



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
MESTRADO ACADÊMICO EM CIÊNCIA DA COMPUTAÇÃO

HÉLDER PAIXÃO FELIX FILHO

**ANÁLISE DIDATICAMENTE ESTRUTURADA DAS
CORRESPONDÊNCIAS ENTRE LÓGICAS DE
DESCRIÇÕES E LÓGICAS MODAIS**

Recife

2022

HÉLDER PAIXÃO FELIX FILHO

**ANÁLISE DIDATICAMENTE ESTRUTURADA DAS
CORRESPONDÊNCIAS ENTRE LÓGICAS DE
DESCRIÇÕES E LÓGICAS MODAIS**

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial à obtenção do título de mestre em Ciência da Computação.

Área de Concentração: Teoria da Computação

Orientador: **Prof. Dr. Frederico Luiz Gonçalves de Freitas**

Co-Orientadora: Profa. Dra. Anjolina Grisi de Oliveira

Recife

2022

Catálogo na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

F316a Felix Filho, Hélder Paixão
Análise didaticamente estruturada das correspondências entre lógicas de descrições e lógicas modais / Hélder Paixão Felix Filho. – 2022.
130 f.: il., fig., tab.

Orientador: Frederico Luiz Gonçalves de Freitas.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2022.
Inclui referências e apêndices.

1. Teoria da computação. 2. Lógica modal. 3. Lógica de descrições. I. Freitas, Frederico Luiz Gonçalves de (orientador). II. Título.

004 CDD (23. ed.)

UFPE - CCEN 2022 – 98

Hélder Paixão Felix Filho

**" Análise didaticamente estruturada das correspondências entre Lógicas
de Descrições e Lógicas Modais"**

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial à obtenção do título de mestre em Ciência da Computação.

Área de Concentração: Teoria da Computação

Aprovado em: 23/02/2022

Orientador: Prof. Dr. Frederico Luiz Gonçalves de Freitas

BANCA EXAMINADORA

Profª. Dra. Anjolina Grisi de Oliveira
Centro de Informática / UFPE
(Co-Orientadora)

Prof. Dr. Ruan Vasconcelos Bezerra Carvalho
Departamento de Computação / UFRPE

Prof. Dr. Cleyton Mário de Oliveira Rodrigues
Escola Politécnica de Pernambuco / UPE

Este trabalho é dedicado às crianças adultas que, quando pequenas, sonharam em se tornar cientistas.

AGRADECIMENTOS

Se palavras fossem suficientes, nem mesmo tudo o que foi escrito pela humanidade seria remotamente suficiente. Por ordem temporal (na tentativa de ser justo estabelecendo alguma), gratidão a Anjolina, paciência e doçura personificadas, por ter confiado e apostado em mim, derradeiramente estabelecendo a conexão à continuação da formação. Gratidão a Fred, que, apesar da condição de orientador, foi imensuravelmente mais que somente o que a função revela. Um oceano de paciência, virtude essencial ao enfrentar tudo o que precisamos, ainda mais sob o contexto em que estávamos.

a Lilo, por estar do meu lado incansavelmente.

a Tia Dulce e Tia Zé, minhas santas, as remanescentes vivas Damas da Realidade.

Não quero mal às ficções, amo-as, acredito nelas, acho-as preferíveis às realidades; nem por isso deixo de filosofar sobre o destino das coisas tangíveis em comparação com as imaginárias. Grande sabedoria é inventar um pássaro sem asas, descrevê-lo, fazê-lo ver a todos, e acabar acreditando que não há pássaros com asas...

Fonte: (ASSIS, 1894, p. 2)

RESUMO

Este trabalho se propõe a apresentar uma análise sobre a correspondência entre ramificações da Lógica Modal com ramificações da Lógica de Descrições. Além disso, este texto tem como principal proposta, pela falta de um material mais palatável, preencher esta lacuna e tornar mais acessível um assunto naturalmente denso lógico e matematicamente. A apresentação didática dos formalismos é feita nos capítulos iniciais; da Lógica de Descrições foi escolhida a linguagem $\mathcal{ALC}$ e da Lógica Modal, o sistema $\mathbf{K}$. É demonstrado por que a correspondência não ocorre a princípio, por isso o sistema $\mathbf{K}$ é enriquecido para a versão multimodal $\mathbf{K}_m$. Posteriormente é apresentada a Lógica Híbrida: ramificação modal que completa a correspondência com uma base de conhecimento de $\mathcal{ALC}$. Por fim, propositalmente alocado no apêndice, são escolhidas ramificações que apresentam correspondência entre si: da Lógica de Descrições a linguagem $\mathcal{ALC}_{reg}$ e a Lógica Dinâmica Proposicional, que é ramificação (modalidade) da Lógica Modal.

