_i)$$

ser uma conjunção de todos os valores no conjunto universo de uma implementação de NL_1*. Se* $t_{\min} = \{t_1, \ldots, t_k, \ldots\}$ *e* $f_{\max} = \{f_1, \ldots, f_k, \ldots\}$*, então o valor do quantificador universal em* NL_1 *é*

$$(t_{\min}, 1 - (t_{\min} + f_{\max}), f_{\max}).$$

Se os valores de $t_{\min}$ ou $f_{\max}$ estão dentro de certos valores, podemos definir o quantificador universal para NL_1 como aquele onde todos os valores da expressão são maiores ou menores do que um valor específico. Na maioria dos casos, qualificações são colocados nos resultados.

Exemplo: Se NL_K é uma implementação da Lógica Neutrosófica onde os conectivos são definidos como os da NL_1 e o valor indeterminado dos elementos de NL_K nunca é maior do que 0.10, então a declaração universal

$$\forall x \in NL_K(x \vee \neg x)$$

deveria ter um valor verdade de no mínimo 0.45.

Definição 4.2.8. *Faça*

$$\bigvee_{i \in U} (t_i, i_i, f_i)$$

ser a disjunção de todos os valores do conjunto universo de uma implementação de NL_1*. Se*

$$t_{\max} = \max\{t_1, \ldots, t_k, \ldots\} \text{ e } f_{\min} = \min\{f_1, \ldots, f_k, \ldots\}$$

então o valor do quantificador existencial em NL_1 *é dado por*

$$(t_{\max}, 1 - (t_{\max} + f_{min}), f_{min}).$$

Definição 4.2.9. *Faça*

$$\bigwedge_{i \in U} (t_i, i_i, f_i)$$

ser a conjunção de todos os valores do conjunto universo de uma implementação de NL_2*. Se*

$$i_{\max} = \max\{i_1, \ldots, i_k, \ldots\} \text{ e } f_{\max} = \max\{f_1, \ldots, f_k, \ldots\},$$

então o valor do quantificador universal em NL_2 *é*

$$(1 - (i + f), i = i_{\max}, f = min\{1 - i_{max}, f_{max}\}).$$

Definição 4.2.10. *Faça*

$$\bigvee_{i \in U} (t_i, i_i, f_i)$$

ser a disjunção de todos os valores no conjunto universo de uma implementação de NL_2*. Se*

$$t_{\max} = \max\{t_1, \dots, t_k, \dots\} \ e \ i_{\max} = \max\{i_1, \dots, i_k, \dots\},$$

então o valor do quantificador existencial em NL_2 é dado por

$$(t = \min\{1 - i_{\max}, t_{\max}\}, i = i_{\max}, f = 1 - (t + i)).$$

Teorema 4.2.10. *Se A, B e C são elementos em INL_1, INL_2 ou PNL_1, então*

i) $(A \wedge B) \wedge C =_{INL_1} A \wedge (B \wedge C)$.
ii) $(A \vee B) \vee C =_{INL_1} A \vee (B \vee C)$.
iii) $(A \wedge B) \wedge C =_{INL_2} A \wedge (B \wedge C)$.
iv) $(A \vee B) \vee C =_{INL_2} A \vee (B \vee C)$.
v) $(A \wedge B) \wedge C =_{PNL_1} A \wedge (B \wedge C)$.
vi) $(A \vee B) \vee C =_{PNL_1} A \vee (B \vee C)$.

Demonstração. Visto que $\min\{\min\{x, y\}, z\} = \min\{x, \min\{y, z\}\}$ e $\max\{\max\{x, y\}, z\} = \max\{x, \max\{y, z\}\}$ para todos os números x, y e z, todos os resultados seguem diretamente das definições dos operadores. □

Corolário: Se $\{A_1, A_2, A_3, \dots, A_n\}$ é o conjunto de elementos em INL_1, então

$$A_1 \wedge A_2 \wedge A_3 \wedge \dots \wedge A_n = (t, i, f, u)$$

onde $t = \max\{t_1, \dots, t_n\}$, $i = \min\{i_1, \dots, i_n\}$, $f = \min\{f_1, \dots, f_n\}$ e $u = 1 - t - i - f$.

Corolário: Se $\{A_1, A_2, A_3, \dots, A_n\}$ é o conjunto de elementos em INL_1, então

$$A_1 \vee A_2 \vee A_3 \vee \dots \vee A_n = (t, i, f, u)$$

onde $t = \max\{t_1, \dots, t_n\}$, $i = \min\{i_1, \dots, i_n\}$, $f = \min\{f_1, \dots, f_n\}$ e $u = 1 - t - i - f$.

Corolário: Se $\{A_1, A_2, A_3, \dots, A_n\}$ é o conjunto de elementos em INL_2, então

$$A_1 \wedge A_2 \wedge A_3 \wedge \dots \wedge A_n = (t, i, f, u)$$

onde $t = \min\{t_1, \dots, t_n\}$, $f = \max\{f_1, \dots, f_n\}$, $u = \min\{u_1, \dots, u_n\}$ e $i = 1 - t - f - u$.

Corolário: Se $\{A_1, A_2, A_3, \dots, A_n\}$ é o conjunto de elementos em INL_2, então

$$A_1 \vee A_2 \vee A_3 \vee \dots \vee A_n = (t, i, f, u)$$

onde $t = \max\{t_1, \dots, t_n\}$, $f = \min\{f_1, \dots, f_n\}$, $u = \min\{u_1, \dots, u_n\}$ e $i = 1 - t - f - u$.

Corolário: Se $\{A_1, A_2, A_3, \ldots, A_n\}$ é um conjunto de elementos em PNL_1, então

$$A_1 \wedge A_2 \wedge A_3 \wedge \ldots \wedge A_n = (t, i, f)$$

onde $t = \min\{t_1, \ldots, t_n\}$, $i = \max\{i_1, \ldots, i_n\}$ e $f = \max\{f_1, \ldots, f_n\}$.

Corolário: Se $\{A_1, A_2, A_3, \ldots, A_n\}$ é um conjunto de elementos em PNL_1, então

$$A_1 \vee A_2 \vee A_3 \vee \ldots \vee A_n = (t, i, f)$$

onde $t = \max\{t_1, \ldots, t_n\}$, $i = \max\{i_1, \ldots, i_n\}$ e $f = \min\{f_1, \ldots, f_n\}$.

Cada um destes seis corolários podem ser provados por indução no número de elementos na conjunção ou disjunção.

Armado destes corolários, é possível definir os quantificadores universal e existencial para INL_1, INL_2 e PNL_1.

Definição 4.2.11. *Faça*

$$\bigwedge_{i \in U} (t_i, i_i, f_i)$$

ser a conjunção de todos os valores no conjunto universal de uma implementação de INL_1*. Se*

$$t_{\min} = \min\{t_1, \ldots, t_k, \ldots\}, i_{\min} = \min\{i_1, \ldots, i_k, \ldots\} \text{ e } f_{\max} = \max\{f_1, \ldots, f_k, \ldots\}$$

então o valor do quantificador universal em INL_1 *é*

$$(t_{\min}, i_{\min}, f_{\max}, u = 1 - t_{\min} - i_{\min} - f_{\max}).$$

Definição 4.2.12. *Faça*

$$\bigwedge_{i \in U} (t_i, i_i, f_i, u_i)$$

ser uma conjunção de todos os valores no conjunto universo de uma implementação de INL_2*. Se*

$$t_{\min} = \min\{t_1, \ldots, t_k, \ldots\}, f_{\max} = \max\{f_1, \ldots, f_k, \ldots\} \text{ e } u_{\min} = \min\{u_1, \ldots, u_k, \ldots\}$$

então o valor do quantificador universal em

$$NL_2$$

é

$$(t_{\min}, 1 - t_{\min} - f_{\max} - u_{\min}, f_{\max}, u_{\min}).$$

Definição 4.2.13. *Faça*

$$\bigwedge_{i \in U} (t_i, i_i, f_i)$$

ser uma conjunção de todos os valores no conjunto universo de uma implementação de PNL_1. *Se*

$$t_{\min} = \min\{t_1, \ldots, t_k, \ldots\}, i_{\max} = \max\{i_1, \ldots, i_k, \ldots\} \text{ e } f_{\max} = \max\{f_1, \ldots, f_k, \ldots\}$$

então o valor do quantificador universal em PNL_1 *é*

$$(t_{\min}, i_{\max}, f_{\max})$$

Definição 4.2.14. *Faça*

$$\bigvee_{i \in U} (t_i, i_i, f_i, u_i)$$

ser uma disjunção de todos os valores no conjunto universo de uma implementação de INL_1. *Se*

$$t_{\max} = \max\{t_1, \ldots, t_k, \ldots\}, i_{\min} = \min\{i_1, \ldots, i_k, \ldots\} \text{ e } f_{\min} = \min\{f_1, \ldots, f_k, \ldots\}$$

então o valor do quantificador existencial em INL_1 *é*

$$(t_{\max}, i_{\min}, f_{\min}, u = 1 - t_{\max} - i_{\min} - f_{\min}).$$

Definição 4.2.15. *Faça*

$$\bigvee_{i \in U} (t_i, i_i, f_i, u_i)$$

ser uma disjunção de todos os valores no conjunto universo de uma implementação de INL_2. *Se*

$$t_{\max} = \max\{t_1, \ldots, t_k, \ldots\}, f_{\min} = \min\{f_1, \ldots, f_k, \ldots\} \text{ e } u_{\min} = \min\{u_1, \ldots, u_k, \ldots\}$$

então o valor do quantificador existencial em INL_2 *é*

$$(t_{\max}, 1 - t_{\max} - f_{\min} - u_{\min}, f_{\min}, u_{\min}).$$

Definição 4.2.16. *Faça*

$$\bigvee_{i \in U} (t_i, i_i, f_i)$$

ser uma disjunção de todos os valores no conjunto universo de uma implementação de PNL_1. *Se*

$$t_{\max} = \max\{t_1, \ldots, t_k, \ldots\}, i_{\max} = \max\{i_1, \ldots, i_k, \ldots\} \text{ e } f_{\min} = \min\{f_1, \ldots, f_k, \ldots\}$$

então o valor do quantificador existencial em PNL_1 *é*

$$(t_{\max}, i_{\max}, f_{\min}).$$

4.3 Propriedades Algébricas das Lógicas Neutrosóficas

Definição 4.3.1. *Um semigrupo é um conjunto S e uma operação binária $\circ$ que é fechada com relação a S e associativa em S.*

Teorema 4.3.1.
i) O conjunto das triplas (t, i, f) onde $t+i+f = 1.0$ com a definição $\wedge$ de NL_1 é um semigrupo.
(ii) O conjunto das triplas (t, i, f) onde $t + i + f = 1.0$ com a definição $\vee$ de NL_1 é um semigrupo.
(iii) O conjunto de triplas (t, i, f) onde $t+i+f = 1.0$ com a definição $\wedge$ de NL_2 é um semigrupo.
(iv) O conjunto de triplas (t, i, f) onde $t+i+f = 1.0$ com a definição $\vee$ de NL_2 é um semigrupo.
(v) O conjunto de 4-tuplas (t, i, f, u) onde $t + i + f + u = 1.0$ com a definição $\wedge$ de INL_1 é um semigrupo.
(vi) O conjunto de 4-tuplas (t, i, f, u) onde $t + i + f + u = 1.0$ com a definição $\vee$ de INL_1 é um semigrupo.
(vii) O conjunto de 4-tuplas (t, i, f, u) onde $t + i + f + u = 1.0$ com a definição de $\wedge$ de INL_2 é um semigrupo.
(viii) O conjunto de 4-tuplas (t, i, f, u) onde $t + i + f + u = 1.0$ com a definição de $\vee$ de INL_2 é um semigrupo.
(ix) O conjunto de triplas (t, i, f) onde $t + i + f \geq 1.0$ com a definição $\wedge$ de PNL_1 é um semigrupo.
(x) O conjunto de triplas (t, i, f) onde $t+i+f \geq 1.0$ com a definição $\vee$ de PNL_1 é um semigrupo.

Demonstração. Visto que um semigrupo é um conjunto fechado sob a operação onde a propriedade associativa é mantida para todos os elementos, todos os dez resultados são consequências diretas dos teoremas anteriores, onde a propriedade associativa foi verificada para todas as lógicas. □

Teorema 4.3.2. *Se A e B são expressões em NL_1, então*
(i) $(A \vee B) \wedge A =_{NL_1} A$.
(ii) $(A \wedge B) \vee A =_{NL_1} A$.

Em outras palavras, $\wedge$ e $\vee$ satisfazem as leis de absorção.

Demonstração. Faça $A = (t_1, i_1, f_1)$ e $B = (t_2, i_2, f_2)$. A prova é pelo exame dos elementos das triplas uma por vez.

(i) $(A \vee B) \wedge A =_{NL1} A$.

Os valores dos primeiros elementos das triplas do lado esquerdo são computadas pela seguinte fórmula: $\min\{\max\{t_1, t_2\}, t_1\}$.

Caso 1: Examinando o primeiro elemento da tripla.

Os valores dos primeiros elementos das triplas do lado esquerdo são computadas pela seguinte fórmula: $\min\{\max\{t_1, t_2\}, t_1\}$.

Subcaso 1-a: $t_1 > t_2$. Então, $\min\{\max\{t_1, t_2\}, t_1\} = \min\{t_1, t_1\} = t_1$.

Subcaso 1-b: $t_1 < t_2$. Então, $\min\{\max\{t_1, t_2\}, t_1\} = \min\{t_2, t_1\} = t_1$.

Consequentemente, o primeiro elemento das triplas são os mesmos em todos os casos.

Caso 2: Examinando o terceiro elemento da tripla.

Os valores dos terceiros elementos das triplas no lado esquerdo são computados pela seguinte fórmula: $\max\{\min\{f_1, f_2\}, f_1\}$.

Subcaso 2-a: $f_1 > f_2$. Então, $\max\{\min\{f_1, f_2\}, f_1\} = \max\{f_2, f_1\} = f_1$.

Subcaso 2-b: $f_1 < f_2$. Então, $\max\{\min\{f_1, f_2\}, f_1\} = \max\{f_1, f_1\} = f_1$.

Consequentemente, os terceiros elementos das triplas são todos os mesmos em todos os casos.

Visto que os primeiro e terceiro elementos das triplas são todos os mesmos em todos os casos, a fórmula é verdadeira.

(ii) $(A \wedge B) \vee A =_{NL_1} A$.

Caso 1: Examinando o primeiro elemento da tripla.

Os valores dos primeiros elementos das triplas no lado esquerdo são computadas pela seguinte fórmula: $\max\{\min\{t_1, t_2\}, t_1\}$.

Subcaso 1-a: $t_1 > t_2$. Então, $\max\{\min\{t_1, t_2\}, t_1\} = \max\{t_2, t_1\} = t_1$.

Subcaso 1-b: $t_1 < t_2$. Então, $\max\{\min\{t_1, t_2\}, t_1\} = \max\{t_1, t_1\} = t_1$.

Consequentemente, os primeiros elementos das triplas são todos os mesmos em todos os casos.

Caso 2: Examinando o terceiro elemento das tripla.

Os valores dos terceiros elementos das triplas no lado esquerdo são computados pela seguinte fórmula: $\max\{\min\{f_1, f_2\}, f_1\}$.

Subcaso 1: $f_1 > f_2$. Então, $\max\{\min\{f_1, f_2\}, f_1\} = \max\{f_2, f_1\} = f_1$.

Subcase 2: $f_1 < f_2$. Então, $\max\{\min\{f_1, f_2\}, f_1\} = \max\{f_1, f_1\} = f_1$.

Consequentemente, os terceiros elementos das triplas são os mesmos para todos os casos.

Visto que o primeiro e terceiro elementos das triplas são os mesmos em todos os casos, a fórmula é verdadeira. □

Teorema 4.3.3. *Se A e B são expressões em NL_2, então*
(i) $(A \vee B) \wedge A \neq_{NL_2} A$.
(ii) $(A \wedge B) \vee A \neq_{NL_2} A$.

Em outras palavras, $\vee$ e $\wedge$ não satisfazem a leis de absorção.

Demonstração. Faça $A = (t_1, i_1, f_1)$ e $B = (t_2, i_2, f_2)$.

i) Se $i_2 > i_1$, então o termo do meio de $A \vee B$ é i_2 e o termo do meio de $(A \vee B) \wedge A$ também é i_2. Consequentemente, $(A \vee B) \wedge A \neq_{NL_2} A$.

ii) Se $i_2 > i_1$ então o termo do meio de $A \wedge B$ é i_2 e o termo do meio de $(A \wedge B) \vee A$ é também i_2. Consequentemente, $(A \vee B) \wedge A \neq_{NL_2} A$. □

Teorema 4.3.4. *NL_1 e NL_2 não satisfazem a lei de absorção.*

Demonstração. Faça $A = (0.1, 0.2, 0.3, 0.4)$ e $B = (0.1, 0.1, 0.3, 0.5)$ serem elementos de NL_1. Então:

$$A \vee B = (0.1, 0.1, 0.3, 0.5) \text{ e}$$
$$\underbrace{(0.1, 0.1, 0.3, 0.5)}_{A \vee B} \wedge \underbrace{(0.1, 0.2, 0.3, 0.4)}_{A} = (0.1, 0.1, 0.3, 0.5) \neq A$$
$$A \wedge B = (0.1, 0.1, 0.3, 0.5) \text{ e}$$
$$\underbrace{(0.1, 0.1, 0.3, 0.5)}_{A \wedge B} \vee \underbrace{(0.1, 0.2, 0.3, 0.4)}_{A} = (0.1, 0.1, 0.3, 0.5) \neq A$$

Faça $A = (0.1, 0.4, 0.3, 0.2)$ e $B = (0.1, 0.5, 0.3, 0.1)$ serem elementos de NL_2. Então:

$$A \vee B = (0.1, 0.5, 0.3, 0.1) \text{ e}$$
$$\underbrace{(0.1, 0.5, 0.3, 0.1)}_{A \vee B} \wedge \underbrace{(0.1, 0.4, 0.3, 0.2)}_{A} = (0.1, 0.5, 0.3, 0.1) \neq A$$
$$A \wedge B = (0.1, 0.5, 0.3, 0.1) \text{ e}$$
$$\underbrace{(0.1, 0.5, 0.3, 0.1)}_{A \wedge B} \vee \underbrace{(0.1, 0.4, 0.3, 0.2)}_{A} = (0.1, 0.5, 0.3, 0.1) \neq A.$$

□

Teorema 4.3.5. *PNL_1 não satisfaz a lei de absorção.*

Demonstração. Faça $A = (0.1, 0.1, 0.9)$ e $B = (0.1, 0.2, 0.9)$ serem elementos em PNL_1. Então:

$$A \vee B = (0.1, 0.2, 0.9) \text{ e}$$
$$\underbrace{(0.1, 0.2, 0.9)}_{A \vee B} \wedge \underbrace{(0.1, 0.1, 0.9)}_{A} = (0.1, 0.2, 0.9) \neq A$$
$$A \wedge B = (0.1, 0.2, 0.9) \text{ e}$$
$$\underbrace{(0.1, 0.2, 0.9)}_{A \wedge B} \vee \underbrace{(0.1, 0.1, 0.9)}_{A} = (0.1, 0.2, 0.9) \neq A.$$

□

Teorema 4.3.6. *Se A é uma elemento de NL_1, então $(1,0,0) \wedge A =_{NL_1} A \wedge (1,0,0) =_{NL_1} A$ e não existe outro elemento que é a identidade para $\wedge$.*

Demonstração. Faça $A = (t_1, i_1, f_1)$. Então $(1, 0, 0) \wedge (t_1, i_1, f_1) = (min\{1, t_1\}, 1 - (min\{1, t_1\} + max\{0, f_1\}), max\{0, f_1\}) = (t_1, i_1, f_1)$.

Já foi provado que $\wedge$ é comutativa, assim $(1, 0, 0)$ é a identidade comutativa para $\wedge$.

Suponha que existe outro elemento (t, i, f) que é uma identidade para $\wedge$. Pelas limitações nos valores, $t < 1.0$. Então é possível escolher um valor t_k tal que $t < t_k < 1$. Executando a operação $(t, i, f) \wedge (t_k, i_k, f_k)$, o resultado na primeira posição é t, visto que ele é o menor valor. A tripla então não pode ser igual a (t_k, i_k, f_k), violando a assunção que (t, i, f) é uma identidade para $\wedge$. □

Definição 4.3.2. *Um monóide é um semigrupo S com um elemento identidade.*

Teorema 4.3.7. *A álgebra $[NL1, \wedge]$ é um monóide.*

Demonstração. Visto que um monóide é um semigrupo com uma identidade, este é uma consequência direta do teorema anterior e do resultado anterior que $[NL1, \wedge]$ é um semigrupo. □

Teorema 4.3.8. *Faça A se um elemento de NL_1. Então*

$$(0,0,1) \vee A =_{NL_1} A \vee (0,0,1) =_{NL_1} A$$

e não há outro elemento tal que ele é uma identidade para $\vee$.