Palavras-chave: lógica modal; lógica de descrições; lógica híbrida; lógica dinâmica; PDL; $\mathcal{ALC}_{reg}$.

ABSTRACT

This work proposes an analysis of the correspondence between ramifications of Modal Logic with ramifications of Logic of Descriptions. In addition, due to the lack of more palatable material, this text has as its main proposal to fill this gap and make a naturally dense subject logically and mathematically more accessible. The didactic presentation of formalisms is made in the opening chapters; for the Description Logic, the language $\mathcal{ALC}$ was chosen, and for the Modal Logic, the **K** system. It is shown why matching does not occur at first, so the **K** system is enriched for the multimodal version $\mathbf{K}_m$. Afterward, the Hybrid Logic is presented: a modal branch that completes the correspondence with a knowledge base of $\mathcal{ALC}$. Finally, intentionally allocated in the appendix, branches that correspond to each other are chosen: from Description Logic, the language $\mathcal{ALC}_{reg}$ and Propositional Dynamic Logic, which is a branch (modality) of Modal Logic.

Keywords: modal logic; description logic; dynamic logic; hybrid logic; PDL; $\mathcal{ALC}_{reg}$.

LISTA DE FIGURAS

Figura 1 – Camadas da <i>Web Semântica</i>	23
Figura 2 – Instâncias dos conceitos: Pessoa(João), Pessoa(Maria); País(Brasil), País(Japão).	27
Figura 3 – Instâncias da relação nasceu: nasceu(João, Japão), nasceu(Maria, Brasil).	28
Figura 4 – <i>Framework</i> de DL	29
Figura 5 – Representação gráfica da ontologia do Exemplo 4.	32
Figura 6 – Representação gráfica da ontologia do Exemplo 8	35
Figura 7 – Visualização do <i>frame</i> do Exemplo 9.	41
Figura 8 – Representação gráfica do frame do Exemplo 10.	42
Figura 9 – Representação gráfica do modelo do Exemplo 11.	43
Figura 10 – Modelo de Kripke do Exemplo 12	44
Figura 11 – Ilustração das configurações dos conjuntos em $\mathcal{ALC}$ e $\mathbf{K}$	48
Figura 12 – Ilustração do Exemplo 15 em que "Data" resume a data por extenso, "h" representa a hora de tal data com a seção hachurada e "(p)" representa a proposição analisada em tal nominal, que é verdade apenas nele.	53
Figura 13 – Representação gráfica da ontologia do Exemplo 4 em DL $\mathcal{ALC}$	55
Figura 14 – Representação gráfica de parte da mesma ontologia do Exemplo 4 em Lógica Híbrida do Exemplo 16; Observe-se que as instâncias estão representadas por conjuntos unitários e que a relação de acessibilidade <i>dirige</i> (d) liga tanto instâncias quanto classes.	55
Figura 15 – Ilustração da análise de comportamento da conjunção.	57
Figura 16 – Ilustração da análise de comportamento da disjunção.	57
Figura 17 – Ilustração da análise de comportamento da dupla negação.	58
Figura 18 – Ilustração da análise de comportamento de um <i>tableau</i> simples.	58
Figura 19 – Ilustração da análise de comportamento do operador de implicação.	59
Figura 20 – <i>Tableau</i> correspondente ao Exemplo 18.	61
Figura 21 – <i>Tableau</i> correspondente ao Exemplo 19.	62
Figura 22 – <i>Tableau</i> correspondente à fórmula $\mathbf{K}$	63
Figura 23 – <i>Tableau</i> correspondente ao Exemplo 21	65
Figura 24 – Diagrama que organiza algumas lógicas modais normais por inclusão própria	77
Figura 25 – Diagrama que organiza lógicas modais normais	78
Figura 26 – Árvore binária que ilustra o Exemplo 27	90