Demonstração. Faça $A = (t_1, i_1, f_1)$. Então

$$(0,0,1) \vee (t_1, i_1, f_1) =$$
$$(\max\{0,t_1\}, 1 - (\max\{0,t_1\} + \min\{1,f_1\}), \min\{1,f_1\}) = (t_1, i_1, f_1)$$

e o resto segue da comutatividade de $\vee$.

Assuma que exista outro elemento (t,i,f) que é uma identidade para $\vee$. Novamente, pelas limitações de valores, $t < 1.0$. Então é possível escolher um valor f_k tal que $f < f_k < 1.0$. Executando a operação, $(t,i,f) \vee (t_k, i_k, f_k)$ o resultado na última posição é f, visto que é o menor. A tripla não pode ser igual a (t_k, i_k, f_k), violando a assunção que (t,i,f) é um elemento identidade para $\vee$. □

Teorema 4.3.9. *A álgebra $[NL_1, \vee]$ é um monóide.*

Demonstração. Uma consequência direta da definição de monóide, do teorema e do resultado anterior resultam que $[NL1, \vee]$ é um semigrupo. □

Teorema 4.3.10. *Se A é um elemento arbitrário de NL_2, então não existe elemento (t,i,f) tal que $(t,i,f) \wedge A =_{NL_2} A \wedge (t,i,f) =_{NL_2} A$. Portanto, não existe elemento identidade para $\wedge$ em NL_2 e $[NL_2, \wedge]$ não é um monóide.*

Demonstração. Faça $A = (t_1, i_1, f_1)$. Então

$$(t,i,f) \wedge (t_1, i_1, f_1) =$$
$$(1 - \max\{i,i_1\} - \min\{1 - \max\{i,i_1\}, \max\{f,f_1\}\}, \max\{i,i_1\}, \min\{1 - \max\{i,i_1\}, \max\{f,f_1\}\})$$

O único modo do termo do meio ser i_1 é se $i = 0$ e o único modo que o terceiro termo pode ser f_1 é se $f = 0$. Isto força t ser 1.0. Contudo, em geral $\min\{1 - i_1, f_1\}$ não é f_1. □

Teorema 4.3.11. *Se A é um elemento arbitrário de NL_2, então não existe elemento (t,i,f) tal que $(t,i,f) \wedge A =_{NL_2} A \vee (t,i,f) =_{NL_2} A$. Portanto, não existe elemento identidade para $\vee$ em NL_2 e $[NL_2, \vee]$ não é um monóide.*

Demonstração. Similar ao do teorema 4.3.10. □

Teorema 4.3.12.
a) Não existe elemento em NL_1 que é uma identidade para $\wedge$.
b) Não existe elemento em NL_1 que é uma identidade para $\vee$.

Consequentemente, nem $[NL_1, \wedge]$ ou $[NL_1, \vee]$ são monóides.

Demonstração.
a) Faça $A = (t_1, i_1, f_1, u_1)$ ser um elemento de NL_1. Para que haja um elemento $I = (t,i,f,u)$ tal que $A \wedge I = A$, segue pode definição que $t \geq t_1$ e $i \geq i_1$ para todos os valores de t_1 e i_1. Os únicos valores que satisfazem isto são $t = 1.0$ e $i = 1.0$. Contudo, as restrições nos valores desabilitam a possibilidade que eles são simultaneamente iguais a um.

b) A prova é similar, exceto que f é usado em lugar de t. □

Teorema 4.3.13.
a) Não existe elemento em NL_2 que é uma identidade para $\wedge$.
b) Não existe elemento em NL_2 que é uma identidade para $\vee$.

Consequentemente, nem $NL_2, \wedge]$ ou $NL_2, \vee]$ são monóides.

Demonstração. Para $\wedge$, aplique o raciocínio do teorema 4.3.12 usando t e u e para $\vee$ aplique o raciocínio usando f e u. □

Teorema 4.3.14.

a) Se existe um valor máximo ($\max_t$) que é o componente verdade que pode se ter emPNL_1, então $(max_t, 0, 0)$ é uma identidade para $\wedge$.
b) Se existe um valor máximo ($\max_f$) que é o componente falso que se pode ter em PNL_1, então $(0, 0, max_f)$ é uma identidade para $\vee$.

Portanto, se existe um valor máximo para o componente verdade, $[PNL_1, \wedge]$ é um monóide e se existe um valor máximo para o componente falso, $[PNL_1, \vee]$ é um monóide.

Demonstração. Faça $A = (t_1, i_1, f_1)$ se um elemento de PNL_1. Então

$$(t_1, i_1, f_1) \wedge (\max_t, 0, 0) = (\min\{t_1, \max_t\}, \max\{i_1, 0\}, \max\{f_1, 0\}) = (t_1, i_1, f_1) \text{ e}$$

$$(t_1, i_1, f_1) \vee (0, 0, \max_f) = (\max\{t_1, 0\}, \max\{i_1, 0\}, \min\{f_1, \max_f\}) = (t_1, i_1, f_1).$$

Claramente, se o valor do componente verdade é ilimitado, não é possível encontrar um valor para a verdade onde o mínimo daquele valor e todos os outros são sempre o outro valor. Um situação similar se mantém para os valores do componente falso. □

Definição 4.3.3. *Faça* $\circ$ *ser uma operação binária fechada em um conjunto* S*. A operação é dita satisfazer a lei idempotente se* $x \circ x = x$ *para todo* $x \in S$*.*

Teorema 4.3.15. *Para todo* $(t, i, f) \in NL_1$*,* $(t, i, f) \wedge (t, i, f) = (t, i, f)$ *e* $(t, i, f) \vee (t, i, f) = (t, i, f)$*. Em outras palavras,* $\vee$ *e* $\wedge$ *são idempotentes em* NL_1*.*

Demonstração. Dado que $\min\{x, x\} = x$ e $\max\{x, x\} = x$, a conclusão segue das definições dos operadores. □

Teorema 4.3.16. *Para todo* $(t, i, f) \in NL_2$*,* $(t, i, f) \wedge (t, i, f) = (t, i, f)$ *e* $(t, i, f) \vee (t, i, f) = (t, i, f)$*. Em outras palavras,* $\wedge$ *e* $\vee$ *são idempotentes em* NL_2*.*

Demonstração. Por definição

$(t, i, f) \wedge (t, i, f) =$

$(1 - \max\{i, i\} - \min\{1 - \max\{i, i\}, \max\{f, f\}\}, \max\{i, i\}, \min\{1 - \max\{i, i\}, \max\{f, f\}\})$

O termo do meio é claramente i com a restrição que $t + f + i = 1.0$. Então, segue que $\max\{f, f\} = f, \max\{i, i\} = i$ e $1 - i$ pode nunca ser menor que f. Consequentemente, o terceiro termo é f, de modo que o primeiro termo deve ser t.

Por definição, $(t, i, f) \wedge (t, i, f) = (\min\{1 - \max\{i, i\}, \max\{t, t\}\}, \max\{i, i\}, 1 - \max\{i, i\} - \min\{1 - \max\}i, i\}, \max\{t, t\}\})$.

O termo do meio claramente é i com a restrição que $t + f + i = 1.0$. Então, segue que $\max\{t, t\} = t, \max\{i, i\} = i$ e $1 - i$ nunca pode ser menor do que t. Portanto, o primeiro termo é t, de modo que o terceiro termo deve ser f. □

Teorema 4.3.17.

a) $\wedge$ *e* $\vee$ *são idempotentes em* NL_1*.*

b) $\wedge$ *e* $\vee$ *são idempotentes em* NL_2*.*

c) $\wedge$ *e* $\vee$ *são idempotentes em* PNL_1*.*

Demonstração. Todas as provas são baseadas no princípio que $\max\{x,x\} = x$ e $\min\{x,x\} = x$ para todos os números x. □

Teorema 4.3.18. *Faça A, B e C serem elementos em NL_1. A lei distributiva é válida para $\wedge$ sobre $\vee$, em outras palavras*

$$A \wedge (B \vee C) =_{NL_1} (A \wedge B) \vee (A \wedge C).$$

Demosntração. Faça $A = (t_1, i_1, f_1)$, $B = (t_2, i_2, f_2)$ e $C = (t_3, i_3, f_3)$. Neste caso, a prova será feita examinando os valores dos componentes das triplas geradas pelos operadores.

Caso 1: Os valores dos primeiros elementos das triplas ordenadas.

No lado esquerdo, teríamos $\min\{t_1, \max\{t_2, t_3\}\}$ e no lado direito teríamos $\max\{\min\{t_1, t_2\}, \min\{t_1, t_3\}\}$.

Subcaso 1: t_1 é menor ou igual a t_2 e t_3. Então

$$\min\{t_1, \max\{t_2, t_3\}\} = t_1 = \max\{t_1, t_1\} = \max\{min\{t_1, t_2\}, \min\{t_1, t_3\}\}.$$

Subcaso 2: $t_2 \geq t_1 \geq t_3$. Então

$$\min\{t_1, \max\{t_2, t_3\}\} = \min\{t_1, t_2\} = t_1 = \max\{t_1, t_3\} = \max\{\min\{t_1, t_2\}, \min\{t_1, t_3\}\}.$$

Subcaso 3: $t_3 \geq t_1 \geq t_2$. Então

$$\min\{t_1, \max\{t_2, t_3\}\} = \min\{t_1, t_3\} = t_1 = \max\{t_2, t_1\} = \max\{\min\{t_1, t_2\}, \min\{t_1, t_3\}\}.$$

Subcaso 4: $t1 \geq t_2 \geq t_3$. Então

$$\min\{t_1, \max\{t_2, t_3\}\} = t_2 = \max\{t_2, t_3\} = \max\{\min\{t_1, t_2\}, \min\{t_1, t_3\}\}.$$

Subcaso 5: $t_1 \geq t_3 \geq t_2$. Então

$$\min\{t_1, \max\{t_2, t_3\}\} = t_3 = \max\{t_2, t_3\} = \max\{\min\{t_1, t_2\}, \min\{t_1, t_3\}\}.$$

Todos os casos foram considerados, assim o primeiro elemento da tripla é o mesmo em ambos os lados da expressão.

Caso 2: O valor do terceiro elemento na tripla ordenada.

Neste caso, o terceiro elemento no lado esquerdo é computador por $\max\{f_1, \min\{f_2, f_3\}\}$ e o terceiro elemento no lado direito é computado por $\min\{\max\{f_1, f_2\}, \max\{f_1, f_3\}\}$.

Subcaso 1: f_1 é maior ou igual a f_2 e f_3. Então

$$\max\{f_1, \min\{f_2, f_3\}\} = f_1 = \min\{f_1, f_1\} = \min\{\max\{f_1, f_2\}, \max\{f_1, f_3\}\}.$$

Subcaso 2: $f_2 \geq f_1 \geq f_3$. Então

$$\max\{f_1, \min\{f_2, f_3\}\} = f_1 = \min\{f_2, f_1\} = \min\{\max\{f_1, f_2\}, \max\{f_1, f_3\}\}.$$

Subcaso 3: $f_2 \geq f_3 \geq f_1$. Então

$$\max\{f_1, \min\{f_2, f_3\}\} = f_3 = \min\{f_2, f_3\} = \min\{max\{f_1, f_2\}, \max\{f_1, f_3\}\}.$$

Subcaso 4: $f_3 \geq f_1 \geq f_2$. Então

$$\max\{f_1, \min\{f_2, f_3\}\} = f_1 = \min\{f_1, f_3\} = \min\{\max\{f_1, f_2\}, \max\{f_1, f_3\}\}.$$

Subcaso 5: $f_3 \geq f2 \geq f1$. Então

$$\max\{f_1, \min\{f_2, f_3\}\} = f_2 = \min\{f_2, f_3\} = \min\{\max\{f_1, f_2\}, \max\{f_1, f_3\}\}.$$

Todos os casos foram considerados, assim o terceiros elementos das triplas ordenadas tem o mesmos valores.

Dado que o primeiro e terceiro valor das triplas ordenadas tem sempre o mesmo valor, todos devem ser iguais e a propriedade distributiva de $\wedge$ sobre $\vee$ é válida. □

Teorema 4.3.19. *Faça A, B e C serem elementos de NL_1. A lei distributiva para $\vee$ sobre $\wedge$ é válida, em outras palavras, $A \vee (B \wedge C) =_{NL_1} (A \vee B) \wedge (A \vee C)$.*

Demonstração. Faça $A = (t_1, i_1, f_1)$, $B = (t_2, i_2, f_2)$ and $C = (t_3, i_3, f_3)$. Neste caso, a prova será feita examinando os valores dos componentes das triplas geradas pelas operações.

Caso 1: O valor do primeiro elemento nas triplas ordenadas.

No lado esquerdo, poderíamos ter $\max\{t_1, \min\{t_2, t_3\}\}$ e no lado direito poderíamos ter $\min\{\max\{t_1, t_2\}, \max\{t_1, t_3\}\}$.

Subcaso 1: t_1 é maior ou igual a t_2 e t_3. Então

$$\max\{t_1, \min\{t_2, t_3\}\} = t_1 = \min\{t_1, t_1\} = \min\{max\{t_1, t_2\}, \max\{t_1, t_3\}\}.$$

Subcaso 2: $t_2 \geq t_1 \geq t_3$. Então

$$\max\{t_1, \min\{t_2, t_3\}\} = t_1 = \min\{t_2, t_1\} = \min\{\max\{t_1, t_2\}, \max\{t_1, t_3\}\}.$$

Subcaso 3: $t_2 \geq t_3 \geq t_1$. Então

$$\max\{t_1, \min\{t_2, t_3\}\} = t_3 = \min\{t_2, t_3\} = \min\{\max\{t_1, t_2\}, \max\{t_1, t_3\}\}.$$

Subcaso 4: $t_3 \geq t_1 \geq t_2$. Então

$$\max\{t_1, \min\{t_2, t_3\}\} = t_1 = \min\{t_1, t_3\} = \min\{\max\{t_1, t_2\}, \max\{t_1, t_3\}\}.$$

Subcaso 5: $t_3 \geq t_2 \geq t_1$. Então

$$\max\{t_1, \min\{t_2, t_3\}\} = t_2 = \min\{t_2, t_3\} = \min\{\max\{t_1, t_2\}, \max\{t_1, t_3\}\}.$$

Isto completa o caso de análise e, consequentemente, segue que os primeiros valores das triplas são os mesmo em todos os casos.

Caso 2: Os valores do terceiro elemento nas triplas ordenadas.

No lado esquerdo, teríamos $\min\{f_1, \max\{f_2, f_3\}\}$ e no lado direito teríamos $\max\{\min\{f_1, f_2\}, \min\{f_1, f_3\}\}$.

Subcaso 1: f_1 é menor ou igual a f_2 e f_3. Então

$$\min\{f_1, \max\{f_2, f_3\}\} = f_1 = \max\{f_1, f_1\} = \max\{\min\{f_1, f_2\}, \min\{f_1, f_3\}\}.$$

Subcaso 2: $f_2 \geq f_1 \geq f_3$. Então

$$\min\{f_1, \max\{f_2, f_3\}\} = f_1 = \max\{f_1, f_3\} = \max\{\min\{f_1, f_2\}, \min\{f_1, f_3\}\}.$$

Subcaso 3: $f_3 \geq f_1 \geq f_2$. Então

$$\min\{f_1, \max\{f_2, f_3\}\} = f_1 = \max\{f_2, f_1\} = \max\{\min\{f_1, f_2\}, \min\{f_1, f_3\}\}.$$

Subcaso 4: $f1 \geq f2 \geq f3$. Então

$$\min\{f_1, \max\{f_2, f_3\}\} = f_2 = \max\{f_2, f_3\} = \max\{\min\{f_1, f_2\}, \min\{f_1, f_3\}\}.$$

Subcaso 5: $f_1 \geq f_3 \geq f_2$. Então

$$\min\{f_1, \max\{f_2, f_3\}\} = f_3 = \max\{f_2, f_3\} = \max\{\min\{f_1, f_2\}, min\{f_1, f_3\}\}.$$

Isto completa os casos de análise e, portanto, segue que os terceiros valores das triplas são os mesmos em todos os casos. Consequentemente, as triplas são iguais em todos os casos e $\vee$ é distributivo sobre $\wedge$. □

Teorema 4.3.20. *Faça A, B e C serem elementos de NL_2. A lei distributiva para $\wedge$ sobre $\vee$ não é válida. Em outras palavras*

$$A \wedge (B \vee C) \neq_{NL_2} (A \wedge B) \vee (A \wedge C).$$

Demonstração. Faça $A = (0.04, 0.0{,}0.96)$, $B = (0.0{,}0.04{,}0.96)$ e $C = (0.04{,}0.0{,}0.96)$. Então

$$A \wedge B = (0.0, 0.04{,}0.96), A \wedge C = (0.04{,}0.0{,}0.96) \text{ e } B \vee C = (0.04, 0.04, 0.92)$$

O lado esquerdo é então

$$(0.04, 0.0, 0.96) \wedge (0.04, 0.04, 0.92) = (0.0, 0.04, 0.96)$$

e o lado direito é

$$(0.0, 0.04, 0.96) \vee (0.04, 0.0, 0.96) = (0.04, 0.04, 0.92).$$

□

Teorema 4.3.21. *Faça A, B e C serem elementos de NL_2. A lei distributiva para $\vee$ sobre $\wedge$ não é válida. Em outras palavras*

$$A \vee (B \wedge C) \neq_{NL_2} (A \vee B) \wedge (A \vee C).$$

Demonstração. Faça $A = (0.0{,}0.03{,}0.97)$, $B = (0.04, 0.0, 0.96)$ e $C = (0.04, 0.04, 0.92)$. Então

$$B \wedge C = (0.0, 0.04, 0.96), A \vee B = (0.04, 0.03, 0.92) \text{ e } A \vee C = (0.04, 0.04, 0.92)$$

O lado esquerdo então é $(0.0, 0.04, 0.96)$ e o lado direito é $(0.03, 0.04, 0.93)$. □

Teorema 4.3.22. *Faça A, B e C ser elementos de NL_1. Então*

i) $A \wedge (B \vee C) =_{NL_1} (A \wedge B) \vee (A \wedge C)$.
ii) $A \vee (B \wedge C) =_{NL_1} (A \vee B) \wedge (A \vee C)$.

Em outras palavras, as propriedades distributivas são válidas em NL_1.

Demonstração.

i) Faça $A = (t_1, i_1, f_1, u_1)$, $B = (t_2, i_2, f_2, u_2)$ e $C = (t_3, i_3, f_3, u_3)$. Visto que o valor de u é computado com base nos valores de t, i e f, se pudermos mostrar que os três primeiros elementos são verdadeiros, segue que os quartos elementos também devem ser verdadeiros. Aqui, $u = 1 - t - i - f$.

$$B \vee C = (t = \max\{t_2, t_3\}, i = \max\{i_2, i_3\}, f = \min\{f_2, f_3\}, u)$$
$$A \wedge (B \vee C) =$$
$$(t = \min\{t_1, \max\{t_2, t_3\}\}, i = \max\{i_1, \max\{i_2, i_3\}\}, f = \max\{f_1, \min\{f_2, f_3\}\}, u)$$
$$A \wedge B = (t = \min\{t_1, t_2\}, i = \max\{i_1, i_2\}, f = \max\{f_1, f_2\}, u)$$
$$A \wedge C = (t = \min\{t_1, t_3\}, i = \max\{i_1, i_3\}, f = \max\{f_1, f_3\}, u)$$
$$(A \wedge B) \vee (A \wedge C) = (\max\{\min\{t_1, t_2\}, \min\{t_1, t_3\}\},$$
$$\max\{\max\{i_1, i_2\}, \max\{i_1, i_3\}\}, \min\{\max\{f_1, f_2\}, \max\{\{f_1, f_3\}\}, u)$$

Claramente, $\max\{i_1, \max\{i_2, i_3\}\} = \max\{\max\{i_1, i_2\}, \max\{i_1, i_3\}\}$ para todos os valores, assim os temos indeterminado de ambas as 4-tuplas são iguais.

O restante da prova é baseada em análise de casos de todas as possibilidades.