Figura 27 – Diagrama que ilustra o exemplo 29.	93
Figura 28 – Representação gráfica dos autômatos correspondentes às linguagens mais expressivas, respectivamente $\{RS, R\}$ e $\{R, RSR\}$	101
Figura 29 – Representação dos autômatos-resultado do Exemplo 34	107
Figura 30 – Uma breve árvore genealógica DL: comparando a expressividade da Lógica de Descrições, onde $\mathcal{ALC} \leq L \leq \mathcal{ALC}_{reg}$. Se houver um caminho de uma lógica L_2 até uma lógica L_1 que contenha ou uma aresta normal ou pelo menos duas arestas, então L_2 é mais expressiva que L_1 com respeito a conceitos positivos, Tboxes e Aboxes. Se o caminho for uma aresta pontilhada então L_2 é mais expressiva que L_1 com respeito a conceitos, conceitos positivos e Tboxes. Se o caminho for uma aresta tracejada, então L_2 é mais expressiva que L_1 com respeito a conceitos, conceitos positivos e Aboxes.	111
Figura 31 – Diagrama relacional referente ao exemplo 40	120
Figura 32 – Grafo que exhibe o ponto de insatisfatibilidade do modelo.	122
Figura 33 – Regras de análise para Lógicas Modais Normais	124

LISTA DE TABELAS

Tabela 1 – Visualização das equivalências entre DL com LPO	24
Tabela 2 – Semântica dos operadores como em Baader et al. (2010)	30
Tabela 3 – Comparação entre sentenças da Abox e fórmulas puras da LH.	54

LISTA DE QUADROS

Quadro 1 – Sistemas Modais Normais.	76
Quadro 2 – Hierarquia de linguagens	125

LISTA DE ABREVIATURAS E SIGLAS

$\mathcal{ALC}$	Attributive Concept Language
ABox	Assertional Axioms
AL	Attributive Language
$\mathcal{ALC}_{reg}$	Attributive Concept Language with regular expressions
DL	Description Logic
DPDL	Strict Propositional Dynamic Logic
DWP	Deterministic While Programs
$\mathcal{FL}^-$	Enriched Frame Language
$\mathcal{FL}_0$	Frame Language
$\mathcal{FL}_{reg}$	Frame Language with regular expressions
$\mathcal{FL}_{trans}$	Frame Language with transitive closure
gfp	Greatest Fixpoint
HML	Hennessy-Milner logic
KB	Knowledge Base
KR	Knowledge Representation
lfp	Least Fixpoint
LH	Lógica Híbrida
LPO	Lógica de Primeira Ordem
LTS	Labeled Transition Systems
MDL	Dynamic Logic
MI	Material Implication
NTF	Language in Hybrid Representation Systems
OWL	Ontology Web Language
PDL	Propositional Dynamic Logic
PML	Propositional Modal Logic
SI	Strict Implication
Tbox	Terminological Axioms
W3C	World Wide Web Consortium

LISTA DE SÍMBOLOS

$\cup$	União
π	Número pi
γ	Número gamma
Γ	Número Gamma
$\circ$	Composição
$*$	Fecho de Kleene
K	Sistema modal normal canônico
$\exists$	Quantificação existencial
$\top$	Tautologia
$\perp$	Absurdo
$\sim$	Relação de equivalência
$\sqcup$	Disjunção em DL
r, s	Papéis em DL
A, B, C	Conceitos em DL
$\neg$	Negação
Δ	Universo de discurso, domínio de interpretação
I	Função de interpretação
$\Delta^1, *^1$	Elemento interpretado por
$:=$	Definição
$\emptyset$	Conjunto vazio
$\setminus$	Diferença de conjuntos
$\cap$	Intersecção
$\in$	Relação de pertinência
$\rightarrow, \supset$	Implicação material clássica
T	T-box
$\leftrightarrow$	Necessário e suficiente
$=$	Igualdade
W	Palavra
$\mathbf{R}_T$	Conjunto de papéis de T

ϵ	Palavra vazia
A, M	Autômatos
A_T	Autômato associado a T-box
f, g	Funções
$L_{\emptyset}$	Linguagem regular
J	Restrição de interpretação
$T_{\mathbf{J}}$	T-box sob interpretação
$\Delta^{\mathbf{J}}$	Domínio sob J
$\geq$	Maior ou igual
$\preceq$	Ordem parcial
$\subseteq_{gfp, T}$	Classificação sob maior ponto fixo em T
R	Conjunto finito de papéis
L	Linguagem qualquer em DL
$\multimap$	Implicação estrita, strict implication
$\Box, \blacksquare$	Necessidade modal
$\Diamond, \blacktriangleleft$	Possibilidade modal
Σ^*	Alfabeto de cadeias
p, q	Proposições
$\wp$	Conjunto de proposições
$\mathcal{F}$	Conjunto de fórmulas
$\mathcal{W}$	Conjunto de mundos possíveis
$\mathcal{R}$	Relação de acessibilidade
v	Função de valoração, avaliação
0	Binário falso
$\mathcal{S}$	Satisfatibilidade
v	v mundos possíveis
$\models$	Dedução
$\mathcal{A}, \mathcal{M}$	Modelos de Kripke
$\mathcal{R}_{\mathcal{A}}$	Acessibilidade sob modelo $\mathcal{A}$
$v_{\mathcal{A}, \mathcal{M}}$	Valoração sob modelos

ϕ, ψ	Fórmulas
$<$	Ordem parcial estrita
F	Frame de Kripke
v_F	Valoração sob o frame de Kripke F
$\vee$	Disjunção clássica
$\stackrel{\text{def}}{=}$	Igualdade por definição
$\wedge$	Conjunção clássica
$\neq$	Diferença
Σ	Alfabeto de símbolos
$\mathcal{M}_\Sigma$	Modelo sob alfabeto Σ
$\mathcal{W}_\Sigma$	Mundos sob alfabeto Σ
$\mathcal{R}_\Sigma$	Acessibilidade sob alfabeto Σ
v_Σ	Valoração sob alfabeto Σ
π, a, b	Transições, programas
Φ_0	Conjunto de fórmulas atômicas
Π_0	Conjunto de transições (programas) atômicas
$;$	Operador de execução sequencial
$\cup$	Operador de escolha não-determinística
$*$	Operador de iteração
Φ	Conjunto de fórmulas
Π	Conjunto de programas
K	Conjunto de estados
u, v	estados
$\#$	Cardinalidade
$\mu X.$	Menor ponto fixo de X
$\nu X.$	Maior ponto fixo de X
$\pi()$	Função de mapeamento

SUMÁRIO

1	INTRODUÇÃO	18
1.1	OBJETIVO DO TRABALHO	19
1.1.1	Objetivo Geral	19
1.1.2	Objetivo específico: correspondências entre Lógicas de Descrições e Mo-	
	dais	20
1.2	ORGANIZAÇÃO DO TRABALHO	21
2	LÓGICA DE DESCRIÇÕES	23
2.1	SINTAXE	25
2.2	SEMÂNTICA	30
2.3	RACIOCÍNIO EM DL	31
3	LÓGICA MODAL	36
3.1	SINTAXE E SEMÂNTICA	38
4	CORRESPONDÊNCIA DEMONSTRADA	46
4.1	ENRIQUECIMENTO DO SISTEMA K : A MULTIMODALIZAÇÃO K_M .	46
4.2	CORRESPONDÊNCIA LIMITADA À TBOX	48
4.2.1	Correspondência com K	48
4.2.2	Correspondência com K_m	49
4.3	CORRESPONDÊNCIA PARA ABOX: A LÓGICA HÍBRIDA DO OPERA-	
	DOR @	51
5	CORRESPONDÊNCIA SOB A PERSPECTIVA DE RACIOCÍNIO AU-	
	TOMÁTICO	56
6	CONCLUSÕES	66
	REFERÊNCIAS	68
	APÊNDICE A – LÓGICAS MODAIS NORMAIS E MAPEAMENTO	
	PARA LPO	76
	APÊNDICE B – MAPEAMENTO ENTRE EXTENSÕES MAIS RICAS	84
	APÊNDICE C – TEOREMAS E PROVAS	118
	APÊNDICE D – APORTE	125

1 INTRODUÇÃO

Os estudos das lógicas de Descrições e Modais são arcabouço para apresentar uma equivalência com respeito à expressividade entre pares de ramos específicos destas linguagens formais (BAADER et al., 2010). A consolidação da Lógica de Descrições se deu através de sua utilização na *web* semântica, ao se tornar necessária devido ao seu poder de expressividade aliado ao fato de ser um fragmento decidível da Lógica de Primeira Ordem (LPO). Na literatura, é possível encontrar autores que, quando dissertam sobre uma ramificação da Lógica de Descrições, sempre fazem menção à correspondência entre as vertentes da Lógica Modal (BAADER; HORROCKS; SATTLER, 2008), (BAADER et al., 2007), (RIJKE, 1998).