Caso 1: t_1 é menor do que t_2 e t_3. Então $t_1 = \min\{t_1, \max\{t_2, t_3\}\}$ e $t_1 = \max\{\min\{t_1, t_2\}, \min\{t_1, t_3\}\}$.

Caso 2: t_1 é maior do que t_2 and t_3. Então $\min\{t_1, \max\{t_2, t_3\}\}$ é maior do que t_2 e t_3 e $\max\{\min\{t1, t_2\}, \min\{t_1, t_3\}\}$ também é maior do que t_2 e t_3.

Caso 3: $t_2 \geq t_1 \geq t_3$. Então $\min\{t_1, \max\{t_2, t_3\}\} = t_1$ e $\max\{\min\{t_1, t_2\}, \min\{t_1, t_3\}\} = t_1$.

Caso 4: $t_3 \geq t_1 \geq t_2$. Então $\min\{t_1, \max\{t_2, t_3\}\} = t_1$ e $\max\{\min\{t_1, t_2\}, \min\{t_1, t_3\}\} = t_1$.

Portanto, os valores dos componentes verdade são iguais para todas as possibilidades.

Caso 5: f_1 é maior do que f_2 e f_3. Então $\max\{f_1, \min\{f_2, f_3\}\} = f_1$ e $\min\{\max\{f_1, f_2\}, \max\{f_1, f_3\}\} = f_1$.

Caso 6: f_1 é menor do que f_2 e f_3. Então $\max\{f_1, \min\{f_2, f_3\}\}$ é o menor de f_2 e f_3 e $\min\{\max\{f_1, f_2\}, \max\{f_1, f_3\}\}$ também é o menor de f_2 e f_3.

Caso 7: $f_2 \geq f_1 \geq f3$. Então $\max\{f_1, \min\{f_2, f_3\}\} = f_1$ e $\min\{\max\{f_1, f_2\}, \max\{f_1, f_3\}\} = f_1$.

Caso 8: $f_3 \geq f_1 \geq f_2$. Então $\max\{f_1, \min\{f_2, f_3\}\} = f_1$ e $\min\{\max\{f_1, f_2\}, \max\{f_1, f_3\}\} = f_1$.

Portanto, os valores dos componentes falsos são iguais para todas as possibilidades.

Dado que os valore dos primeiros três elementos das 4-tuplas são iguais e o quarto é computado a partir dos três primeiros, segue que eles são iguais.

ii) A prova é similar a de (i) e assim é omitida. □

Teorema 4.3.23. *Faça A, B e C serem elementos de INL_2. Então*
i) $A \wedge (B \vee C) =_{INL_2} (A \wedge B) \vee (A \wedge C)$.
ii) $A \vee (B \wedge C) =_{INL_2} (A \vee B) \wedge (A \vee C)$.

Em outras palavras, as propriedades distributivas são válidas em INL_2.

Demonstração. Aplique análise de casos similar à feita para o Teorema 4.3.22. □

Teorema 4.3.24. *Faça A, B e C serem elementos de PNL_1. Então*

i) $A \wedge (B \vee C) =_{PNL_1} (A \wedge B) \vee (A \wedge C)$. *ii)* $A \vee (B \wedge C) =_{PNL_1} (A \vee B) \wedge (A \vee C)$.
Em outras palavras, as propriedades distributivas são válidas em PNL_1.

Demonstração. Aplique análise de casos similar à feita para o Teorema 4.3.22. □

Definição 4.3.4. *Para muitas das propriedades que examinamos, não existe distinção entre $\vee$ e $\wedge$. Se uma propriedade era verdadeira, então a propriedade com todas as instâncias de $\vee$ trocadas por $\wedge$ e todas as instâncias de $\wedge$ trocadas por $\vee$ também eram verdadeiras. Quando uma álgebra é definida com duas operações que podem ser intercambiadas deste modo, as duas expressões são ditas serem duais uma da outra.*

Exemplo: *As duas leis distributivas que foram verificadas nos teoremas anteriores são duais uma da outra.*

Definição 4.3.5. *Para qualquer conjunto S, se existe um elemento $e \in S$ e uma operação $\circ$ tal que $x \circ e = e \circ x = e$ para todo $x \in S$, então e édito se um elemento dominante para $\circ$ em S.*

Teorema 4.3.25. *O elemento $(0,0,1)$ é dominante para $\wedge$ em NL_1 e não existe outro elemento (q,r,s) em NL_1 tal que $(q,r,s) \wedge (t,i,f) = (q,r,s)$ para todo (t,i,f) em $NL)1$. Em outras palavras, $(0,0,1)$ é o único elemento dominante em NL_1.*

Demonstração. Claramente, $(0,0,1) \wedge (t,i,f) = (0,0,1)$, visto que $\max\{1,f\} = 1$. Os outros dois casos especiais de $(1,0,0)$ e $(0,1,0)$ podem ser eliminados dado que a conjunção com (t,i,f), onde f é diferente de zero, produz um resultado com uma terceira entrada diferente de zero.

Portanto, assuma que um elemento (q,r,s) existe que que no mínimo duas das entradas na tripla são diferentes de zero.

Caso 1: Suponha que $0 < q < 1$. Então, existe um valor $0 < q_k < q < 1$ onde $(q,r,s) \wedge (q_k,i,f)$ tem um primeiro valor na tripla de q_k. Pela escolha de q_k, temo uma contradição.

Caso 2: Suponha que $0 < s < 1$. Então, existe um valor $0 < s < s_k < 1$ onde $(q,r,s) \wedge (t,i,s_k)$ tem um terceiro valor na tripla de s_k. Pela escolha de s_k, temos uma contradição.

Dado que no mínimo uma das duas entradas examinadas nos casos 1 e 2 deve ser diferente de zero, nenhum elemento adicional pode existir. □

Teorema 4.3.26. *O elemento $(1,0,0)$ é dominante para $\vee$ em NL_1 e não existe outro elemento (q,r,s) em NL_1 tal que $(q,r,s) \vee (t,i,f) = (q,r,s)$ para todo (t,i,f) em NL_1. Em outras palavras, existe somente um elemento dominante para $\vee$ em NL_1.*

Demonstração. Similar a aquela do Teorema 4.3.25, assim, ela é omitida. □

Teorema 4.3.27.
i) $(0,1,0)$ *é o único elemento dominante em NL_2 para $\wedge$.*
ii) $(0,1,0)$ *é o único elemento dominante em NL_2 para $\vee$.*

Demonstração.

(a) Claramente, (0,1,0) $\wedge (t,i,f) = (0,1,0)$ para todo (t,i,f), dado que o termo do meio do resultado é o máximo dos termos do meio. Suponha que existe outro elemento dominante (t_1,i_1,f_1) para $\wedge$. Então $i_1 < 1.0$. Contudo, seria possível então encontrar outro valor i_2 tal que $i_1 < i_2 \leq 1.0$. Portanto,

$$(t_1,i_1,f_1) \wedge (t_2,i_2,f_2)$$

teria um termo do meio de i_2, contradizendo a assunção que (t_1,i_1,f_1) é um termo dominante.

(b) Similar ao do item (a), e desta feita, é omitida. □

Teorema 4.3.28.
i) O elemento $(0,0,1,0)$ *é um elemento dominante para* $\wedge$ *em* INL_1.
ii) O elemento $(1,0,0,0)$ *é um elemento dominante para* $\vee$ *em* INL_1.
iii) O elemento $(0,0,1,0)$ *é um elemento dominante para* $\wedge$ *em* INL_2.
iv) O elemento $(1,0,0,0)$ *é um elemento dominante para* $\vee$ *em* INL_2.

Demonstração.

a) Faça $A = (t,i,f,u)$ ser um elemento de INL_1 e considere $A \wedge (0,0,1,0)$. Então $\min\{t,0\} = 0, \min\{I,0\} = 0, \max\{f,1\} = 1$ e $n = 1\breve{}0\breve{}0\breve{}1 = 0$.

b) – d) Estas provas são similares a parte (a) e por isso são omitidas. □

Teorema 4.3.29. *Assuma que existem valores máximos para as componentes* t, i *e* f *das triplas em* PNL *e chame-as de* max_t, max_i *e* max_f, *respectivamente. Então:*

i) O elemento $(0, max_i, max_f)$ *é um elemento dominante para* $\wedge$ *em* PNL_1.
ii) O elemento $(max_t, max_i, 0)$ *é um elemento dominante para* $\vee$ *em* PNL_1.

Demonstração.

i) Faça $A = (t,i,f)$ ser um elemento de PNL_1 e considere $A \wedge (0, max_i, max_f)$. Por definição, $\min\{t,0\} = 0$, $\max\{0, max_i\} = max_i$ e $\max\{f, max_f\} = max_f$. Portanto, o valor da conjunção é $(0, max_i, max_f)$.

ii) Faça $A = (t,i,f)$ ser um elemento de PNL_1 e considere $A \vee (max_t, max_i, 0)$. Por definição, $\max\{t, max_t\} = max_t$, $\max\{0, max_i\} = max_i$ e $min\{f,0\} = 0$. Portanto, o valor da disjunção é $(max_t, max_i, 0)$. □

Definição 4.3.6. *Dado um conjunto S com operação $\circ$ e uma identidade E, se x é um elemento qualquer, então o elemento x^{-1} é o inverso de x se $x \circ x^{-1} = E$.*

Definição 4.3.7. *Um monóide $[S, \circ]$ com identidade E é um grupo se para todo elemento x tem um inverso x^{-1} tal que $x \circ x^{-1} = E$.*

Teorema 4.3.30.
(i) $[NL_1, \vee]$ *não é um grupo.*
(ii) $[NL_1, \wedge]$ *não é um grupo.*
(iii) $[NL_2, \vee]$ *não é um grupo.*
(iv) $[NL_2, \wedge]$ *não é um grupo.*

Demonstração.

(i) A única propriedade de um grupo que não foi verificada ainda é a presença de inversos. Faça $(t, i, f) \in NL_1$, onde t, i, f são todos diferentes de zero. O inverso com relação a $\vee$ seria um elemento (t_1, i_1, f_1) tal que $(t, i, f) \vee (t_1, i_1, f_1) = (0, 0, 1)$, dado que $(0, 0, 1)$ é a identidade para $\vee$. Contudo, pela definição $\max\{t, t_1\} = 0$ se, e somente se, $t = t_1 = 0$, contradizendo a escolha geral de t. Portanto, não existe tal inverso.

(ii) Faça $(t, i, f) \in NL_1$, onde t, i, f são diferentes de zero. O inverso com relação a $\wedge$ seria um elemento (t_1, i_1, f_1) tal que $(t, i, f) \wedge (t_1, i_I, f_I) = (1, 0, 0)$, dado que $(1, 0, 0)$ é a identidade para $\wedge$. Contudo, por definição, $\min\{t, t_1\} = 1$ se, e somente se, $t = t_1 = 1$, contradizendo a escolha geral de t. Consequentemente, não existe inverso.

As provas de (iii) e (iv) são similares e serão omitidas. □

Teorema 4.3.31.

i) $[INL_1, \wedge]$ *não é um grupo.*
ii) $[INL_2, \vee]$ *não é um grupo.*
iii) $[INL_2, \wedge]$ *não é um grupo.*
iv) $[INL_2, \vee]$ *não é um grupo.*

Se existem identidade para $\wedge$ *e* $\vee$ *em* PNL_1*:*

v) $[PNL_1, \wedge]$ *não é um grupo.*
vi) $[PNL_1, \vee]$ *não é um grupo.*

Demonstração. Já foi provado que não existe identidade para $\wedge$ ou $\vee$ em INL_1 e não existe identidade para $\wedge$ ou $\vee$ em INL_2. Dado que um conjunto deve ter identidade para ele ser um grupo, os resultados de (i) − (iv) seguem imediatamente.

v) A identidade para $\wedge$ em PNL_1 é $(max_t, 0, 0)$. Tome um elemento $A = (t_1, i_1, f_1)$ em PNL_1 onde i_1 é diferente de zero. Se este elemento tem um inverso $B = (t_2, i_2, f_2)$, então deve ser verdadeiro que $\max i + 1, i_2 = 0$. Isto é impossível, visto que i_1 é diferente de zero.

A prova de vi) é similar aquela dada para v) e será omitida. □

Definição 4.3.8. *A clássica definição de uma álgebra booelana é como segue.*

Uma álgebra booleana é um conjunto B no qual são definidas uma relação de equivalência "=" e duas operações "+" e "" tais que as seguintes propriedades são satisfeitas:*

(i) Para todo x e y em B, $a + b$ e $x * y$ também estão em B. (Fechamento)
(ii) Existem os elementos 0 e 1 em B tais que para todos x em B, $x + 0 = 0$ e $x * 1 = x$. (Identidade)
(iii) Para todo x e y em B, $x + y = y + x$ e $x * y = y * x$. (Comutatividade)
(iv) Para todo x, y e z em B, $x + (y * z) = (x + y) * (x + z)$ e $x * (y + z) = (x * y) + (x * z)$. (Distributividade)
(v) Para todo x em B, existe outro elemento x' também em B, tal que $x + x' = 1$ e $x * x' = 0$. (Leis de Complementariedade)
(vi) Existe no mínimo dois diferentes elementos em B.
(vii) Se $x = y$, então para todo z em B, $x + z = y + z$ e $x * z = y * z$. (Princípio de Substituição)

Para nossos propósitos, tomaremos NL com nosso conjunto e a igualdade para significar que

$$(t, i, f) = (t_I, i_1, f_1)$$

se, e somente se, $t = t_1$, $i = i_1$ e $f = f_1$.

A operação de $\wedge$ em NL_1 será a operação + e $\wedge$ a operação *. O elemento $(0, 0, 1)$ é a identidade para $\vee$ e $(1, 0, 0)$ a identidade para $\wedge$.

Teorema 4.3.32.

i) A álgebra $[NL1, \wedge, vee]$ não é uma álgebra booleana.
ii) A álgebra $[NL_2, \wedge, vee]$ não é uma álgebra booleana.
iii) A álgebra $[INL_1, \wedge, vee]$ não é uma álgebra booleana.
iv) A álgebra $[INL_2, \wedge, vee]$ não é uma álgebra booleana.
v) A álgebra $[PNL_1, \wedge, vee]$ não é uma álgebra booleana.

Demonstração.
i) Foi provado no Teorema 4.3.30 que existem elementos em NL_1 que não tem inversos. Consequentemente, a propriedade (v) de uma álgebra booleana não é válida. De fato, o resultado é até mais forte pelo fato de que nenhum elemento que não é uma identidade não tem um inverso.

As provas de ii) a v) são idênticas a da parte (i). □

Foi provado que a propriedade (v) de uma álgebra booleana não é válida para NL_2. Com exceção da propriedade (vii) de uma álgebra booelana, foi provado que NL_1 satisfaz todas as outras, exceto a distributiva.

A prova de (vii) para NL_1 e NL_2 é um tanto fácil, sendo o assunto do Teorema 4.3.33.

Teorema 4.3.33.
i) NL_1 satisfaz o princípio de substituição de uma álgebra booleana. Em outras palavras, se A, B e C são elementos de NL_1 e $B = C$, então

$$A \vee B =_{NL_1} A \vee C \text{ e } A \wedge B =_{NL_1} A \wedge C.$$

ii) NL_2 satisfaz o princípio de substituição de uma álgebra booleana. Em outras palavras, se A, B e C são elementos de NL_2 e $B = C$, então

$$A \vee B =_{NL_2} A \vee C \text{ e } A \wedge B =_{NL_2} A \wedge C.$$

Demonstração.

i) Faça $A = (t_1, i_1, f_1)$, $B = (t_2, i_2, f_2)$ e $C = (t_3, i_3, f_3)$.

$A \wedge B =$
$(t_1, i_1, f_1) \vee (t_2, i_2, f_2) = (\max\{t_1, t_2\}, 1 - \max\{t_1, t_2\} - \min\{f_1, f_2\}, \min\{f_1, f_2\})$
$t_1, i_1, f_1) \vee (t_3, i_3, f_3) = (\max\{t_1, t_3\}, 1 - \max\{t_1, t_3\} - \min\{f_1, f_3\}, \min\{f_1, f_3\})$

Se $t_2 = t_3$, $i_2 = i_3$ e $f_2 = f_3$, então
$(\max\{t_1, t_2\}, 1 - \max\{t_1, t_2\} - \min\{f_1, f_2\}, \min\{f_1, f_2\}) =$
$(\max\{t_1, t_3\}, 1 - \max\{t_1, t_3\} - \min\{f_1, f_3\}, \min\{f_1, f_3\})$

A prova para $\wedge$ é similar, e assim sendo, é omitida.

ii) Similar a daquela de (i), e assim, é omitida. □

Teorema 4.3.34.

i) INL_1 *satisfaz o princípio de substituição de uma álgebra booleana. Em outras palavras, se* A*,* B *e* C *são elementos de* INL_1 *e* $B = C$*, então*

$$A \vee B =_{INL_1} A \vee C \text{ e } A \wedge B =_{INL_1} A \wedge C.$$

ii) INL_2 *satisfaz o princípio de substituição de uma álgebra booleana. Em outras palavras, se* A*,* B *e* C *são elementos de* INL_2 *e* $B = C$*, então*

$$A \vee B =_{INL_2} A \vee C \text{ e } A \wedge B =_{INL_2} A \wedge C.$$

iii) PNL_1 *satisfaz o princípio de substituição de uma álgebra booleana. Em outras palavras, se* A*,* B *e* C *são elementos de* PNL_1 *e* $B = C$*, então*

$$A \vee B =_{PNL_1} A \vee C \text{ e } A \wedge B =_{PNL_1} A \wedge C.$$

Demonstração. A prova de cada resultado é similar a do Teorema 4.3.33. □

Consequentemente, com exceção das propriedade da inversa, NL_1 satisfaz todas as propriedades de uma álgebra booleana. Visto que ela está bem próxima de uma álgebra booleana, NL_1 satisfaz várias outras propriedades comumente associadas com uma álgebra booelana. NL_2 está longe de uma álgebra booelana, dado que ela não satisfaz as propriedades distributivas. INL_1, INL_2 e PNL_1 também são muito similares a uma álgebra booleana.

Teorema 4.3.35. NL_1 *satisfaz as leis de DeMorgan de distribução da negação. Em outras palavras, se* A *e* B *são elementos de* NL_1*, então*

(i) $\neg(A \vee B) =_{NL_1} \neg A \wedge \neg B$
(ii) $\neg(A \wedge B) =_{NL_1} \neg A \vee \neg B$.

Demonstração. Faça $A = (t_1, i_1, f_1)$ e $B = (t_2, i_2, f_2)$.

(i) $\neg(A \vee B) = \neg(\max\{t_1, t_2\}, 1 - \max\{t_1, t_2\} - \min\{f_1, f_2\}, \min\{f_1, f_2\}) = (\min\{f_1, f_2\}, 1 - \max\{t_1, t_2\} - \min\{f_1, f_2\}, \max\{t_1, t_2\}) = (f_1, i_1, t_1) \wedge (f_2, i_2, t_2) = \neg(t_1, i_1, f_1) \wedge \neg(t_2, i_2, f_2) = \neg A \wedge \neg B$.

(ii) $\neg(A \wedge B) = \neg(\min\{t_1, t_2\}, 1 - \min\{t_1, t_2\} - \max\{f_1, f_2\}, \max\{f_1, f_2\}) = (\max\{f_1, f_2\}, 1 - \min\{t_1, t_2\} - \max\{f_1, f_2\}, \min\{t_1, t_2\}) = (f_1, i_1, t_1) \vee (f_2, i_2, t_2) = \neg(t_1, i_1, f_1) \vee \neg(t_2, i_2, f_2) = \neg A \vee \neg B$. □

Teorema 4.3.36. *NL_2 satisfaz as leis DeMorgan da distribuição da negação. Em outras palavras, se A e B são elementos de NL_1, então*

i) $\neg(A \vee B) =_{NL_2} \neg A \wedge \neg B$.
ii) $\neg(A \wedge B) =_{NL_2} \neg A \wedge \neg B$.