A Lógica de Descrições (DL) nem sempre foi chamada assim. Era intitulada Lógica Terminológica, oriunda de melhorias de outros formalismos de representação de conhecimento, as redes semânticas e os *frames* (BRACHMAN; SCHMOLZE, 1985 apud BAADER, 1991, p. 3). Nestes formalismos também é possível definir domínios, como se pode na Lógica de Descrições. Apesar dessas semelhanças, a diferença fundamental das linguagens anteriores para a Lógica de Descrições reside no aspecto da formalidade lógica e matemática: somente neste formalismo é possível encontrar uma linguagem formal não ambígua, bem definida e fundamentada matematicamente. Meghini e Straccia (1996) escreve que ela seria apresentada como um modelo de "recuperação" de conhecimento, que daria às suas relações de inferência o sabor de relevância, ou seja, uma conexão estrita em significado entre as premissas e a conclusão. As Lógicas de Descrições passaram a ser bastante estudadas e aplicadas após serem incorporadas na *Onthology Web Language* (OWL, Linguagem de Ontologia da Rede em português) (PATEL-SCHNEIDER; HAYES; HORROCKS, 2004) e tornou-se linguagem padrão da *web* semântica (BEZERRA, 2019). Além disso, nos âmbitos do direito e da biologia, ramificações das Lógicas de Descrições são utilizadas e preferidas em detrimento a tantas outras lógicas. Devido a tal, uma vasta gama de fragmentos de Lógicas de Descrições é usada e pesquisada, sendo o fragmento *ALC* (linguagem atributiva com uniões e complementos, do inglês, *Attributive Language with Unions and Complements*)(BAADER; HORROCKS; SATTLER, 2008) considerado o mais próximo do sistema mais básico modal, mas com expressividade minimamente superior.

Vários fragmentos de DL correspondem a sistemas da Lógica Modal. As primeiras aparições do conceito de modalidade em lógica, ou modos da verdade, remetem a Aristóteles (FELIX FILHO, 2019). As modalidades dizem respeito à interpretação filosófica dos operadores básicos do formalismo da Lógica Modal "necessário" e "possível". Nesta lógica, é estudada

a semântica dos mundos possíveis de Kripke. Nela, os mundos possíveis são conjuntos onde são interpretadas as proposições e interligam-se através de uma relação chamada relação de acessibilidade.

Nas modalidades em que se faz o uso de mais de uma versão destes operadores devido a uma semântica mais complexa, são utilizados índices nos operadores e a esta lógica é atribuído o título de multimodal. Como exemplo, existe uma modalidade temporal, em que podem ser analisadas as "necessidades" e "possibilidades" no passado e no futuro, sendo respectivamente "foi sempre o caso" e "às vezes foi o caso"; "será sempre o caso" e "será às vezes o caso" (GORANKO; RUMBERG, 2021).

1.1 OBJETIVO DO TRABALHO

1.1.1 Objetivo Geral

Inicialmente, esta dissertação se propõe a trabalhar a respeito da problemática da correspondência entre lógicas de Descrições e Modais. Nesse sentido, uma densa revisão da literatura revelou a carência de um texto único que descrevesse minuciosamente os formalismos em questão, bem como onde e de quais formas as equivalências (entre as lógicas de Descrições e lógicas Modais) ocorrem. Para além de um compilado de definições formais e mapeamentos, este texto tem como principal proposta, pela falta de um material mais palatável, preencher esta lacuna e tornar mais acessível um assunto naturalmente denso matematicamente e complexo filosoficamente.

Na literatura, é possível encontrar vários exemplos de correspondências envolvendo as lógicas de Descrições e Modais (RIJKE, 1998). Tanto em (BAADER; HORROCKS; SATTTLER, 2008), quanto em (BAADER et al., 2007), há mencionada a correspondência de ramificações específicas de cada uma das linguagens formais, cada uma com sua particularidade e propósito específico. Este trabalho se propõe a listá-las e, mais que isto, entregar de modo mais matematicamente acessível, bem detalhada e didaticamente estruturada, começando desde as ramificações mais básicas: da Lógica Modal, o sistema **K** (e sua versão multimodal) e, da Lógica de Descrições, a linguagem $\mathcal{ALC}$. A primeira correspondência se dá exatamente entre estas linguagens, e, através dela, confere ao sistema **K** a característica de ser um fragmento decidível da Lógica de Predicados (ou de Primeira Ordem), atributo inerente à Lógica de Descrições.

A análise comparativa entre estas duas lógicas, de Descrições e Modais, será concluída com a apresentação da correspondência também de raciocínio automático através do método de *tableaux* (BAADER; SATTTLER, 2000), (GORÉ, 1999). Toda esta discussão está repleta de

exemplos ilustrativos que auxiliarão o entendimento.