Demonstração. Faça $A = (t_1, i_1, f_1)$ e $B = (t_2, i_2, f_2)$.

i) $A \vee B =$
$(\min\{1 - \max\{i_1, i_2\}, \max\{t_1, t_2\}\},$
$\max\{i_1, i_2\}, 1 - \max\{i_1, i_2\} - \min\{1 - \max\{i_1, i_2\}, \max\{t_1, t_2\}\})$

$\neg(A \vee B) = (1 - \max\{i_1, i_2\} - \min\{1 - \max\{i_1, i_2\}, \max\{t_1, t_2\}\},$
$\max\{i_1, i_2\}, \min\{1 - \max\{i_1, i_2\}, \max\{t_1, t_2\}\})$

$\neg A = (f_1, i_1, t_1) \quad \neg B = (f_2, i_2, t_2)$

$\neg A \wedge \neg B = (1 - \max\{i_1, i_2\} - \min\{1 - \max\{i_1, i_2\}, \max\{t_1, t_2\}\}, \max\{i_1, i_2\},$
$\min\{1 - \max\{i_1, i_2\}, \max\{t_1, t_2\}\})$

ii) A prova de (ii) é similar, e assim sendo, é omitida. □

Teorema 4.3.37. *INL_1 satisfaz as leis DeMorgan da distribuição da negação. Em outras palavras, se A e B são elementos de INL_1, então*

i) $\neg(A \vee B) =_{INL_1} \neg A \wedge \neg B$.
ii) $\neg(A \wedge B) =_{NL_2} \neg A \wedge \neg B$.

Demonstração.
Faça $A = (t_1, i_1, f_1, u_1)$ e $B = (t_2, i_2, f_2, u_2)$.

i) Então, $A \vee B = (\max\{t_1, t_2\}, \min\{i_1, i_2\}, \min\{f_1, f_2\},$ computado $u)$ e
$\neg(A \vee B) = (\min\{f_1, f_2\}, \min\{i_1, i_2\}, \max\{t_1, t_2\},$ computado $u)$.

$\neg A = (f_1, i_1, t_1, u_1)$, $\neg B = (f_2, i_2, t_2, u_2)$ e
$\neg A \wedge \neg B = (\min\{f_1, f_2\}, \min\{i_1, i_2\}, \max\{t_1, t_2\},$ computado $u)$.

ii) A prova é similar, sendo portanto, omitida. □

Teorema 4.3.38. *INL_2 satisfaz as leis DeMorgan de distribuição da negação. Em outras palavras, se A e B são elementos de INL_2, então*

i) $\neg(A \vee B) =_{INL_2} \neg A \wedge \neg B$.
ii) $\neg(A \wedge B) =_{INL_2} \neg A \vee \neg B$.

Demonstração. Quase idêntica a do Teorema 4.3.37. □

Teorema 4.3.39. *PNL_1 satisfaz as leis DeMorgan de distribuição da negação. Em outras palavras, se A e B são elementos de PNL_1, então*

i) $\neg(A \vee B) =_{PNL_1} \neg A \wedge \neg B$.
ii) $\neg(A \wedge B) =_{PNL_1} \neg A \vee \neg B$.

Demonstração. Similar ao do Teorema 4.3.37. □

Definição 4.3.9. *Uma expressão em NL_1 é dita ser bem formada se ela satisfaz as seguintes propriedades:*

(i) (t, i, f) *onde* $t + i + f = 1.0$ *é bem formada.*
(ii) As constantes $T = (1, 0, 0)$, $I = (0, 1, 0)$ *e* $F = (0, 0, 1)$ *são bem formadas.*
(iii) Se A é bem formada em NL_1, então assim o é $\neg A$.
(iv) Se A é bem formada, então assim o é (A).
(v) Se A e B são bem formadas em NL_1, então assim o são $A \wedge B$ e $A \vee B$.
(vi) Nada que não possa ser formado usando as regras (i) – (v) em um número finito de vezes é bem formado.

Uma definição similar pode ser usada para NL_2, INL_1, INL_2 e PNL_1.

Lema 4.3.1. *Se A e B são elementos de NL_1, então:*

i) Se $A \vee B = (1, 0, 0)$, *então ou A ou B é* $(1, 0, 0)$.
ii) Se $A \wedge B = (1, 0, 0)$, *então ou A ou B é* $(1, 0, 0)$.
iii) Se $A \vee B = (0, 0, 1)$, *então ou A ou B é* $(0, 0, 1)$.
iv) Se $A \wedge B = (0, 0, 1)$, *então ou A ou B é* $(0, 0, 1)$.
v) Se $\neg A = (1, 0, 0)$, *então* $A = (0, 0, 1)$. *vi) Se* $\neg A = (0, 0, 1)$, *então* $A = (1, 0, 0)$.

Demonstração. Os resultados de i) a iv) são consequência das definições de $\vee$ e $\wedge$, v) e vi) são consequência da definição de negação. □

Teorema 4.3.40.

i) Se uma expressão bem formada A em NL_1 é uma tautologia absoluta , então A contém o elemento $(1, 0, 0)$.
ii) Se uma expressão bem formada A em NL_1 é uma contradição absoluta , então A contém o elemento $(0, 0, 1)$.

Demonstração. Como sempre, a expressão A é construída usando os três conectivos, $\{\neg, \wedge, \vee\}$. A prova de ambas as partes será combinada em uma, dado que o resultado é similar.

Dado que A é bem formada em NL_1, ela é construída usando um número finito de símbolos em $\{\neg, \wedge, \vee\}$. Consequentemente, a prova será por indução sobre o número de operações usadas para construir a expressão.

Passo básico: Claramente se a expressão A é uma tripla, então A é uma absoluta tautologia se, e somente se, $A = (1, 0, 0)$ e uma contradição absoluta se, e somente se, $A = (0, 0, 1)$.

Passo indutivo: Assuma que para qualquer expressão A construída usando k ou menos passos, se A é uma tautologia absoluta , então $(1, 0, 0)$ aparece em A. Também assuma que se A é construída da mesma maneira e é uma contradição absoluta , então $(0, 0, 1)$ aparece em A.

Faça A e B serem expressões quaisquer satisfazendo as assunções indutivas. Existe quatro regras que podem ser aplicadas.

Caso 1: A regra iii) da definição de uma fbf é aplicada a A.

O único valor para A que pode produzir $(1, 0, 0)$ é $(0, 0, 1)$ e o único valor que produz $(0, 0, 1)$ é $(1, 0, 0)$. Consequentemente, o único modo que $\neg A$ pode ser uma tautologia absoluta é se A é uma contradição absoluta e o único modo que $\neg A$ pode ser uma contradição absoluta se A é uma absoluta tautologia. Aplicando a hipótese de indução, segue que se $\neg A$ é uma tautologia ou uma contradição absoluta, isto decorre porque A contém $(1, 0, 0)$ ou $(0, 0, 1)$.

Caso 2: A regra iv) da definição de uma fbf é aplicada a A.

Dado que encerrando as expressões em parêntesis não muda em nada o valor da expressão, segue que (A) é uma tautologia ou contradição absoluta porque A contém $(1, 0, 0)$ ou $(0, 0, 1)$.

Caso 3: A regra v) da definição de uma fbf é aplicada para a forma $A \wedge B$.

A partir do lema, temos que se a expressão é uma absoluta tautologia, então ou A ou B é. Podemos então aplicar a hipótese indutiva para concluir que se $A \wedge B$ é uma absoluta tautologia, então ela contém o elemento $(1, 0, 0)$. Raciocínio similar irá produzir o resultado comparável para $A \wedge B$ sendo uma contradição absoluta.

Caso 4: A regra v) da definição de uma fbf é aplicada para a forma $A \vee B$.

A partir do lema, temos que se a expressão é uma tautologia absoluta , então ou A ou B é. Podemos então aplicar a hipótese indutiva para concluir que se $A \vee B$ é uma tautologia absoluta , então ela contém o elemento $(1, 0, 0)$. Raciocínio similar irá produzir o resultado comparável para $A \vee B$ sendo uma contradição absoluta.

Consequentemente, pelo princípio da indução matemática, temos alcançado a conclusão desejada. □

O teorema para NL_2 é similar, mas tem uma restrição adicional.

Lema 4.3.2. *Se A e B são elementos de NL_2, então:*

i) Se $A \vee B = (1, 0, 0)$, então A ou B é $(1, 0, 0)$ e os termos do meio nas triplas de A e B são zero.
ii) Se $A \wedge B = (1, 0, 0)$, então A ou B é $(1, 0, 0)$ e os termos do meio nas triplas de A e B são zero.
iii) Se $A \vee B = (0, 0, 1)$, então A ou B é $(0, 0, 1)$ e os temos do meio nas triplas de A e B são zero.
iv) Se $A \wedge B = (0, 0, 1)$, então A ou B é $(0, 0, 1)$ e os termos do meio na triplas de A e B são zero.
v) Se $\neg A = (1, 0, 0)$, então $A = (0, 0, 1)$.
vi) Se $\neg A = (0, 0, 1)$, então $A = (1, 0, 0)$.

Demonstração. (i) Se A ou B tem um termo do meio que não é zero, então pela definição de $\vee$, o termo do meio da expressão deve ser diferente de zero. Com os termos do meio iguais a zero, o primeiro termo é o máximo dos dois primeiros termos e o resulto segue.

(iii) – (vi) As provas são similares a de (i) e são omitidas. □

Teorema 4.3.41.

i) Se uma expressão bem formada A em NL_2 é uma tautologia absoluta, então A contém o elemento $(1, 0, 0)$ e todos os termos do meio das triplas em A são zero.
ii) Se uma expressão bem formada A em NL_2 é uma contradição absoluta, então ela contém o elemento $(0, 0, 1)$ e todos os termos do meio em A são zero.

Demonstração. Similar ao do Teorema 4.3.40. □

Teorema 4.3.42.

i) Se A é um elemento de INL_1 e $A = (1,0,0,0)$, então A contém o elemento $(1,0,0,0)$.
ii) Se A é um elemento de INL_1 e $A = (0,0,1,0)$, então A contém o elemento $(0,0,1,0)$.
iii) Se A é um elemento de INL_2 e $A = (1,0,0,0)$, então A contém o elemento $(1,0,0,0)$.
iv) Se A é um elemento de INL_2 e $A = (0,0,1,0)$, então A contém o elemento $(0,0,1,0)$.

Demonstração. Similar ao do Teorema 4.3.40. □

4.4 Definido outros Conectivos em Lógica Neutrosófica

Como foi mencionado nas seções sobre lógica clássica, trivalente e fuzzy, é possível definir conectivos adicionais como abreviações para expressões construídas usando $\{\neg, \wedge, \vee\}$. Contudo, é difícil definir igualdade lógica quando se usa números reais, sendo a diferença devida a erros de arredondamento que podem levar a conclusão de desigualdade. Um problema similar ocorre quando se usa desigualdade lógica. Portanto, se estes conectivos serão definidos de maneira útil, deve haver um termo de erro que pode ser usado para definir uma região, "próxima o suficiente para ser considerada igual". Por estas razões, não definiremos os operadores equivalentes de equivalência e ou-exclusivo em NL_1 e NL_2.

Definição 4.4.1. *Se A e B são elementos de NL_1, então a implicação $A \rightarrow_{NL_1} B$ é uma abreviação para $\neg A \vee B$.*

Exemplo:

Se $A = (.12, .24, .64)$ e $B = (.45, .06, .49)$, então:

$$A \rightarrow_{NL_1} B = (.64, .24, .12) \vee (.45, .06, .49) = (.64, .24, .12).$$

Definição 4.4.2. *Se A e B são elementos de NL_2, então a implicação $A \rightarrow_{NL_2} B$ é uma abreviação para $\neg A \vee B$.*

Exemplo:

Se $A = (.12, .24, .64)$ e $B = (.45, .06, .49)$, então:

$$A \rightarrow_{NL_2} B = (.64, .24, .12) \vee (.45, .06, .49) = (.64, .24, .12)$$

Definição 4.4.3. *Se A e B são elementos de INL_1, então a implicação $A \rightarrow_{INL_1} B$ é uma abreviação para $\neg A \vee B$.*

Exemple:

Se $A = (.12, .24, .34, .30)$ e $B = (.45, .06, .29, .20)$, então:

$$A \rightarrow_{INL_1} B = (.34, .24, .12, .30) \vee (.45, .06, .29, .20) = (.45, .06, .12, .37)$$

Definição 4.4.4. *Se A e B são elementos de INL_2, então a implicação $A \rightarrow_{INL_2} B$ é uma abreviação para $\neg A \vee B$.*

Exemplo:

Se $A = (.12, .24, .34, .30)$ e $B = (.45, .06, .29, .20)$, então:

$$A \rightarrow_{INL_2} B = (.34, .24, .12, .30) \vee (.45, .06, .29, .20) = (.45, .23, .12, .20)$$

Definição 4.4.5. *Se A e B são elementos de PNL_1, então a implicação $A \rightarrow_{PNL_1} B$ é uma abreviação para $\neg A \vee B$.*

Exemplo:

Se $A = (.12, .24, .74)$ e $B = (.42, .06, .59)$, então:

$$A \rightarrow_{PNL_1} B = (.74, .24, .12) \vee (.45, .06, .59) = (.74, .24, .12)$$

Definição 4.4.6. *Se A e B são elementos de NL_1, então a negação conjunta $\downarrow_{NL_1}$ é uma abreviação para $\neg(A \vee B)$.*

Exemplo:

Se A = (.12, .24, .64) and B = (.45, .06, .49), então:

$$A \downarrow_{NL_1} B = \neg(A \vee B) = \neg(.45, .06, .49) = (.49, .06, .45).$$

Definição 4.4.7. *Se A e B são elementos de NL_2, então a negação conjunta $\downarrow_{NL_2}$ é uma abreviação para $\neg(A \vee B)$.*

Exemplo:

Se A = (.12, .24, .64) and B = (.45, .06, .49), então:

$$A \downarrow_{NL_2} B = \neg(A \vee B) = \neg(.45, .24, .31) = (.31, .24, .45)$$

Definição 4.4.8. *Se A e B são elementos of INL_1, então a negação conjunta $\downarrow_{INL_1}$ é uma abreviação para $\neg(A \vee B)$.*

Exemplo:

Se $A = (.12, .24, .34, .30)$ e $B = (.45, .06, .29, .20)$, então:

$$A \downarrow_{INL_1} B = \neg(A \vee B) = \neg(.45, .06, .29, .30) = (.29, .06, .45, .30)$$

Definição 4.4.9. *Se A e B são elementos of INL_2, então a negação conjunta $\downarrow_{INL_2}$ é uma abreviação para $\neg(A \vee B)$.*

Exemplo:

Se $A = (.12, .24, .34, .30)$ e $B = (.45, .06, .29, .20)$, então:

$$A \downarrow_{INL_2} B = \neg(A \vee B) = \neg(.45, .06, .29, .20) = (.29, .06, .45, .20).$$

Definição 4.4.10. *Se A e B are elementos de PNL_1, então a negação conjunta $\downarrow_{PNL_1}$ é uma abreviação para $\neg(A \vee B)$.*

Exemplo:

Se $A = (.12, .24, .74)$ e $B = (.42, .06, .59)$, então:

$$A \downarrow_{PNL_1} B = \neg(A \vee B) = \neg(.42, .24, .59) = (.59, .24, .42).$$

Definição 4.4.11. *Se A e B são elementos de NL_1, então a negação alternativa $|_{NL_1}$ é uma abreviação para $\neg(A \wedge B)$.*

Exemplo:

Se $A = (.12, .24, .64)$ e $B = (.45, .06, .49)$, então:

$$A \,|_{NL_1}\, B = \neg(A \wedge B) = \neg(.12, .24, .64) = (.64, .24, .12)$$

Definição 4.4.12. *Se A e B são elementos de NL_2, então a negação alternativa $|_{NL_2}$ é uma abreviação para $\neg(A \wedge B)$.*

Exemplo:

Se $A = (.12, .24, .64)$ e $B = (.45, .06, .49)$, então:

$$A \,|_{NL_2}\, B = \neg(A \wedge B) = \neg(.12, .24, .64) = (.64, .24, .12).$$

Definição 4.4.13. *Se A e B são elementos de INL_1, então a negação alternativa $|_{NL_1}$ é uma abreviação para $\neg(A \wedge B)$.*

Exemplo:

Se $A = (.12, .24, .34, .30)$ e $B = (.45, .06, .29, .20)$, então:

$$A \,|_{INL_1}\, B = \neg(A \wedge B) = \neg(.12, .06, .34, .48) = (.34, .06, .12, .48).$$

Definição 4.4.14. *Se A e B são elementos de INL_2, então a negação alternativa $|_{INL_2}$ é uma abreviação para $\neg(A \wedge B)$.*

Exemplo:

Se $A = (.12, .24, .34, .30)$ e $B = (.45, .06, .29, .20)$, então

$$A \ |_{INL_2} \ B = \neg(A \wedge B) = \neg(.12, .34, .34, .20) = (.34, .34, .12, .20).$$

Definição 4.4.15. *Se A e B são elementos de PNL_2, então a negação alternativa $|_{PNL_1}$ é uma abreviação para $\neg(A \wedge B)$.*

Exemplo:

Se $A = (.12, .24, .74)$ e $B = (.42, .06, .59)$, então:

$$A \ |_{PNL_1} \ B = \neg(A \wedge B) = \neg(.12, .24, .74) = (.74, .24, .12).$$

Teorema 4.4.1. *Os conectivos de negação conjunta e alternativa para NL_1, NL_2, INL_1, INL_2 e PNL_1 são comutativos. Contudo, a implicação não é.*

Demonstração. Dado que a negação conjunta $(A \downarrow B)$ é uma abreviação para $\neg(A \vee B)$ e a negação alternativa $(A \ | \ B)$ é uma abreviação para $\neg(A \wedge B)$ e ambos os operadores $\vee$ e $\wedge$ são comutativos, o resultado segue. Contudo, a implicação $(A \to B)$ é uma abreviação para $(\neg A \vee B)$, que não é o mesmo que $(\neg B \vee A)$.