1.1.2 Objetivo específico: correspondências entre Lógicas de Descrições e Modais

Como será possível inferir do texto, a correspondência nem sempre é direta. A partir de e após a leitura dele, o leitor vai ser capaz de entender que a correspondência isomórfica só ocorre a partir de versões enriquecidas do sistema $\mathbf{K}$ da Lógica Modal e de $\mathcal{ALC}$ da Lógica de Descrições. Em comparação a $\mathbf{K}$, que é monomodal, $\mathcal{ALC}$ é multimodal. Sendo assim, faz-se necessário uma lógica mais expressiva para que acompanhe esta expressividade, no caso, o sistema multimodal $\mathbf{K}_m$. Por outro lado, será visto que este sistema corresponde apenas à parte terminológica de $\mathcal{ALC}$, ou seja, a parte em que é definida a terminologia com os axiomas de cada classe, bem como as relações entre elas; já a parte de asserções —em que são definidas as instâncias destas classes e relações— apenas poderá ser correspondida por um enriquecimento do sistema $\mathbf{K}$ chamado Lógica Híbrida (LH).

Este é o objetivo primordial do texto: mostrar a correspondência entre a Lógica de Descrições $\mathcal{ALC}$ e o sistema $\mathbf{K}$ e $\mathbf{K}_m$ da Lógica Modal de forma bastante ilustrada, trazendo exemplos, os modelos de ambas as Lógicas, bem como uma análise comparativa usando como elemento comum o sistema de *tableaux*.

É trazido ainda no texto uma correspondência mais complexa. Ao se tentar estabelecer a correspondência a partir de uma ramificação Modal essencialmente multimodal, a Lógica Dinâmica Proposicional, conhecida como PDL (do inglês, *Propositional Dynamic Logic*), a percepção de PDL ser mais expressiva que $\mathcal{ALC}$ é direta. A Lógica Dinâmica Proposicional consiste num formalismo multimodal que melhor corresponderia a $\mathcal{ALC}$, não fosse por ela, neste caso, ser agora mais expressiva. Somente com o enriquecimento de $\mathcal{ALC}$ para a versão $\mathcal{ALC}_{\text{reg}}$ é que pode ser estabelecida correspondência completa na sintática e semântica. Apesar de ser uma lógica modal multimodal, a PDL embora seja expressiva o suficiente para que seja possível construir uma correspondência em nível de Abox, esta organização não faz parte regularmente da linguagem. Esta correspondência foi alocada propositalmente no apêndice (B) devido ao nível de complexidade.

Essencialmente, as correspondências entre as ramificações que são apresentadas com maior riqueza de detalhes aqui podem ser encontradas em (RIJKE, 1998) e (BAADER et al., 2010). Outras referências pontuais serão feitas em situações específicas, como (OHLBACH; KOEHLER, 1999, p. 2). É comum encontrar na literatura, quando se menciona a Lógica

Dinâmica Proposicional, a menção da correspondência com um enriquecimento de $\mathcal{ALC}$, que contém transitividade e reflexividade em sua estrutura, chamada $\mathcal{SR}$. Entretanto, neste trabalho, foi escolhido uma ramificação diferente para a Lógica de Descrições, chamada $\mathcal{ALC}_{reg}$, que trata definições cíclicas e expressões regulares sobre papéis.

1.2 ORGANIZAÇÃO DO TRABALHO

No Capítulo 2, são detalhadas as características de Lógica de Descrições, com ênfase na ramificação $\mathcal{ALC}$ desta família de formalismos. Esta linguagem é apresentada com detalhes, sintática e semanticamente, as ABox e TBox, além dos modos de raciocínio fundamentais praticados. Vários exemplos acompanham cada uma das definições, entregando uma construção didática do conhecimento.

No Capítulo 3, Lógica Modal, são apresentados os fundamentos de uma lógica tal que seja modal. As sintaxe e semântica fundamentais do sistema monomodal $\mathbf{K}$ são apresentadas, bem como são citadas outras modalidades, enriquecimentos a partir de $\mathbf{K}$, com interpretações filosóficas distintas, características para cada uma.

O Capítulo 4, intitulado Correspondência Demonstrada, traz em suas seções respectivamente: a evidência que só havia correspondência a nível de TBox e, para tal, o enriquecimento do sistema monomodal $\mathbf{K}$ para sua versão multimodal $\mathbf{K}_m$; a correspondência a nível de ABox, com a introdução de um enriquecimento para o sistema $\mathbf{K}$ que lida com nominais, conjuntos unitários que sozinhos validam uma proposição, chamado de Lógica Híbrida.

No Capítulo 5, é feita uma demonstração das duas lógicas no que diz respeito ao raciocínio automático. São apresentados os métodos de *tableaux* para $\mathcal{ALC}$ e para o sistema $\mathbf{K}$ e sua versão multimodal $\mathbf{K}_m$. Assim será possível perceber a semelhança entre métodos de raciocínio destas lógicas.