Se estes três adicionais conectivos serão usados em NL_1, NL_2, INL_1, INL_2 e PNL_1, então a regra v) da definição de uma expressão bem formada poderia ser modificada para

v) Se A e B são bem formadas em N_L, então assim o são $A \wedge B$, $A \vee B$, $A \to B$, $A \downarrow B$ e $A \ | \ B$. □

4.5 Implementando os Conectivos Neutrosóficos em Programas de Computador

A seguinte classe em Java é uma implementação dos elementos de NL_1.

```
/*
This class is an implementation of the elements of the Neutrosophic Logic
(NL). It was developed by Charles Ashbacher 12/31/2000.
*/

public class NLElement {
    private float truthvalue;
    private float indeterminatevalue;
    private float falsevalue;
```

```java
/*
This function is called when a new instance is created. The values are checked
for conformance to the rule that the values must sum to 1.0. Since floating
point addition is not precise, the test allows for some inaccuracy.
*/
public NLElement(float tvalue, float ivalue, float fvalue)
{
    float test;
    test = Math.abs(1.0f - (tvalue+ivalue+fvalue));

    if(test < 0.00001f) {
        truthvalue = tvalue;
        indeterminatevalue = ivalue;
        falsevalue = fvalue;
    }
    else {
        System.out.println("The inputs do not satisfy the criteria that they
            sum to 1.0");
        System.out.println("The object has been set to indeterminate");
        truthvalue = falsevalue = 0.0f;
        indeterminatevalue = 1.0f;
    }
}

/*
This function is used to update the contents of an NL element. The values are
checked for conformance to the rule that the values must sum to 1.0. Since
floating point addition is not precise, the test allows for some inaccuracy.
*/
public void updatevalues(float tvalue, float ivalue, float fvalue)
{
    float test;
    test = Math.abs(1.0f - (tvalue + ivalue + fvalue));

    if(test < 0.00001f) {
        truthvalue = tvalue;
        indeterminatevalue = ivalue;
        falsevalue = fvalue;
    }
    else {
        System.out.println("The inputs do not satisfy the criteria that they
            sum to 1.0");
        System.out.println("The object has been set to indeterminate");
        truthvalue = falsevalue = 0.0f;
        indeterminatevalue = 1.0f;
    }
}
```

```
    public float gettrue()
    {
        return truthvalue;
    }

    public float getindeterminate()
    {
        return indeterminatevalue;
    }

    public float getfalse()
    {
        return falsevalue;
    }

    public void printvalues()
    {
        System.out.println("The truth value is "+truthvalue);
        System.out.println("The indeterminate value is "+indeterminatevalue);
        System.out.println("The false value is "+falsevalue);
    }
}
```

O programa Java seguinte contém funções que implementam os conectivos $\wedge$, $\vee$, $\neg$, $\to$, $\downarrow$ e $|$ para a lógica NL_1.

```
/*
This program implements the connectives defined in the NL1 logic. It was
written by Charles Ashbacher 12/31/2000. The names of the function should be
self-explanatory as to which connective it implements.
*/

public class UsingNLElement
{
    public static NLElement NLand(NLElement nle1, NLElement nle2)
    {
        float nle1true, nle1false;
        float nle2true, nle2false;
        float assigntrue, assignfalse;
        nle1true = nle1.gettrue();
        nle1false = nle1.getfalse();
        nle2true = nle2.gettrue();
        nle2false = nle2.getfalse();

        if(nle1true >= nle2true)
        {
            assigntrue = nle2true;
        }
```

```
        if(nle1false >= nle2false)
        {
            assignfalse = nle1false;
        }
        else
        {
            assignfalse = nle2false;
        }

        NLElement tempNL = new NLElement(assigntrue, 1.0f - (assigntrue +
            assignfalse), assignfalse);
        return tempNL;
    }

    public static NLElement NLor(NLElement nle1, NLElement nle2)
    {
        float nle1true, nle1false;
        float nle2true, nle2false;
        float assigntrue, assignfalse;
        nle1true = nle1.gettrue();
        nle1false = nle1.getfalse();
        nle2true = nle2.gettrue();
        nle2false = nle2.getfalse();

        if(nle1true >= nle2true)
        {
            assigntrue = nle1true;
        }
        else
        {
            assigntrue = nle2true;
        }

        if(nle1false >= nle2false)
        {
            assignfalse = nle2false;
        }
        else
        {
            assignfalse = nle1false;
        }

        NLElement tempNL = new NLElement(assigntrue, 1.0f - (assigntrue +
            assignfalse), assignfalse);

        return tempNL;
    }
```

```
public static NLElement NLnegation(NLElement nle1)
{
    float assigntrue = nle1.getfalse();
    float assignfalse = nle1.gettrue();
    NLElement tempNL = new NLElement(assigntrue, 1.0f - (assigntrue +
        assignfalse), assignfalse);
    return tempNL;
}

public static NLElement NLimplication(NLElement nle1, NLElement nle2)
{
    NLElement nle3 = NLnegation(nle1);
    NLElement nle4 = NLor(nle3, nle2);
    return(nle4);
}

public static NLElement NLjointdenial(NLElement nle1, NLElement nle2)
{
    NLElement nle3 = NLor(nle1, nle2);
    NLElement nle4 = NLnegation(nle3);
    return (nle4);
}

public static NLElement NLaltdenial(NLElement nle1,NLElement nle2)
{
    NLElement nle3 = NLand(nle1,nle2);
    NLElement nle4=NLnegation(nle3);
    return (nle4);
}

public static void main(String args[])
{
    NLElement nle1 = new NLElement(0.2f, 0.3f, 0.5f);
    NLElement nle2 = new NLElement(0.1f, 0.3f, 0.6f);
    NLElement nle3;
    nle3 = NLand(nle1, nle2);
    System.out.println("The original values are ");
    nle1.printvalues();
    nle2.printvalues();
    System.out.println("The result of NL and is ");
    nle3.printvalues();
    nle3=NLor(nle1, nle2);
    System.out.println("The result of NL or is ");
    nle3.printvalues();
    nle3 = NLnegation(nle1);
    System.out.println("The result of NL negation is ");
    nle3.printvalues();
    nle3 = NLimplication(nle1, nle2);
    System.out.println("The result of NL implication is ");
    nle3.printvalues();
```

```
        nle3 = NLjointdenial(nle1, nle2);
        System.out.println("The result of NL joint denial is ");
        nle3.printvalues();
        nle3 = NLaltdenial(nle1, nle2);
        System.out.println("The result of NL alternative denial is ");
        nle3.printvalues();
    }
}
```

A saída neste caso é

```
The original values are
The truth value is .2
The indeterminate value is .3
The false value is .5
The truth value is .1
The indeterminate value is .3
The false value is .6
The result of NL and is
The truth value is .1
The indeterminate value is .3
The false value is .6
The result of NL or is
The truth value is .2
The indeterminate value is .3
The false value is .5
The result of NL negation is
The truth value is .5
The indeterminate value is .3
The false value is .2
The result of NL implication is
The truth value is .5
The indeterminate value is .3
The false value is .2
The result of NL joint denial is
The truth value is .5
The indeterminate value is .3
The false value is .2
The result of NL alternative denial is
The truth value is .6
The indeterminate value is .3
The false value is .1
```

O seguinte código Java implementa os elementos da Lógica Neutrosófica Intuicionista.

```
/* Written by Charles Ashbacher June, 2002. */
public class INLElement
{
    private float truthvalue;
    private float indeterminatevalue;
    private float falsevalue;
    private float unknownvalue;

    public INLElement(float tvalue, float ivalue, float fvalue, float uvalue)
    {
        float test;
        test = Math.abs(1.0f - (tvalue + ivalue + fvalue + uvalue));

        if(test < 0.00001f)
    {
        truthvalue = tvalue;
        indeterminatevalue = ivalue;
        falsevalue = fvalue;
        unknownvalue = uvalue;
    }
    else
    {
        System.out.println("The inputs do not satisfy the criteria that they
            sum to 1.0");
        System.out.println("The object has been set to unknown");
        truthvalue = falsevalue = indeterminatevalue = 0.0f;
        unknownvalue = 1.0f;
    }
}

public void updatevalues(float tvalue, float ivalue, float fvalue, float
    uvalue)
{
    float test;
    test = Math.abs(1.0f - (tvalue + ivalue + fvalue + uvalue));

    if(test < 0.00001f) {
        truthvalue = tvalue;
        indeterminatevalue = ivalue;
        falsevalue = fvalue;
        unknownvalue = uvalue;
    }
    else {
        System.out.println("The inputs do not satisfy the criteria that they
            sum to 1.0");
        System.out.println("The object has been set to unknown");
        truthvalue = falsevalue = indeterminatevalue = 0.0f;
        unknownvalue = 1.0f;
    }
}
```

```
    public float gettrue()
    {
        return truthvalue;
    }

    public float getindeterminate()
    {
        return indeterminatevalue;
    }

    public float getfalse()
    {
        return falsevalue;
    }

    public float getunknown()
    {
        return unknownvalue;
    }

    public void printvalues()
    {
        System.out.println("The truth value is " + truthvalue);
        System.out.println("The indeterminate value is " + indeterminatevalue)
            ;
        System.out.println("The false value is " + falsevalue);
        System.out.println("The unknown value is " + unknownvalue);
        System.out.println(" ");
    }
}
```

O seguinte programa Java contém funções que implementam os conectivos $\wedge$, $\vee$, $\neg$, $\rightarrow$, $\downarrow$ e $|$ para a lógica INL_1.

```
/* Written by Charles Ashbacher, June, 2002. */
public class UsingINLElement
{
    public static INLElement INL1and(INLElement nle1, INLElement nle2)
    {
        float nle1true, nle1false;
        float nle2true, nle2false;
        float assigntrue, assignfalse, assignindeterminate, assignunknown;
        float nle1indeterminate, nle2indeterminate;

        // Get the values of the components needed in the computation
        nle1true = nle1.gettrue();
        nle1false = nle1.getfalse();
        nle1indeterminate = nle1.getindeterminate();
        nle2true = nle2.gettrue();
```

```
        nle2false = nle2.getfalse();
        nle2indeterminate = nle2.getindeterminate();

        // Compute the and
        assigntrue = getMin(nle1true, nle2true);
        assignfalse = getMax(nle1false, nle2false);
        assignindeterminate = getMin(nle1indeterminate, nle2indeterminate);
        assignunknown = 1.0f - assigntrue - assignfalse - assignindeterminate;
        INLElement tempNL = new INLElement(assigntrue, assignindeterminate,
            assignfalse, assignunknown);
        return tempNL;
    }

    public static INLElement INL2and(INLElement nle1, INLElement nle2)
    {
        float nle1true, nle1false;
        float nle2true, nle2false;
        float assigntrue, assignfalse, assignindeterminate, assignunknown;
        float nle1unknown,nle2unknown;

        // Get the values of the components needed in the computation
        nle1true = nle1.gettrue();
        nle1false = nle1.getfalse();
        nle1unknown = nle1.getunknown();
        nle2true = nle2.gettrue();
        nle2false = nle2.getfalse();
        nle2unknown = nle2.getunknown();

        // Compute the and
        assigntrue = getMin(nle1true, nle2true);
        assignfalse = getMax(nle1false, nle2false);
        assignunknown = getMin(nle1unknown, nle2unknown);
        assignindeterminate = 1.0f - assigntrue - assignfalse - assignunknown;
        INLElement tempNL = new INLElement(assigntrue, assignindeterminate,
            assignfalse, assignunknown);
        return tempNL;
    }

    public static INLElement INL1or(INLElement nle1, INLElement nle2)
    {
        float nle1true, nle1false;
        float nle2true, nle2false;
        float nle1indeterminate, nle2indeterminate;
        float assigntrue, assignfalse, assignindeterminate, assignunknown;

        // Get the values to be used in the computation
        nle1true = nle1.gettrue();
        nle1false = nle1.getfalse();
        nle1indeterminate = nle1.getindeterminate();
        nle2true = nle2.gettrue();
```

```
        nle2false = nle2.getfalse();
        nle2indeterminate = nle2.getindeterminate();

        // Compute the or
        assigntrue = getMax(nle1true, nle2true);
        assignfalse = getMin(nle1false, nle2false);
        assignindeterminate = getMin(nle1indeterminate, nle2indeterminate);
        assignunknown = 1.0f - assigntrue - assignfalse - assignindeterminate;
        INLElement tempNL = new INLElement(assigntrue, assignindeterminate,
            assignfalse, assignunknown);
        return tempNL;
    }

    public static INLElement INL2or(INLElement nle1, INLElement nle2)
    {
        float nle1true, nle1false;
        float nle2true, nle2false;
        float nle1unknown, nle2unknown;
        float assigntrue, assignfalse, assignindeterminate, assignunknown;

        // Get the values to be used in the computation
        nle1true = nle1.gettrue();
        nle1false = nle1.getfalse();
        nle1unknown = nle1.getunknown();
        nle2true = nle2.gettrue();
        nle2false = nle2.getfalse();
        nle2unknown = nle2.getunknown();

        // Compute the or
        assigntrue = getMax(nle1true, nle2true);
        assignfalse = getMin(nle1false, nle2false);
        assignunknown = getMin(nle1unknown, nle2unknown);
        assignindeterminate = 1.0f - assigntrue - assignfalse - assignunknown;
        INLElement tempNL = new INLElement(assigntrue, assignindeterminate,
            assignfalse, assignunknown);
        return tempNL;
    }

    // The definition of the negation is the same for INL1 and INL2,
    // so there is only one negation operation.
    public static INLElement INLnegation(INLElement nle1) {
        float assigntrue = nle1.getfalse();
        float assignfalse = nle1.gettrue();
        float assignindeterminate = nle1.getindeterminate();
        float assignunknown = nle1.getunknown();
        INLElement tempNL = new INLElement(assigntrue, assignindeterminate,
            assignfalse, assignunknown);
        return tempNL;
    }
```

```
public static INLElement INL1implication(INLElement nle1, INLElement nle2)
{
    INLElement nle3 = INLnegation(nle1);
    INLElement nle4 = INL1or(nle3, nle2);
    return(nle4);
}

public static INLElement INL2implication(INLElement nle1, INLElement nle2)
{
    INLElement nle3 = INLnegation(nle1);
    INLElement nle4 = INL2or(nle3,nle2);
    return(nle4);
}

public static INLElement INL1jointdenial(INLElement nle1, INLElement nle2)
{
    INLElement nle3 = INL1or(nle1, nle2);
    INLElement nle4 = INLnegation(nle3);
    return (nle4);
}

public static INLElement INL2jointdenial(INLElement nle1, INLElement nle2)
{
    INLElement nle3 = INL2or(nle1, nle2);
    INLElement nle4 = INLnegation(nle3);
    return (nle4);
}

public static INLElement INL1altdenial(INLElement nle1, INLElement nle2)
{
    INLElement nle3 = INL1and(nle1, nle2);
    INLElement nle4 = INLnegation(nle3);
    return (nle4);
}

public static INLElement INL2altdenial(INLElement nle1, INLElement nle2)
{
    INLElement nle3 = INL2and(nle1, nle2);
    INLElement nle4 = INLnegation(nle3);
    return (nle4);
}

public static float getMin(float x1, float x2)
{
    float theMin;

    if( x1 >= x2)
    {
        theMin = x2;
    }
```

```
        else
        {
            theMin = x1;
        }
        return theMin;
    }

    public static float getMax(float x1,float x2)
    {
        float theMax;

        if(x1 >= x2)
        {
            theMax = x1;
        }
        else
        {
            theMax = x2;
        }
        return theMax;
    }

    public static void main(String args[])
    {
        INLElement nle1 = new INLElement(0.3f, 0.1f, 0.5f, 0.1F);
        INLElement nle2 = new INLElement(0.1f,0.3f,0.2f,0.4f);
        INLElement nle3;
        nle3 = INL1and(nle1, nle2);
        System.out.println("The original values are ");
        nle1.printvalues();
        nle2.printvalues();
        System.out.println("The result of INL and is ");
        nle3.printvalues();
        nle3 = INL1or(nle1,nle2);
        System.out.println("The result of INL or is ");
        nle3.printvalues();
        nle3 = INLnegation(nle1);
        System.out.println("The result of NL negation is ");
        nle3.printvalues();
        nle3 = INL1implication(nle1, nle2);
        System.out.println("The result of NL implication is ");
        nle3.printvalues();
        nle3 = INL1jointdenial(nle1, nle2);
        System.out.println("The result of NL joint denial is ");
        nle3.printvalues();
        nle3 = INL1altdenial(nle1, nle2);
        System.out.println("The result of NL alternative denial is ");
        nle3.printvalues();
    }
}
```

Note que existe operações separadas para os conectivos em INL_1 e INL_2. A saída quando este programa é executado, onde os únicos conectivos usados são aqueles para INL_1, é como segue.

```
The original values are
The truth value is 0.3
The indeterminate value is 0.1
The false value is 0.5
The unknown value is 0.1

The truth value is 0.1
The indeterminate value is 0.3
The false value is 0.2
The unknown value is 0.4

The result of NL and is
The truth value is 0.1
The indeterminate value is 0.1
The false value is 0.5
The unknown value is 0.29999998

The result of NL or is
The truth value is 0.3
The indeterminate value is 0.1
The false value is 0.2
The unknown value is 0.4

The result of NL negation is
The truth value is 0.5
The indeterminate value is 0.1
The false value is 0.3
The unknown value is 0.1

The result of NL implication is
The truth value is 0.5
The indeterminate value is 0.1
The false value is 0.2
The unknown value is 0.20000002

The result of NL joint denial is
The truth value is 0.2
The indeterminate value is 0.1
The false value is 0.3
```

```
The unknown value is 0.4

The result of NL alternative denial is
The truth value is 0.5
The indeterminate value is 0.1
The false value is 0.1
The unknown value is 0.29999998
```

Se a função `main` é modificada como segue, onde os conectivos para INL_2 são usados:

```java
public static void main(String args[])
{
    INLElement nle1 = new INLElement(0.3f, 0.1f, 0.5f, 0.1F);
    INLElement nle2 = new INLElement(0.1f, 0.3f, 0.2f, 0.4f);
    INLElement nle3;
    nle3 = INL2and(nle1, nle2);
    System.out.println("The original values are ");
    nle1.printvalues();
    nle2.printvalues();
    System.out.println("The result of NL and is ");
    nle3.printvalues();
    nle3 = INL2or(nle1, nle2);
    System.out.println("The result of NL or is ");
    nle3.printvalues();
    nle3 = INLnegation(nle1);
    System.out.println("The result of NL negation is ");
    nle3.printvalues();
    nle3 = INL2implication(nle1, nle2);
    System.out.println("The result of NL implication is ");
    nle3.printvalues();
    nle3 = INL2jointdenial(nle1, nle2);
    System.out.println("The result of NL joint denial is ");
    nle3.printvalues();
    nle3 = INL2altdenial(nle1, nle2);
    System.out.println("The result of NL alternative denial is ");
    nle3.printvalues();
}
```

A saída é como segue:

```
The original values are
The truth value is 0.3
The indeterminate value is 0.1
The false value is 0.5
The unknown value is 0.1

The truth value is 0.1
The indeterminate value is 0.3
The false value is 0.2
```

```
The unknown value is 0.4

The result of NL and is
The truth value is 0.1
The indeterminate value is 0.29999998
The false value is 0.5
The unknown value is 0.1

The result of NL or is
The truth value is 0.3
The indeterminate value is 0.4
The false value is 0.2
The unknown value is 0.1

The result of NL negation is
The truth value is 0.5
The indeterminate value is 0.1
The false value is 0.3
The unknown value is 0.1

The result of NL implication is
The truth value is 0.5
The indeterminate value is 0.20000002
The false value is 0.2
The unknown value is 0.1

The result of NL joint denial is
The truth value is 0.2
The indeterminate value is 0.4
The false value is 0.3
The unknown value is 0.1

The result of NL alternative denial is
The truth value is 0.5
The indeterminate value is 0.29999998
The false value is 0.1
The unknown value is 0.1
```

O código seguinte implementa os elementos da Lógica Neutrosófica Paraconsistente.

```
/* Written by Charles Ashbacher June, 2002. */
public class PNLElement {
    private float truthvalue;
```

```
private float indeterminatevalue;
    private float falsevalue;

    public PNLElement(float tvalue, float ivalue, float fvalue)
    {
        float test = tvalue + ivalue + fvalue;

        if(test >= 1.0)
        {
            truthvalue = tvalue;
            indeterminatevalue = ivalue;
            falsevalue = fvalue;
        }
        else
        {
            System.out.println("The inputs do not satisfy the criteria that
                the sum be greater than or equal to 1.0");
            System.out.println("The object has been modified to have a high
                indeterminate value");
            truthvalue = tvalue;
            falsevalue = fvalue;
            indeterminatevalue = 1.0f;
        }
    }

    public void updatevalues(float tvalue, float ivalue, float fvalue)
    {
        float test = tvalue + ivalue + fvalue;

        if(test >= 1.0)
        {
            truthvalue = tvalue;
            indeterminatevalue = ivalue;
            falsevalue = fvalue;
        }
        else
        {
            System.out.println("The inputs do not satisfy the criteria that
                the sum be greater than or equal to 1.0");
            System.out.println("The object has been set to have a high
                indeterminate value");
            truthvalue = tvalue;
            falsevalue = fvalue;
            indeterminatevalue = 1.0f;
        }
    }

    public float gettrue() {
        return truthvalue;
    }
```

```java
    public float getindeterminate()
    {
        return indeterminatevalue;
    }

    public float getfalse()
    {
        return falsevalue;
    }

    public void printvalues()
    {
        System.out.println("The truth value is "+truthvalue);
        System.out.println("The indeterminate value is "+indeterminatevalue);
        System.out.println("The false value is "+falsevalue);
        System.out.println(" ");
    }
}
```