Outra correspondência com uma lógica essencialmente Modal, a Lógica Dinâmica e uma ramificação de mais expressiva das Lógicas de Descrições é detalhada no apêndice:

Na seção de Lógica Dinâmica, uma lógica fundamentalmente (multi)modal, é discutido o campo de atuação desta modalidade. Sequencialmente, a sintaxe e semântica de uma versão proposicional são apresentadas. No final deste Capítulo, é apresentado também o método de filtragem como incentivo à comparação com o método para o sistema $\mathbf{K}$.

A seção seguinte se dedica a apresentar ramificações dos dois formalismos, as Lógicas Modal Dinâmica e de Descrições, e as correspondências existentes entre eles. São brevemente

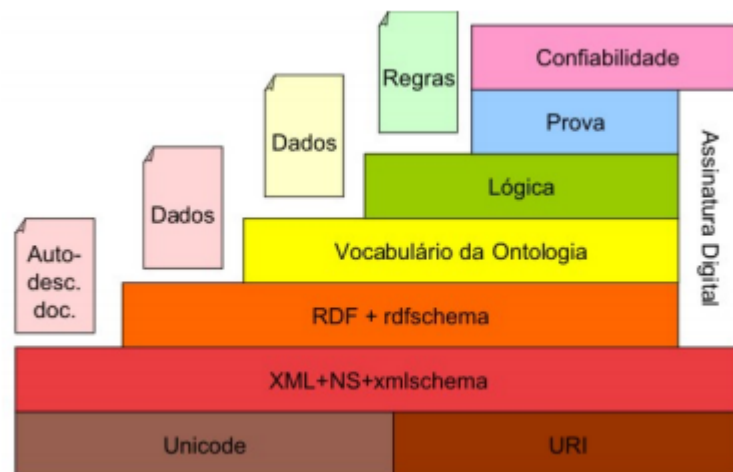
apresentadas linguagens como $\mathcal{ALC}_{\text{reg}}$ (BAADER, 1991).

As conclusões sobre como foi elaborar um trabalho como este e as possibilidades que surgem quando do entendimento notável de ambos os formalismos e suas aplicações fecham o texto.

2 LÓGICA DE DESCRIÇÕES

Lógicas de Descrições (do inglês, *Description Logic*, DL) são uma família de formalismos de representação do conhecimento (do inglês, *knowledge representation*, KR), tal que certos ramos servem como base para a linguagem OWL (PATEL-SCHNEIDER; HAYES; HORROCKS, 2004), aprovada como o padrão W3C (*World Wide Web Consortium*, ou Consórcio da Rede Mundial em português) para representar a camada mais expressiva da *Web Semântica*. São também utilizadas ramificações para representar domínios na biologia e no direito e raciocinar sobre os conceitos importantes como inteligência artificial. Neste capítulo, o foco será a linguagem *ALC*.

Figura 1 – Camadas da *Web Semântica*



Fonte: (BERNERS-LEE; HENDLER; LASSILA, 2001 apud BEZERRA, 2019)

As Lógicas de Descrições são fragmentos decidíveis da Lógica de Primeira Ordem, sem funções, com até duas variáveis.

Em DL, como escreve Bezerra (2019),

a representação de um domínio é constituída de "conceitos", que são termos de um domínio, "propriedades" desses conceitos, e "indivíduos" desse domínio. Além disso, é possível estabelecer relações entre os conceitos, como a relação "É-UM", que permite a criação de uma hierarquia de conceitos, além de axiomas terminológicos.

Assim, sendo elementos-chave da linguagem, é fundamental compreender do que se tratam os conceitos, papéis e instâncias. A seguir, estas ideias são ilustradas e comparadas lado a lado com o que seria sua versão em LPO.

Definição 1. (Conceitos): Um conceito, denotado com palavras iniciadas com letra maiúscula, é um predicado unário que portanto define uma classe de objetos.

CONCEITO: predicados unários, classes, conjuntos de instâncias, subconjunto do domínio; exemplos: Estudante que significa $\{x \mid \text{Estudante}(x)\}$

Casado que significa $\{x \mid \text{Casado}(x)\}$

Definição 2. (Papéis): Um papel é um predicado binário que define relações entre os conceitos (ou instâncias dela). Os papéis serão aqui representados como palavras com inicial minúscula, ou somente letras minúsculas.

PAPEL: predicados binários, relações, conjuntos de pares de instâncias;

amigoDe que significa a relação entre dois objetos x e y , tal que

x é amigo de y $\{(x, y) \mid \text{amigoDe}(x, y)\}$

temConjuge $\{(x, y) \mid \text{temConjuge}(x, y)\}$

Definição 3. (Instâncias): Uma instância é uma constante, um elemento do domínio. São representadas aqui como nomes próprios, portanto de inicial maiúscula; podem conter números ou alguma simbologia para ordem, como "Gato1", "Gato2", mas estarão sempre no escopo dos conceitos, assim não é possível confundir com os conceitos propriamente ditos.