O seguinte programa em Java contém funções que implementam os conectivos $\wedge$, $\vee$, $\neg$, $\rightarrow$, $\downarrow$ e $|$ da lógica PNL.

```java
/* Written by Charles Ashbacher, June, 2002. */
public class UsingPNLElement
{
    public static PNLElement PNLand(PNLElement nle1, PNLElement nle2)
    {
        float nle1true, nle1false;
        float nle2true, nle2false;
        float assigntrue, assignfalse, assignindeterminate;
        float nle1indeterminate, nle2indeterminate;

        // Get the values of the components needed in the computation
        nle1true = nle1.gettrue();
        nle1false = nle1.getfalse();
        nle1indeterminate = nle1.getindeterminate();
        nle2true = nle2.gettrue();
        nle2false = nle2.getfalse();
        nle2indeterminate = nle2.getindeterminate();

        // Compute the and
        assigntrue = getMin(nle1true ,nle2true);
        assignfalse = getMax(nle1false, nle2false);
        assignindeterminate = getMax(nle1indeterminate, nle2indeterminate);
        PNLElement tempNL = new PNLElement(assigntrue, assignindeterminate,
            assignfalse);
        return tempNL;
    }
```

```
public static PNLElement PNLor(PNLElement nle1, PNLElement nle2)
{
    float nle1true, nle1false;
    float nle2true, nle2false;
    float nle1indeterminate, nle2indeterminate;
    float assigntrue, assignfalse, assignindeterminate;

    // Get the values to be used in the computation
    nle1true = nle1.gettrue();
    nle1false = nle1.getfalse();
    nle1indeterminate = nle1.getindeterminate();
    nle2true = nle2.gettrue();
    nle2false = nle2.getfalse();
    nle2indeterminate = nle2.getindeterminate();

    // Compute the or
    assigntrue=getMax(nle1true,nle2true);
    assignfalse=getMax(nle1false,nle2false);
    assignindeterminate=getMin(nle1indeterminate,nle2indeterminate);
    PNLElement tempNL = new PNLElement(assigntrue, assignindeterminate,
        assignfalse);
    return tempNL;
}

public static PNLElement PNLnegation(PNLElement nle1)
{
    float assigntrue = nle1.getfalse();
    float assignfalse = nle1.gettrue();
    float assignindeterminate = nle1.getindeterminate();
    PNLElement tempNL = new PNLElement(assigntrue, assignindeterminate,
        assignfalse);
    return tempNL;
}

public static PNLElement PNLimplication(PNLElement nle1, PNLElement nle2)
{
    PNLElement nle3 = PNLnegation(nle1);
    PNLElement nle4 = PNLor(nle3, nle2);
    return(nle4);
}

public static PNLElement PNLjointdenial(PNLElement nle1, PNLElement nle2)
{
    PNLElement nle3 = PNLor(nle1, nle2);
    PNLElement nle4 = PNLnegation(nle3);
    return (nle4);
}

public static PNLElement PNLaltdenial(PNLElement nle1, PNLElement nle2) {
    PNLElement nle3 = PNLand(nle1, nle2);
```

```
        PNLElement nle4 = PNLnegation(nle3);
        return (nle4);
    }

    public static float getMin(float x1,float x2)
    {
        float theMin;

        if(x1 >= x2)
        {
            theMin = x2;
        }
        else
        {
            theMin = x1;
        }
        return theMin;
    }

    public static void main(String args[])
    {
        PNLElement nle1 = new PNLElement(0.6f, 0.1f, 0.5f);
        PNLElement nle2 = new PNLElement(0.1f, 0.5f, 0.7f);
        PNLElement nle3;
        nle3 = PNLand(nle1, nle2);
        System.out.println("The original values are ");
        nle1.printvalues();
        nle2.printvalues();
        System.out.println("The result of NL and is ");
        nle3.printvalues();
        nle3 = PNLor(nle1, nle2);
        System.out.println("The result of NL or is ");
        nle3.printvalues();
        nle3 = PNLnegation(nle1);
        System.out.println("The result of NL negation is ");
        nle3.printvalues();
        nle3 = PNLimplication(nle1, nle2);
        System.out.println("The result of NL implication is ");
        nle3.printvalues();
        nle3 = PNLjointdenial(nle1, nle2);
        System.out.println("The result of NL joint denial is ");
        nle3.printvalues();
        nle3 = PNLaltdenial(nle1, nle2);
        System.out.println("The result of NL alternative denial is ");
        nle3.printvalues();
    }
}
```

A saída quando este programa é executado é como segue:

```
The original values are
```

```
The truth value is 0.6
The indeterminate value is 0.1
The false value is 0.5

The truth value is 0.1
The indeterminate value is 0.5
The false value is 0.7

The result of NL and is
The truth value is 0.1
The indeterminate value is 0.5
The false value is 0.7

The result of NL or is
The truth value is 0.6
The indeterminate value is 0.1
The false value is 0.7

The result of NL negation is
The truth value is 0.5
The indeterminate value is 0.1
The false value is 0.6

The result of NL implication is
The truth value is 0.5
The indeterminate value is 0.1
The false value is 0.7

The result of NL joint denial is
The truth value is 0.7
The indeterminate value is 0.1
The false value is 0.6

The result of NL alternative denial is
The truth value is 0.7
The indeterminate value is 0.5
The false value is 0.1
```

4.6 Ordenação dos Elementos da Lógica Neutrosófica

Nosso próximo passo na análise dos elementos de NL é impor uma relação de ordenação R neles. Ao considerar a definição de tal relação para NL_1, o valor verdade é considerado mais significante do que todos os outros, com o valor indeterminado mínimo. Com isto como consideração principal, o seguinte é a definição de uma relação de ordem em NL_1.

Definição 4.6.1. *Faça* (t_1, i_1, f_1) *e* (t_2, i_2, f_2) *serem elementos de* NL_1. *Então* $(t_1, i_1, f_1) <_{NL_1} (t_2, i_2, f_2)$ *se uma das seguintes condições é verdadeira:*

(i) $t_1 < t_2$
(ii) $t_1 = t_2$ *e* $f_1 < f_2$

A expressão $(t_1, i_1, f_1) \leq_{NL_1} (t_2, i_2, f_2)$ *é usada se* $t_1 = t_2$, $i_1 = i_2$ *e* $f_1 = f_2$ *pi* $(t_1, i_1, f_1) >_{NL_1} (t_2, i_2, f_2)$.

Em outras palavras, uma elementos em NL_1 *é maior do que outro se seu valor verdade é maior ou se seu valor falso é maior se os valores verdade são iguais.*

Esta definição é consistente com a ideia de que um elemento é maior do que outro se ele é conhecido com grande precisão do que o outro. Em geral, o raciocínio é executado aplicando o que é conhecido ser verdadeiro, de modo é que o primeiro ponto de ênfase. No caso onde os valores verdade são iguais, o interesse está em qual é mais preciso, que seria a maior soma para t e f.

Para NL_2, o valor indeterminado é considerado o mais significante, com o valor falso sendo o menos significante.

Definição 4.6.2. *Faça* (t_1, i_1, f_1) *e* (t_2, i_2, f_2) *serem elementos de* NL_2. *Então* $(t_1, i_1, f_1) <_{NL_2} (t_2, i_2, f_2)$ *se uma das seguintes condições é verdadeira:*

i) $i_1 < i_2$
ii) $i_1 = i_2$ *e* $t_1 < t_2$

A expressão $(t_1, i_1, f_1) \leq_{NL_2} (t_2, i_2, f_2)$ *é usada se* $t_1 = t_2$, $i_1 = i_2$ *e* $f_1 = f_2$ *ou* $(t_1, i_1, f_1) <_{NL_2} (t_2, i_2, f_2)$.

Em outras palavras, um elemento em NL_2 *é maior do que outros e seu valor indeterminado é maior ou se seu valor verdade é maior se os valores indeterminados são iguais.*

Definição 4.6.3. *Faça* (t_1, i_1, f_1, u_1) *and* (t_2, i_2, f_2, u_2) *serem elementos de* INL_1. *Então* $(t_1, i_1, f_1, u_1) <_{INL_1} (t_2, i_2, f_2, u_2)$ *se uma das seguintes condições é verdadeira:*

i) $t_1 < t_2$.
ii) $t_1 = t_2$ *e* $f_1 < f_2$.
iii) $t_1 = t_2$, $f_1 = f_2$ *e* $i_1 < i_2$.
A expressão $(t_1, i_1, f_1, u_1) \leq_{INL_1} (t_2, i_2, f_2, u_2)$ *é usada se* $t_1 = t_2$, $i_1 = i_2$, $f_1 = f_2$ *e* $u_1 = u_2$ *ou* $(t_1, i_1, f_1, u_1) <_{INL_1} (t_2, i_2, f_2, u_2)$.

Definição 4.6.4. *Faça* (t_1, i_1, f_1, u_1) *and* (t_2, i_2, f_2, u_2) *serem elementos de* INL_2. *Então* $(t_1, i_1, f_1, u_1) <_{INL_1} (t_2, i_2, f_2, u_2)$ *se uma das seguintes condições é verdadeira:*

i) $t_1 < t_2$.
ii) $t_1 = t_2$ *e* $f_1 < f_2$.
iii) $t_1 = t_2$, $f_1 = f_2$ *e* $u_1 < u_2$.

A expressão $(t_1, i_1, f_1, u_1) \leq_{INL_2} (t_2, i_2, f_2, u_2)$ *é usada se* $t_1 = t_2$, $i_1 = i_2$, $f_1 = f_2$ *e* $u_1 = u_2$ *ou* $(t_1, i_1, f_1, u_1) <_{INL_2} (t_2, i_2, f_2, u_2)$.

Definição 4.6.5. *Faça* (t_1, i_1, f_1) *e* (t_2, i_2, f_2) *serem elementos de* PNL_1. *Então* $(t_1, i_1, f_1) <_{PNL_1} (t_2, i_2, f_2)$ *se uma das seguintes condições é verdadeira:*

i) $t_1 < t_2$.
ii) $t_1 = t_2$ *e* $f_1 < f_2$.
iii) $t_1 = t_2$, $f_1 = f_2$ *e* $i_1 < i_2$.

A expressão $(t_1, i_1, f_1) \leq_{PNL_1} (t_2, i_2, f_2)$ *é suada se* $t_1 = t_2$, $i_1 = i_2$ *e* $f_1 = f_2$ *ou* $(t_1, i_1, f_1, u_1) < (t_2, i_2, f_2, u_2)$.

Teorema 4.6.1.

a) A relação de ordem (<) como definida em NL_1 *é*
(i) não reflexiva
(ii) assimétrica
(iii) transitiva
(iv) conectada

b) A relação de ordem (<) como definida em NL_2 *é*
(v) não reflexiva
(vi) assimétrica
(vii) transitiva
(viii) conectada

c) A relação de ordem (<) como definida em INL_1 *é*
(ix) não reflexiva
(x) assimétrica
(xi) transitiva
(xii) conectada

d) A relação de ordem (<) como definida em INL_2 *é*
(xiii) não reflexiva
(xiv) assimétrica
(xv) transitiva

(xvi) conectada

e) A relação de ordem (<) como definida em PNL_1 é
(xvii) não reflexiva
(xviii) assimétrica
(xix) transitiva
(xx) conectada

Demonstração.

(i) Se $(NL_1, <)$ não fosse não reflexiva, então haveria um elemento (t, i, f) tal que

$$(t, i, f) < (t, i, f)$$

Claramente, não é possível para $t < t$ ou $f < f$, assim isto é impossível.

ii) Assuma que existem elementos (t_1, i_1, f_1) e (t_2, i_2, f_2) tal que $(t_1, i_1, f_1) < (t_2, i_2, f_2)$ e $(t_1, i_1, f_1) < (t_2, i_2, f_2)$.

Caso 1: $t_1 \neq t_2$

Por definição, isto poderia significar que $t_1 < t_2$ $t_1 > t_2$, que é impossível.

Caso 2: $t_1 = t_2$

Por definição, isto significaria que $f_1 < f_2$ e $f_1 > f_2$ que é impossível.

Portanto, < em NL_1 é assimétrica.

iii) Faça $(t_1, i_1, f_1) < (t_2, i_2, f_2)$ e $(t_2, i_2, f_2) < (t_3, i_3, f_3)$. Existem vários casos a considerar.

Caso 1: $t_1 = t_2 = t_3$

Consequentemente, as desigualdades implicam que $f_1 < f_2$ e $f_2 < f_3$, o que pela transitividade de < sobre números reais produz $f_1 < f_3$. Portanto, pela definição $(t_1, i_1, f_1) < (t_3, i_3, f_3)$.

Caso 2: $t_1 = t_2$ e $t_2 < t_3$

Então, $t_1 < t_3$ e $(t_1, i_1, f_1) < (t_3, i_3, f_3)$ por definição.

Caso 3: $t_1 < t_2$ e $t_2 = t_3$

Então, $t_1 < t_3$ e $(t_1, i_1, f_1) < (t_3, i_3, f_3)$ por definição.

Caso 4: $t_1 < t_2$ e $t_2 < t_3$

Pela transitividade de $<$ sobre números reais, segue que $t_1 < t_3$ e $(t_1, i_1, f_1) < (t_3, i_3, f_3)$ por definição.

iv) Faça (t_1, i_1, f_1) e (t_2, i_2, f_2) serem elementos de NL_1. Se eles não são iguais, então existem dois casos.

Caso 1: $t_1 = t_2$ e $f1 \neq f_2$

Pela propriedade de tricotomia dos números reais, então segue que $f_1 > f_2$ ou $f_1 < f_2$. Se $f_1 > f_2$, então $(t_1, i_1, f_1) > (t_2, i_2, f_2)$ e se $f_1 < f_2$ então $(t_1, i_1, f_1) < (t_2, i_2, f_2)$.

Caso 2: $t_1 \neq t_2$

Pela propriedade de tricotomia novamente, isto significaria que $t_1 > t_2$ ou $t_1 < t_2$. Se $t_1 > t_2$, então $(t_1, i_1, f_1) > (t_2, i_2, f_2)$ e se $t_1 < t_2$, então $(t_1, i_1, f_1) < (t_2, i_2, f_2)$.

As provas de (b) até (e) são similares e são omitidas. □

Teorema 4.6.2. *Se A é um elemento de NL_1, então $A \leq_{NL_1} (1, 0, 0)$ e $A \geq_{NL_1} (0, 1, 0)$. Isto significa, é claro, que os elementos de NL_1 tem um limite superior e inferior.*

Faça $A = (t_1, i_1, f_1)$. Se $t_1 = 1$, então $A = (1, 0, 0)$ e se $t_1 < 1$, então $A < (1, 0, 0)$ por definição. Se $i_1 = 1.0$, então $A = (0, 1, 0)$ e se $i_1 \neq 1.0$, então um ou ambos, t_1 ou f_1, é maior do que zero;. Em outro caso, isto faria $A > (0, 1, 0)$.

Teorema 4.6.3. *Se A é um elemento de NL_2, então $A \leq_{NL_2} (0, 1, 0)$ e $A \geq_{NL_2} (0, 0, 1)$. Isto significa, é claro, que os elementos de NL_2 tem um limite superior e inferior.*

Demonstração. Faça $A = (t_1, i_1, f_1)$. Se $i_1 = 1$, então $A = (0, 1, 0)$ e se $i_1 < 1$, então $A < (0, 1, 0)$ por definição. Se $f_1 = 1.0$, então $A = (0, 0, 1)$ e se $f_1 \neq 1.0$, então um ou ambos, i_1 ou t_1, é maior do que zero. Em outro caso, isto faria $A > (0, 1, 0)$. □

Teorema 4.6.4.

i) Se A é um elemento de INL_1, então $A \leq_{INL_1} (1,0,0,0)$ e $A \geq_{INL_1} (0,1,0,0)$. Consequentemente, os elementos de INL_1 tem um limite superior e inferior.
ii) Se A é um elemento de INL_2, então $A \leq_{INL_2} (1,0,0,0)$ e $A \geq_{INL_2} (0,0,0,1)$. Consequentemente, os elementos de INL_2 tem um limite superior e inferior.

Demonstração. O resultado é consequência direta da definição de $\leq_{INL_1}$ e $\leq_{INL_2}$. □

Teorema 4.6.5. *Se o valor de verdade em PNL_1 é limitado por max_t, então PNL_1 tem tanto um limite superior quanto inferior.*

Demonstração. Faça $A = (t_1, i_1, f_1)$ ser um elemento de PNL_1. Dado que $t_1 \leq max_t$, segue que $A \leq_{PNL_1} (max_t, 0, 0)$. Por definição dos elementos de PNL_1, $t_1 + i_1 + f_1 \geq 1.0$, assim $A \leq_{PNL_1} (0,0,1)$. □

4.7 Regras de Inferênia em NL1

A Lógica Neutrosófica pode ser usada para construir um sistema de raciocínio onde premissas podem ser usadas para inferir ou justificar conclusões. Para começar este processo, é necessário definir o que se entende por inferir e a notação usada para a representar.

Definição 4.7.1. *Suponha $A = (t_1, i_1, f_1)$ e que $A \rightarrow B = (t_2, i_2, f_2)$ em NL_1. Então podemos inferir B com os valores$B = (t_3, i_3, f_3)$ onde*

$t_3 = t_2$ *se* $t_2 > f_1$
$t3 = 0.0$ *se* $f_1 \leq t_2$
$f_3 = f_2$ *se* $f_2 < t_1$
$f_3 = 0.0$ *se* $f_2 \geq t_1$
$i_3 = 1.0 - t_3 - f_3$

Esta regra de inferência é a regra modus ponens para NL_1 ($MPNL_1$).

A justificativa para esta regra é como segue. Comece com os valores para $A = (t_1, i_1, f_1)$ e a definição da implicação em termos da negação e conjunção.

$$A \rightarrow B = \neg A \vee B = (t_2, i_2, f_2)$$

Se $B = (t_3, i_3, f_3)$, então $\max\{f_1, t_3\} = t_2$ e $\min\{t_1, f_3\} = f_2$. Se $t_2 > f_1$, então segue que $t_2 = t_3$. Contudo, se $f_1 \leq t_2$, então não temos informação sobre t_3 assim não fazemos assunções e fazemos ser igual a zero. Se $f_2 < t_1$, então segue que $f_3 = f_2$. Novamente, se $f_2 \geq t_1$, não temos informação sobre o valor de f_3, de modo que o fazemos ser igual a zero. A computação de i_3 é feita do modo usual tomando a diferença de 1.0.

Exemplos:

Faça $A = (1,0,0)$ e $A \to B = (1,0,0)$. Então podemos inferir $B = (1,0,0)$ aplicando $MPNL$. Isto é consistente com a regra modus ponens para a lógica clássica, ode se A e $A \to B$ são verdadeiros, podemos inferir que B também é verdadeiro.

Se $A = (0,0,1)$ e $A \to B = (1,0,0)$, então a definição poderia produzir $B = (1,0,0)$. Isto também é consistente com a regra da lógica clássica que uma hipótese falsa pode ser usada para provar nada.

Definição 4.7.2. *Suponha que $A = (t_1, i_1, f_1)$ e que $A \to B = (t_2, i_2, f_2)$ em NL_2. Então podemos inferir B com os valores $B = (t_3, i_3, f_3)$ onde*

$i_3 = i2$ *se* $i_2 \geq i1$
$i_3 = 1.0$ *se* $i_1 > i_1$
$t_3 = 0.0$ *se* $i_3 = 1.0$
$t_3 = t_2$ *se* $t_2 \geq t_1$ *e* $t_2 < 1 - i_3$
$t_3 = 0.0$ *se* $t_2 \geq 1 - i_3$ *ou* $t_2 \leq 1 - i_3$
$f_3 = 1.0 - i_3 - t_3$

Esta regra de inferência é a regra modus ponens para NL_2 ($MPNL_2$).

A justificativa para esta regra é similar a daquela de $MPNL_1$. Comece com os valores para $A = (t_1, i_1, f_1)$ e a definição da implicação em termos da negação e conjunção.

$$A \to B = \neg A \vee B = (t_2, i_2, f_2)$$

Se $i_1 > i_2$, então não sabemos nada sobre o valor de i_3, assim o fazemos igual a 1.0.

A computação do valor verdade é dado por

$$\min\{1 - i_3, \max\{t_1, t_3\}\} = t_2$$

Se $t_2 \leq t_1$ e $t_2 < 1 - i_3$, segue que o valor de t_2 é aquele de t_3. Nada pode ser inferido sobre o valor de t_3 para os outros casos, assim o fazemos igual a 0.0.

Faça $A = (t_1, i_1, f_1, u_1)$ e $A \to B = (t_2, i_2, f_2, u_2)$ serem elementos de INL_1. Usando a forma alternativa da implicação, fazendo $B = (t_3, i_3, f_3, u_3)$ e aplicando as definições de $\neg$ e $\vee$ para INL_1, temos

$$(f_1, i_1, t_1, u_1) \vee (t_3, i_3, f_3, u_3) = (t_2, i_2, f_2, u_2)$$

assim,

$\max\{t_1, t_3\} = t_2$
$\min\{i_1, i_3\} = i_2$

$\min\{f_1, f_3\} = f_2$
$u_2 = 1 - t_2 - i_2 - f_2$

Que nos dá um modo natural de definir modus ponens em INL_1.

Definição 4.7.3. *Dado* $A = (t_1, i_1, f_1, u_1)$ *e* $A \to B = (t_2, i_2, f_2, u_2)$ *elementos de* INL_1*, podemos inferir* $B = (t_3, i_3, f_3, u_3)$ *com os seguintes valores*

$t_3 = t_2$ *se* $t_2 \leq t_1$
$t_3 = 0.0$ *se* $t_2 < t_1$
$i_3 = i_2$ *se* $i_2 \geq i_1$
$i_3 = 0.0$ *se* $i_2 > i_1$
$f_3 = f_2$ *se* $f_2 \geq f_1$
$f_3 = 0.0$ *se* $f_2 > f_1$
$u_3 = 1 - t_3 - i_3 - f_3$

Esta regra de inferência é a regra modus ponens para INL_1 *(*$MPINL_1$*).*

A justificativa para esta definição de modus ponens é similar a daquela para NL_1 e NL_2.

Definição 4.7.4. *Dado* $A = (t_1, i_1, f_1, u_1)$ *e* $A \to B = (t_2, i_2, f_2, u_2)$ *serem elementos de* INL_2*, podemos inferir* $B = (t_3, i_3, f_3, u_3)$ *com os seguintes valores*

$t_3 = t_2$ *se* $t_2 \geq t_1$
$t_3 = 0.0$ *se* $t_2 < t_1$
$u_3 = u_2$ *se* $u_2 \leq u_1$
$u_3 = 0.0$ *se* $u_2 > u_1$
$f_3 = f_2$ *se* $f_2 \leq f_1$
$f_3 = 0.0$ *se* $f_2 > f_1$
$i_3 = 1 - t_3 - u_3 - f_3$

Esta regra de inferência é a regra modus ponens para INL_2 *(*$MPINL_2$*).*

As diferenças nas definições de modus ponens para INL_1 e INL_2 são devidas aos diferentes papéis que as partes desconhecidas e indeterminadas encena. Para INL_1, o desconhecido é o valor computado a partir de todos os outros e, portanto, tem a menor significância. Em INL_2, o valor indeterminado que é computado a partir de todos os outros.

Faça $A = (t_1, i_1, f_1)$ e $A \to B = (t_2, i_2, f_2)$ serem elementos de PNL_1. Usando a forma equivalente da implicação $(\neg A \vee B)$ e fazendo $B = (t_3, i_3, f_3)$, aplicando as definições dos conectivos, temos as seguintes expressões.

$\max\{t_1, t_3\} = t_2$
$\max\{i_1, i_3\} = i_2$

$\min\{f_1, f_3\} = f_2$

Que fornece a justificativa para as definições de modus ponens em PNL_1.

Definição 4.7.5. *Dado* $A = (t_1, i_1, f_1)$ *e* $A \to B = (t_2, i_2, f_2)$ *elementos de* PNL_1*, podemos inferir* $B = (t_3, i_3, f_3)$ *com os seguintes valores*

$t_3 = t_2$ *se* $t_2 \geq t_1$
$t_3 = 0.0$ *se* $t_2 < t_1$
$f_3 = f_2$ *se* $f_2 \leq f_1$
$f_3 = 0.0$ *se* $f_2 > f_1$
$i_3 = \max\{1.0 - t_3 - f_3, \max\{i_1, i_2\}\}$

Neste caso, dado que a soma dos valores deve ser no mínimo 1.0, segue que o potencial para as porções verdadeira e falsa serem zero requer que as definições do valor indeterminado produza um valor que é no mínimo 1.0.

4.8 Teorias Formais em Lógica Neutrosófica

Definição 4.8.1. *O seguinte é a definição da teoria formal* LNL_1 *em lógica neutrosófica.*

a) *O conjunto de símbolos* $\{\neg, \vee, \wedge, \to, (,), T, F, A, B, \cdots\}$*, onde os símbolos* $A, B, \cdots$ *são abreviações da forma* (t, i, f) *onde* $t + i + f = 1.0$ *e as constantes* T*,* I *e* F *que são abreviações para* $(1,0,0), (0,1,0)$ *e* $(0,0,1)$*, respectivamente.*

b)

1) *As constantes* T*,* I *e* F *são bem formadas.*

2) *Os símbolos* $A, B, \ldots$ *são bem formados.*

3) *Se* A *é bem formado, então também o são* $\neg A$ *e* (A).

4) *Se* A *e* B *são bem formadas, então o são* $A \vee B$*,* $A \wedge B$ *e* $A \to B$.

5) *Somente expressões que podem ser formadas usando as regras 1–4 são bem formadas.*

c) *Se* A *e* B *são bem formadas, então o que segue são os axiomas de* LNL_1.

1) $A \to (A \vee B)$ *tem o valor* $(1,0,0)$. LNL_{1A_1} *(Axioma 1,* LNL_1*)*

2) $(A \wedge B) \to A$ *tem o valor* $(1,0,0)$. LNL_{1A_2} *(Axioma 2,* LNL_1*)*

3) $(A \wedge B) \to B$ *tem o valor* $(1,0,0)$. LNL_{1A_3} *(Axioma 3,* LNL_1*)*

d) *A única regra de inferência é* $MPNL_1$ *(Modus Ponens* NL_1*).*

Definição 4.8.2. *A teoria formal* LNL_2 *pode ser definida tomando as propriedade de (a)-(c) de* LNL_1 *com renomeação adequada e seguindo a seguinte definição alternatica de (d).*

d) *A única regra de inferência é* $MPNL_2$*. (modus ponens* NL_2*)*

Definição 4.8.3. *O que segue é a definição da teoria formal NL_1 na lógica neutrosófica.*

a) O conjunto de símbolos $\{\neg, \vee, \wedge, \rightarrow, (,), T, F, A, B, \ldots\}$, *onde os símbolos* $A, B, \ldots$ *são abreviação para 4-tuplas da forma* (t, i, f, u) *onde* $t + i + f + u = 1.0$ *e as constantes* T, I, F *e* U *são abreviações para* $(1,0,0,0)$, $(0,1,0,0)$, $(0,0,1,0)$ *e* $(0,0,0,1)$, *respectivamente.*

b)

1) As constantes T, I, F *e* U *são bem formadas.*

2) Os símbolos $A, B, \ldots$ *são bem formados.*

3) Se A *é bem formado, então também o são* $\neg A$ *e* (A).

4) Se A *e* B *são bem formadas, então também o são* $A \vee B$, $A \wedge B$ *e* $A \rightarrow B$.

5) Somente expressões que podem ser formadas usando as regras 1–4 são bem formadas.

c) Se A *e* B *são bem formadas, então o que segue são os axiomas de* NL_1.

1) $A \rightarrow (A \vee B)$ *tem o valor* $(1,0,0,0)$. $NL_{1_{A_1}}$ *(Axioma 2,* NL_1*)*

2) $(A \wedge B) \rightarrow A$ *tem o valor* $(1,0,0,0)$. $NL_{1_{A_2}}$ *(Axioma 2,* NL_1*)*

3) $(A \wedge B) \rightarrow B$ *tem o valor* $(1,0,0,0)$. $NL_{1_{A_3}}$ *(Axioma 3,* NL_1*)*

d) A única regra de inferência é MPNL_1 *(modus ponens* NL_1*).*

Definição 4.8.4. *A teoria formal* INL_2 *pode ser definida tomando as propriedade de (a)-(c) de* $ILNL_1$ *com adequada renomeação e a seguinte definição alternativa de (d).*

d) A única regra de inferência é $MPINL_2$. *(modus ponens* INL_2*)*

Definição 4.8.5. *O que segue e a definição da teoria formal* PNL_1 *na lógica neutrosófica.*

a) O conjunto de símbolos $\{\neg, \vee, \wedge, \rightarrow, (,), A, B, \ldots\}$, *onde os símbolos* $A, B, \ldots$ *são abreviações para triplas da forma* (t, i, f) *onde* $t + i + f \geq 1.0$.

b)

1) Os símbolos $A, B, \ldots$ *são bem formados.*

2) Se A *é bem formado, assim o é* $\neg A$ *e* (A).

3) Se A *e* B *são bem formadas, então assim o são* $A \vee B$, $A \wedge B$ *e* $A \rightarrow B$.

4) Somente expressões que podem ser formadas usando as regras 1–4 são bem formadas.

c) Se A *e* B *são bem formadas, então o que segue são os axiomas de* PNL_1.

1) $A \rightarrow (A \vee B)$ *tem um valor onde o componente verdade é maior ou igual a 1.0.* $PNL_{1_{A_1}}$ *(Axioma 1,* PNL_1*)*

2) $(A \wedge B) \rightarrow A$ *tem um valor onde o componente verdade é maior ou igual a 1.0.* $PNL_{1_{A_2}}$ *(Axioma 2,* PNL_1*)*

3) $(A \wedge B) \rightarrow B$ *tem um valor onde o componente verdade é maior ou igual a 1.0.* $PNL_{1_{A_3}}$ *(Axioma 3,* PNL_1*)*

d) A única regra de inferência é $MPPNL_1$ (modus ponens PNL_1).

Teorema 4.8.1. *Se $A \to B$ tem o valor $(1,0,0)$ em LNL_1 (LNL_2) e $A \neq F$, então podemos inferir $B = (1,0,0)$ em LNL_1 (LNL_2).*

Prova: Faça $A = (t,i,f) \neq F$ ser um elemento arbitrário de LNL_1. Consequentemente, f deve ser menor do que 1, assim por $MPNL_1$ o primeiro elemento da tripla de B é o primeiro elemento de $A \to B$, que é 1. Por definição, os outros dois valores da tripla devem ser zero e $B = (1,0,0)$. Dada que a definição de $MPNL_2$ é a mesma daquela de $MLNL_1$ para o os componentes verdade e falsidade, o teorema também é verdadeiro para LNL_2.

Corolário: Faça A e B serem elementos de LNL_1 (LNL_2).
i) Se $A \neq F$, então podemos inferir $(A \vee B) = T$.
ii) Se $(A \wedge B) \neq F$, então podemos inferir $A = T$.
iii) Se $(A \wedge B) \neq F$, então podemos inferir $B = T$.

Prova:

i) Aplique o teorema 4.8.1 em $LNL_{1_{A_1}}$ ($LNL_{2_{A_1}}$).
ii) Aplique o teorema 4.8.1 em $LNL_{1_{A_2}}$ ($LNL_{2_{A_2}}$).
iii) Aplique o teorema 4.8.1 em $LNL_{1_{A_3}}$ ($LNL_{2_{A_3}}$).

Teorema 4.8.2. *Se $A \to B$ tem o valor $(1,0,0,0)$ em NL_1 (NL_2) e $A \neq F$, então nós podemos inferir $B = (1,0,0,0)$ em NL_1 (NL_2).*

Prova: Faça $A = (t,i,f,u) \neq F$ ser um elemento arbitrário de NL_1. Consequentemente, f deve ser menor do que 1, assim, por $MPNL_1$, o primeiro elemento da quádrupla de B é o primeiro elemento de $A \to B$, que é 1. Por definição, os outros três valores da 4-tupla devem ser zero e $B = (1,0,0,0)$. Visto que a definição de $MPNL_2$ é a mesma daquela de $MPNL_1$ para os componentes verdade e falsidade, o teorema também é verdadeiro para NL_2.

Teorema 4.8.3. *Se $A \to B$ tem um valor verdadeiro que é maior ou igual a 1.0 em PNL_1 e o valor falso de A é menor do que 1.0, então podemos inferir que B tem um valor verdadeiro maior ou igual a 1.0.*

Prova: Faça $A = (t,i,f)$ onde $f < 1.0$, então o valor verdade de $\neg A$ deve ser menor do que 1.0. Consequentemente, dado que o valor verdade de $\neg A \vee B$ é maior ou igual a 1.0, segue de $MPPNL_1$ que o valor de B deve ser maior ou igual a 1.0.

Os Teoremas 4.8.1, 4.8.2 e 4.8.3 apontam algumas das dificuldades que existem quando se define axiomas para a lógica neutrosófica. Assinalando aos axiomas o valor de $(1,0,0)$, tudo o que necessitamos são antecedentes diferentes de zero para provar que a consequência é T. isto é um resultado mais forte do que queremos ou esperamos.

Exemplo: Construa a teoria formal LNL_{12} usando a definição de LNL_1, com exceção de que o seguinte lista de axiomas deve ser usada.

$(A \to (B \to A))$ tem o valor T. $LNL_{12_{A_1}}$ (Axioma A_1, LNL_1)
$((A \to (B \to C)) \to ((A \to B) \to (A \to C)))$ tem o valor T. $LNL_{12_{A_2}}$ (Axioma A_2, LNL_1)
$((\neg B \to \neg A) \to ((\neg B \to A) \to B))$ tem o valor T. $LNL_{12_{A_3}}$ (Axioma A_3, LNL_1)

Teorema 4.8.4. *Se A, B e C são elementos de LNL_{12}:*

i) Se $A \neq F$, então $(B \to A)$ tem o valor T.
ii) Se $((A \to (B \to C)) \neq F$, então $((A \to B) \to (A \to C))$ tem o valor T.
iii) Se $(\neg B \to \neg A) \neq F$, então $((\neg B \to A) \to B)$ tem o valor T.

Prova: Idêntica a daquela do teorema 4.8.1.

Teorema 4.8.5. *Se A, B e C são elementos de LNL_2:*

i) Se $A \neq T$, então $((A \to B) \to (A \to C))$ tem o valor T.
ii) Se $B \neq T$ ou $C \neq F$, então $((A \to B) \to (A \to C))$ tem o valor T.
iii) Se $B \neq F$ ou $A \neq T$, então $((\neg B \to A) \to B)$ tem o valor T.

Demonstração.

i) Dado que $A \neq T$, podemos aplicar a definição dos conectivos *vee* e $\neg$ para concluir que $\neg A \vee (B \to C)$ não é F. Dado que isto é uma abreviação para $A \to (B \wedge C)$, esta expressão também não é F. Aplicamos então o Teorema 4.8.4 com $LNL_{2_{A_2}}$ para inferir que $((A \to B) \to (A \to C))$ tem o valor T.

ii) Se $B \neq T$ ou $C \neq F$, então $\neg A \vee (\neg B \vee C)$ não é falso. Esta expressão é uma abreviação para $A \to (B \to C)$, assim ela também não é falsa. Podemos então aplicar o Teorema 4.8.4 para concluir que $((A \to B) \to (A \to C))$ tem o valor T.

iii) Se $B \neq F$, então $\neg B \neq T$ e se $A \neq F$, então $\neg A \neq T$. Consequentemente, $B \vee \neg A \neq F$ e por abreviação de $\neg B \to A \neq F$. Podemos aplicar o Teorema 4.8.4 com $LNL_{2_{A_3}}$ para concluir que $((\neg B \to A) \to B)$ tem o valor T. □

O conjunto alternativo de axiomas $LNL_{12_{A_1}}$, $LNL_{12_{A_2}}$ e $LNL_{12_{A_3}}$ também pode ser usado para criar teorias formais adicionais com NL_1, NL_2 e PNL_1. Dado que todos são baseados na representação de $A \to B$ como $\neg A \vee B$ e envolvem o valor de T, o teorema 4.8.5 é verdadeiro para NL_1, NL_2 e PNL_1, onde a lista de axiomas pe trocada por

$(A \to (B \to A))$ tem o valor T.
$((A \to (B \to C)) \to ((A \to B) \to (A \to C)))$ tem o valor T.
$((\neg B \to \neg A) \to ((\neg B \to A) \to B))$ tem o valor T.

4.9 Raciocínio em Lógica Neutrosófica

Definição 4.9.1. *Se A e B são expressões bem formadas em LNL_1, LNL_2, INL_1, INL_2 ou PNL_1, então B é dita ser uma consequência de A, denotado por $A \vdash B$, se o valor da expressão B pode ser inferida para ser maior ou igual a daquela da expressão A. Uma expressão equivalente poderia ser que B pode ser inferida de A. A expressão*

$$\vdash A$$

somente é válida quando $A = T$. No outro caso de $A \vdash B$, A é a hipótese e B a conclusão. A sequência de passos que começam com A e levam a B é conhecida como uma prova.

Definição 4.9.2. *Se A e B são expressões bem formadas em PNL_1, então B é dita ser uma consequência de A, denotado por $A \vdash B$, se o valor da expressão B pode ser inferido para ser maior ou igual a da expressão de A. Uma expressão equivalente seria que B pode ser inferida a partir de A. A expressão*

$$\vdash A$$

somente é válida quando o valor verdade de A é maior ou igual a 1.0.

Teorema 4.9.1. *Se A é um elemento de LNL_1, LNL_2, INL_1, INL_2 ou PNL_1, então*

(i) $A \vdash A$
(ii) $A \vee A \vdash A$
(iii) $A \wedge A \vdash A$
(iv) $A \vdash A \vee A$
(v) $A \vdash A \wedge A$
(vi) $A \vdash \neg\neg A$
(vii) $\neg\neg A \vdash A$

Demonstração. (Somente para LNL_1).

(i) Dado que $A \leq_{NL_1} A$, a definição é satisfeita.
(ii) Foi provado que $A \vee A = A$, assim o problema simplifica-se para o caso (i).
(iii) Foi provado que $A \wedge A = A$, assim o problema simplifica-se para o caso (i).
(iv) Similar ao caso (ii).
(v) Similar ao caso (iii).
(vi) Foi provado que $\neg\neg A = A$, assim o problema simplifica-se para o caso (i).
(vii) Similar ao caso (vi). □

Teorema 4.9.2. *Se A, B e C são elementos de LNL_1, LNL_2, INL_1, INL_2 ou PNL_1 então o operador $\vdash$ é*

i) $A \vdash A$. (Reflexividade.)
ii) Se $A \vdash B$ e $B \vdash A$, então $B = A$. (Anti-simetria)
iii) Se $A \vdash B$ e $B \vdash C$, então $A \vdash C$. (Transitividade)

Demonstração. (Somente para LNL_1).

i) Este é o resultado do Teorema 4.9.1 parte (i).
ii) Se $A \vdash B$, então $A \leq_{NL_1} B$ e se $B \vdash A$, então $B \leq_{NL_1} A$. Isto é válido se, e somente se, $A = B$.
iii) Se $A \vdash B$, então a expressão A pode ser usada para inferir que o valor de B é maior ou igual a A. Se $B \vdash C$, então a expressão B pode ser usada para inferir que o valor de C é maior ou igual a B. Primeiro execute os passos que começam em A e levam a B. Depois disto executando os passos que começam em B e levam a C. A combinação será então uma sequência de passos que começam em A e levam a C. Consequentemente, A pode ser usado para inferir C e $A \vdash C$. □

Teorema 4.9.3. *Se A é qualquer fbf de LNL_1 ou LNL_2.*

(i) $(0,1,0) \vdash A$.
(ii) $A \vdash (1,0,0)$.
(iii) $(1,0,0) \vdash A$ *se, e somente se,* $A = (1,0,0)$.
(iv) Se $A = (t,i,f)$, *então* $(0,0,1) \vdash A$ *se, e somente se,* $t > 0.0$ *ou* $A = (0,0,1)$.

Demonstração. Prova somente para LNL_1, a prova para LNL_2 é similar.

(i) Dado que $(0,1,0) \leq_{NL_1} A$ para todo A em LNL_1, o resultado é imediato.
(ii) Dado que $(1,0,0) \geq A$ para todo A em LNL_1, o resultado é imediato.

(iii) (Se) Dado que $(1, 0, 0) \geq A$ com igualdade se, e somente se, $A = (1, 0, 0)$, o resultado é imediato. (Somente se) Se $A = (1, 0, 0)$ o resultado segue do teorema anterior.
(iv) (Se) Assuma que $(0, 0, 1) \vdash A$ e suponha que $t = 0.0$ e $A \neq (0, 0, 1)$. Então $i > 0.0$ e $f < 1.0$. Por definição, $A < (0, 0, 1)$ e não pode ser inferido a partir de $(0, 0, 1)$. Consequentemente, t deve ser maior do que zero ou $A = (0, 0, 1)$. (Somente se) Se $t > 0.0$ ou $A = (0, 0, 1)$, então $A \geq (0, 0, 1)$ e $(0, 0, 1) \vdash A$. □

Teorema 4.9.4. *Se A é qualquer fbf de INL_1.*

(i) $(0, 0, 0, 1) \vdash A$.
(ii) $A \vdash (1, 0, 0, 0)$.
(iii) $(1, 0, 0, 0) \vdash A$ *se, e somente se,* $A = (1, 0, 0, 0)$.
(iv) Se $A = (t, i, f, u)$ *então* $(0, 0, 1, 0) \vdash A$ *se, e somente se,* $t > 0.0$ *ou* $A = (0, 0, 1, 0)$.

Demonstração.

i) Dado que $(0, 0, 0, 1) \leq_{ILNL_1} A$ para todo A em INL_1, o resultado é imediato.
ii) Dado que $(1, 0, 0, 0) \geq_{ILNL_1} A$ para todo A em INL_1, o resultado é imediato.
iii) (Se) Dado que $(1, 0, 0, 0) \geq_{ILNL_1} A$ com igualdade se, e somente se, $A = (1, 0, 0, 0)$, o resultado é imediato. (Somente se) Se $A = (1, 0, 0, 0)$ o resultado segue do teorema anterior.
iv) (Se) Assuma que $(0, 0, 1, 0) \vdash A$ e suponha que $t = 0.0$ e $A \neq (0, 0, 1, 0)$. Então, $i > 0.0$ e $f < 1.0$. Por definição, $A <_{ILNL_1} (0, 0, 1, 0)$ e não pode ser inferido a partir de $(0, 0, 1, 0)$. Consequentemente, t deve ser maior do que zero ou $A = (0, 0, 1, 0)$. (Somente se) Se $t > 0.0$ ou $A = (0, 0, 1, 0)$, então $A \geq_{ILNL_1} (0, 0, 1, 0)$ e $(0, 0, 1, 0) \vdash A$. □

Teorema 4.9.5. *If A is any wff of INL2.*

(i) $(0, 1, 0, 0) \vdash A$.
(ii) $A \vdash (1, 0, 0, 0)$.
(iii) $(1, 0, 0, 0) \vdash A$ *se, e somente se,* $A = (1, 0, 0, 0)$.
(iv) Se $A = (t, i, f, u)$ *então* $(0, 0, 1, 0) \vdash A$ *se, e somente se,* $t > 0.0$ *ou* $A = (0, 0, 1, 0)$.

Demonstração.
i) Dado que $(0, 1, 0, 0) \leq_{ILNL_2} A$ para todo A em INL_2, o resultado é imediato.

As provas de (ii) a (iv) são similares a aquela do Teorema 4.9.4. □

Teorema 4.9.6. *Se A é qualquer fbf de PNL_1:*

(i) $(0, 0, 0) \vdash A$.
(ii) Se existe um valor máximo t_max para o valor verdade em PNL_1, então $A \vdash (t_max, 0, 0, 0)$.

Demonstração.
i) Dado que $(0, 0, 0) \leq_{PNL_1} A$ para todo A em PNL_1, o resultado é imediato.
ii) Se t_max é o maior valor possível para o componente verdade, então $(t_max, 0, 0, 0) \geq_{PNL_1} A$ para todo A em PNL_1. □

A estratégia usada na prova das várias seções do teorema anterior podem ser generalizadas, o que é ponto do próximo teorema.

Teorema 4.9.7. *Suponha que A e B são bem formadas em LNL_1. Se $A =_{NL_1} B$, então $A \vdash B$ e $B \vdash A$. O mesmo resultado permanece para LNL_2, INL_1, INL_2 e PNL_1.*

Demonstração. Assuma que $A =_{NL_1} B$. Por definição, isto significa que os valores das expressões é o mesmo. Dado que o valor de B é maior ou igual a A. $A \vdash B$ por definição. Dado que a igualdade é reflexiva, a segunda inferência também permanece. As provas para LNL_2, INL_1, INL_2 e PNL_1 são similares e são omitidas. □

Corolário: Se A, B e C são bem formadas em LNL_1, então todas as inferências seguintes são válidas:

(i) $A \vee B \vdash B \vee A$.
(ii) $A \wedge B \vdash B \wedge A$.
(iii) $(A \vee B) \vee C \vdash A \vee (B \vee C)$.
(iv) $(A \wedge B) \wedge C \vdash A \wedge (B \wedge C)$.
(v) $(A \vee B) \wedge A \vdash A$.
(vi) $(A \wedge B) \vee A \vdash A$.
(vii) $(1, 0, 0) \wedge A \vdash A$.
(viii) $(0, 0, 1) \vee A \vdash A$.
(ix) $(0, 0, 1) \wedge A \vdash (0, 0, 1)$.
(x) $(1, 0, 0) \vee A \vdash (1, 0, 0)$.
(xi) $A \wedge (B \vee C) \vdash (A \wedge B) \vee (A \wedge C)$.
(xii) $(A \wedge B) \vee (A \wedge C) \vdash A \wedge (B \vee C)$.
(xiii) $A \vee (B \wedge C) \vdash (A \vee B) \wedge (A \vee C)$.
(xiv) $(A \vee B) \wedge (A \vee C) \vdash A \vee (B \wedge C)$.
(xv) $\neg(A \vee B) \vdash \neg A \wedge \neg B$.
(xvi) $\neg A \wedge \neg B \vdash \neg(A \vee B)$.
(xvii) $\neg(A \wedge B) \vdash \neg A \vee \neg B$.
(xviii) $\neg A \vee \neg B \vdash \neg(A \wedge B)$.

Fórmulas similares, apesar de não serem todas, também são válidas em LNL_2, INL_1, INL_2 e PNL_1.

Demonstração. Todas estas inferências são consequências dos teoremas já provados com relação as propriedades algébricas de LNL_1. □

Teorema 4.9.8. *Se A e B são bem formadas em LNL_1:*

i) Se $A <_{LNL_1} B$, então $A \vdash (A \vee B)$.
ii) Se $A >_{LNL_1} B$, então $A \vdash (A \vee B)$ se, e somente se, $(A \vee B) = A$.
iii) Se $A <_{LNL_1} B$, então $A \vdash (A \wedge B)$.
iv) Se $A >_{LNL_1} B$, então $A \vdash (A \wedge B)$ se, e somente se, $(A \wedge B) = A$.

Demonstração. Faça $A = (t_1, i_1, f_1)$, $B = (t_2, i_2, f_2)$, $A \vee B = (t_3, i_3, f_3)$ e $A \wedge B = (t_4, i_4, f_4)$.

i) **Caso 1:** $t_2 > t_1$.
Então $t_3 = t_2 > t_1$ e $(A \vee B) > A$.

Caso 2: $t_1 = t_2$ e $f_1 < f_2$.
Então, $t_3 = t_1$ e $f_3 = f_1$, que significa que $(A \vee B) = A$.

ii)

(Se) **Caso 1:** $t_1 > t_2$.
Então $t_3 = t_1$ e $(A \vee B) \leq A$ se, e somente se, $f_3 \leq f_1$. Dado que $f_3 = \min\{f_1, f_2\}$, isto pode ocorrer se $f_1 = f_3$. Isto faria $(A \vee B) = A$.

Caso 2: $t_1 = t_2$
Isto significaria também que $t_3 = t_1$ e se reduz ao caso 1.

(Somente se) Se $(A \vee B) = A$, então isto se reduz a $A \vdash A$, que foi provado no Teorema 4.9.2.

iii) **Caso 1:** $t_2 > t_1$
Então $t_3 = t_1$ e $f_3 = \max\{f_1, f_2\} \leq f_1$. Portanto, pela definição, $(A \wedge B) > A$.

Caso 2: $t_2 = t_1$
Então $t_3 = t_1$ e isto se reduz ao caso 1.

iv)
(Se) **Caso 1:** $t_1 > t_2$
Então $t_3 = \min\{t_1, t_2\} = t_2 < t_1$ e segue que $(A \wedge B) < A$.

Caso 2: $t_1 = t_2$ e $f_1 > f_2$.
Então $f_3 = \max\{f_1, f_2\} = f_1$ e $A = (A \wedge B)$.

(Somente se) Se $(A \wedge B) = A$, então isto se reduz a $A \vdash A$, que foi provado no Teorema 4.9.2.

Podemos pensar do Theorema 4.9.8 como aquele que permite "adicionar matéria", na medida em que ele nos permite inferir uma expressão que é maior do que a original. Estes teoremas podem, de fato, serem generalizados para um número arbitrário de adições. □

Teorema 4.9.9. *Se $A <_{NL_1} B_i$ para todo $B_1, B_2, \ldots, B_k$, então*

i) $A \vdash (A \vee B_1 \vee B_2 \vee \ldots \vee B_k)$.
ii) $A \vdash (A \wedge B_1 \wedge B_2 \wedge \ldots \wedge B_k)$.

Demonstração. A prova irá confiar nos resultados anteriores expressos em corolários.

Faça $A = (t_1, i_1, f_I)$ e $(A \vee B_1 \vee B_2 \vee \ldots \vee B_k) = (t_2, i_2, f_2)$.

i) Pelo corolário 1, t_2 é o maior de todos os primeiros elementos das triplas em $(A \vee B_1 \vee B_2 \vee \ldots \vee B_k)$. Existem dois casos a considerar:

Caso 1: t_1 não é o maior dos primeiros elementos das triplas. Então $t_2 > t_1$ e $A <_{Nl_1} (A \vee B_1 \vee B_2 \vee \ldots \vee B_k)$, o que implica

$$A \vdash (A \vee B_1 \vee B_2 \vee \ldots \vee B_k)$$

Caso 2: t_1 é igual ao maior dos primeiros elementos da triplas. Então $f_1 \leq f_2$ e $A <_{NL_1} (A \vee B_1 \vee B_2 \vee \ldots \vee B_k)$, o que implica

$$A \vdash (A \vee B_1 \vee B_2 \vee \ldots \vee B_k)$$

ii) Por hipótese, dado que $A < B_i$ para todo i, t_1 é o menor dos primeiros elementos em $(A \wedge B_1 \wedge B_2 \wedge \ldots \wedge B_k)$ e segue que $t_1 = t_2$. Dado que f_1 deve ser menor ou igual a todas as terceiras entradas nas triplas em $(A \wedge B_1 \wedge B_2 \wedge \ldots \wedge B_k)$, $f_2 \geq f_1$. Portanto, $(A \wedge B_1 \wedge B_2 \wedge \ldots \wedge B_k) \geq A$ e $A \vdash (A \wedge B_1 \wedge B_2 \wedge \ldots \wedge B_k)$. □

Teorema 4.9.10. *Se A e B são bem formadas em LNL_2*

i) Se $A <_{LNL_2} B$, então $A \vdash (A \vee B)$.
ii) Se $A >_{LNL_2} B$, então $A \vdash (A \vee B)$ se, e somente se, $(A \vee B) = A$.
iii) Se $A <_{LNL_2} B$, então $A \vdash (A \wedge B)$.
iv) Se $A >_{LNL_2} B$, então $A \vdash (A \wedge B)$ se, e somente se, $(A \wedge B) = A$.

Demonstração. A prova é similar a aquela do Theorema 4.9.8 e também é baseada na definição da desigualdade em LNL_2. □

Resultados similares a aqueles dos Teoremas 4.9.8 e 4.9.9 também são válidos para INL_1, INL_2 e PNL_1. As provas de todas são baseadas nas definições de ordenação dos elementos na teoria formal.

Definição 4.9.3. *Faça $x_1, x_2, \ldots, x_k, y$ serem elementos de NL_1. Então, y pode ser inferido do conjunto de expressões $H = x_1, x_2, \ldots, x_k$, escrito $x_1, x_2, \ldots, x_k \vdash y$, se dadas as expressões em H, y tem um valor maior ou igual a $\min\{x_1, x_2, \ldots, x_k\}$. Os elementos em H são conhecidos como* ***hipótese****. Neste caso, o operador de desigualdade usado para computar o mínimo é $\leq_{NL_1}$.*

Definições similares podem ser escritas com NL_1 substituída por NL_2, INL_1, INL_2 ou PNL_1.

Teorema 4.9.11. *Faça $\{x_1, x_2, x_3, \ldots, x_k\}$ ser qualquer conjunto de expressões em NL_1 (NL_2) e para as expressões no conjunto $x_i = (t_i, i_i, f_i)$, $1 \leq i \leq k$.*

i) $x_1, x_2, x_3, \ldots, x_k \vdash x_i$, for all i, $1 \leq i \leq k$.
ii) $x_1, x_2, x_3, \ldots, x_k, (0,1,0) \vdash y$, para qualquer expressão y.
iii) $x_1, x_2, x_3, \ldots, x_k \vdash (0,1,0)$.
iv) $x_1, x_2, x_3, \ldots, x_k \vdash (x_1 \wedge x_2 \wedge \ldots \wedge x_k)$.
v) $x_1, x_2, x_3, \ldots, x_k \vdash (x_1 \vee x_2 \vee \ldots \vee x_k)$.

Demonstração.

i) Dado que $x_i \geq \min\{x_1, x_2, \ldots, x_k\}$, o resultado é imediato.
ii) Dado que $y \geq (0,1,0)$ para todo y em NL_1 (NL_2), o resultado é imediato.
iii) Dado que $x_i \geq (0,1,0)$ para todo $1 \leq i \leq k$, o resultado é imediato.
iv) Por definição, $(x_1 \wedge x_2 \wedge \ldots \wedge x_k) = (t,i,f)$, onde $t = \min\{t_1, t_2, \ldots, t_k\}$ e $f = \max\{f_1, f_2, \ldots, f_k\}$. Faça $x_i = \min\{x_1, x_2, \ldots, x_k\}$, onde $x_i = (t_i, i_i, f_i)$. Pela definição da ordenação dos elementos $t = t_i$ e $f \geq f_i$, que produz $(t,i,f) \geq (t_i, i_i, f_i)$.
v) Por definição, $(x_1 \vee x_2 \vee \ldots \vee x_k) = (t,i,f)$, onde $t = \max\{t_1, t_2, \ldots, t_k\}$ e $f = \min\{f_1, f_2, \ldots, f_k\}$. Existem dois casos a considerar.

Caso 1: Os valores de $t_1, t_2, \ldots, t_k$ não são todos iguais. Então existe um máximo t_max e um mínimo t_min, onde $t_max > t_min$. Claramente, $t = t_max$ e o valor verdade de $\min\{x_1, x_2, \ldots, x_k\}$ deve ser t_min. Consequentemente, o valor de $(x_1 \vee x_2 \vee \ldots \vee x_k)$ deve ser maior do o valor mínimo da hipótese.

Caso 2: Os valores de $t_1, t_2, \ldots, t_k$ são todos iguais. Então a hipótese com o menor valor será a expressão com o menor valor para o componente falso. A expressão $(x_1 \vee x_2 \vee \ldots \vee x_k)$ terá um valor verdade correspondente comum a todas as hipóteses e um valor falso igual a aquele da hipótese tendo o menor componente falso. Isto significa que $\min\{x_1, x_2, \ldots, x_k\} = (x_1 \vee x_2 \vee \ldots \vee x_k)$, que satisfaz a definição de inferência. □

Teorema 4.9.12. *Faça $\{x_1, x_2, x_3, \ldots, x_k\}$ serem quaisquer expressões em INL_1 e para as expressões no conjunto $x_i = (t_i, i_i, f_i, u_i)$, $1 \leq i \leq k$.*

i) $x_1, x_2, x_3, \ldots, x_k \vdash x_i$, *para todo* i, $1 \leq i \leq k$.
ii) $x_1, x_2, x_3, \ldots, x_k, (0, 0, 0, 1) \vdash y$, *para qualquer expressão* y.
iii) $x_1, x_2, x_3, \ldots, x_k \vdash (0, 0, 0, 1)$.
iv) $x_1, x_2, x_3, \ldots, x_k \vdash (x_1 \wedge x_2 \wedge \ldots \wedge x_k)$.
v) $x_1, x_2, x_3, \ldots, x_k \vdash (x_1 \vee x_2 \vee \ldots \vee x_k)$.

Demonstração. Similar ao Theorema 4.9.11. □

Teorema 4.9.13. *Faça $\{x_1, x_2, x_3, \ldots, x_k\}$ serem quaisquer conjunto de expressões em INL_2 e para as expressões no conjunto $x)i = (t_i, i_i, f_i, u_i)$, $1 \leq i \leq k$.*

i) $x_1, x_2, x_3, \ldots, x_k \vdots x_i$, *para todo* i, $1 \leq i \leq k$.
ii) $x_1, x_2, x_3, \ldots, x_k, (0, 1, 0, 0) \vdash y$, *para qualquer expressão* y.
iii) $x_1, x_2, x_3, \ldots, x_k \vdash (0, 1, 0, 0)$.
iv) $x_1, x_2, x_3, \ldots, x_k \vdash (x_1 \wedge x_2 \wedge \ldots \wedge x_k)$.
v) $x_1, x_2, x_3, \ldots, x_k \vdash (x_1 \vee x_2 \vee \ldots \vee x_k)$.

Demonstração. Similar a do Theorema 4.9.11. □

Teorema 4.9.14. *Faça $\{x_1, x_2, x_3, \ldots, x_k\}$ ser qualquer expressão em PNL_1 e para as expressões no conjunto $x_i = (t_i, i_i, f_i, u_i)$, $1 \leq i \leq k$.*

i) $x_1, x_2, x_3, \ldots, x_k \vdash x_i$, *para todo* i, $1 \leq i \leq k$.
ii) $x_1, x_2, x_3, \ldots, x_k, (0, 0, 0) \vdash y$, *para qualquer expressão* y.
iii) Se existe um valor máximo (t_{max}) para o valor verdade de uma expressão em PNL_1, então $x_1, x_2, x_3, \ldots, x_k \vdash (t_max, 0, 0)$.
iv) $x_1, x_2, x_3, \ldots, x_k \vdash (x_1 \wedge x_2 \wedge \ldots \wedge x_k)$.
v) $x_1, x_2, x_3, \ldots, x_k \vdash (x_1 \vee x_2 \vee \ldots \vee x_k)$.

Demonstração. Similar a do Theorema 4.9.11. □

Teorema 4.9.15. *Suponha que* $\{x_1, x_2, x_3, \ldots, x_k, y, z\}$ *são expressões em* NL_1 *e adicionalmente, suponha que* $x_1, x_2, x_3, \ldots, x_k \vdash y$ *e* $y = z$*. Então* $x_1, x_2, x_3, \ldots, x_k \vdash z$*. Resultados similares também são verdadeiros em* NL_2*,* INL_1*,* INL_2 *e* PNL_1*.*

Demonstração. Dado que $min\{x_1, x_2, x_3, \ldots, x_k\} \leq y = z$, seque que $min\{x_1, x_2, x_3, \ldots, x_k\} \leq z$ e $x_1, x_2, x_3, \ldots, x_k \vdash z$ por definição. A prova em outras teorias é idêntica dado que ela é baseada somente na habilidade de ordenar os elementos. □

Teorema 4.9.16. *Faça* $\{x_1, x_2, x_3, \ldots, x_k, y_1, y_2, \ldots, y_n, z\}$ *serem expressões em* NL_1*. Assuma que para todo i,* $1 \leq i \leq n$*,* $x_1, x_2, x_3, \ldots, x_k, \vdash y_i$ *e que* $y_1, y_2, \ldots, y_n \vdash z$*. Então* $x_1, x_2, x_3, \vdots, x_k, \vdash z$*. Resultados similares também são verdadeiros em* NL_2*,* INL_1*,* INL_2 *e* PNL_1*.*

Demonstração. De $x_1, x_2, x_3, \ldots, x_k, \vdash y_i$, segue que $\min\{x_1, x_2, x_3, \ldots, x_k\} \leq y_i$ para todo $1 \leq i \leq n$ e de $y_1, y_2, \ldots, y_n \vdash z$, $\min\{y_1, y_2, \vdots, y_n\} \leq z$. Consequentemente, dado que $\leq$ é transitiva em LNL_1, segue que $x_1, x_2, x_3, \ldots, x_k \vdash z$. □

Referências Bibliográficas

[1] D. A. Bochvar. On a three-valued logical calculus and its application to the analysis of contradictions. *Rec. Math. Moscou, n. Ser.*, 4:287–308, 1938.

[2] Edward V. Huntington. Sets of independent postulates for the algebra of logic. *Transactions of the American Mathematical Society*, 5(3):288–309, 1904.

[3] Jan Łukasiewicz. O logice trójwartościowej. *Ruch filozoficzny*, 5:170–171, 1920.

[4] Elliot Mendelsohn. *Introduction to Mathematical Logic.* Springer, 1987.

[5] Florentin Smarandache. *A Unifying Field in Logics: Neutrosophic Logic.* American Research Press, 1999.

[6] Florentin Smarandache. *Neutrosophy: Neutrosophic Probability, Set, and Logic: Analytic Synthesis & Synthetic Analysis.* American Research Press, 1998.

[7] Florentin Smarandache, editor. *Proceedings of the First International Conference on Neutrosophy, Neutrosophic Logic, Neutrosophic Set, Neutrosophic Probability and Statistics.* University of New Mexico, 2002.

[8] Alfred Tarski. *Introduction to Logic and to the Methodology of the Deductive Sciences.* Oxford University Press, 1941.

[9] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338 – 353, 1965.

Introdução à Lógica Neutrosófica

Lógica neutrosófica é um novo tipo de lógica não clássica surgida em meados dos anos 90 do século passado, sendo uma extensão/generalização das lógicas trivalentes que usam um valor indeterminado, lógica fuzzy e intuicionista fuzzy, caracterizando-se por incluir a indeterminação diretamente em seu arcabouço teórico.

Na presente obra, os primeiros três capítulos são uma clara introdução às lógicas clássica, trivalentes e fuzzy, as quais servem de prefação ao quarto capítulo, o qual é devotado à lógica neutrosófica. Neste, os muitos aspectos da lógica neutrosófica são pormenorizados de maneira clara e objetiva.

ISBN: 978-65-00-05717-1

9 786500 057171