INSTÂNCIAS: indivíduos, constantes;

Estudante(Abel)

amigoDe(Abel, Caim)

É possível comparar as correspondências entre LPO e DL através da Tabela 1:

Tabela 1 – Visualização das equivalências entre DL com LPO

LPO	DL
constante	instância
predicado unário	conceito
predicado binário	papel

Fonte: Elaborada pelo autor (2022)

Exemplo 1. O exemplo a seguir exibe a mesma expressão escrita em DL e LPO:

DL	LPO
$\text{Estudante} \sqcap \exists \text{amigo.Casado} \leftrightarrow \{x \mid \text{Estudante}(x) \wedge \exists y(\text{amigo}(x, y) \wedge \text{Casado}(y))\}$	
$\text{Estudante} \sqsubseteq \exists \text{tem.Professor} \leftrightarrow \{x \mid \text{Estudante}(x) \rightarrow \exists y(\text{tem}(x, y) \wedge \text{Professor}(y))\}$	

2.1 SINTAXE

Em DL há a distinção das implicações material e estrita, assim como será visto também na Lógica Modal no Capítulo seguinte (3). Para tratar do conhecimento geral, o conhecimento "intensional" (do inglês, *intensional*), é concebida a Tbox (do inglês, *Terminological Box*) como sendo a terminologia, contendo declarações que descrevem as propriedades gerais dos conceitos. O conhecimento "extensional", ou assertivo, é organizado na Abox (do inglês, *Assertional Box*), cujas atribuições são específicas para os elementos do domínio. Como em Baader et al. (2007) e Baader, Horrocks e Sattler (2008):

Definição 4. (Base de conhecimento ou ontologia): Uma base de conhecimento, ou KB (do inglês, *Knowledge Base*), ou ainda ontologia, é definida como a união entre Tbox e Abox, ou o par ordenado $\langle T, A \rangle$. Segundo Perez apud Bezerra (2019, p. 23) são componentes de uma ontologia: classe ou conceito; indivíduo ou instância; relação ou propriedade; e axioma.

Geralmente, uma teoria em DL, ou base de conhecimento, ou ainda ontologia, tem uma definição composta de duas partes da seguinte forma: Abox é a parte assertiva da base e a Tbox a parte terminológica. Na Abox, ocorre o instanciamento, a atribuição de nomes de instâncias a conceitos e papéis, segundo uma interpretação (RUDOLPH, 2011).

Introduzida por Schmidt-Schauß e Smolka (1991 apud BAADER, 1991, p. 5), a menor DL (completa¹ sob álgebra booleana, ou seja, toda fórmula da DL é válida sob esta álgebra) é a $\mathcal{ALC}$. Abaixo, na Definição 5, é definida a sintaxe usada em $\mathcal{ALC}$:

Definição 5. (Alfabeto): O alfabeto de $\mathcal{ALC}$ é o conjunto dos seguintes símbolos:

- Letras maiúsculas e minúsculas denotam nomes de conceitos, papéis e instâncias;
- $\top$ e $\perp$ denotam o conceito contendo todos os indivíduos e o "conceito vazio" respectivamente;
- $\equiv$ e $\sqsubseteq$ para denotar equivalência de conceitos e subsunção de conceitos respectivamente;
- os quantificadores universais $\forall$ e $\exists$ para denotar restrições aos papéis
- $\cdot$ para separar visualmente um papel quantificado do conceito a que se aplica

¹ Explicação muito breve sobre completude, cuja recíproca é corretude. Há diversas formas de completude: sintática e semântica, expressiva, funcional etc.

- $\sqcap$ (corresponde ao $\wedge$, a conjunção da lógica clássica) e o $\sqcup$ (corresponde ao $\vee$, a disjunção da lógica clássica) e o $\neg$.

A próxima definição (6) orienta sobre como são formadas as expressões legítimas de $\mathcal{ALC}$, fundamental para consolidação do conhecimento. Inspirada em Baader et al. (2017, p. 11), foi preservada sua estrutura.

Definição 6. (Nomes e Descrições): Seja $\mathbf{C}$ um conjunto de *nomes de conceito* e $\mathbf{R}$ um conjunto de *nomes de papéis* disjunto de $\mathbf{C}$. O conjunto de *descrições de conceito* de $\mathcal{ALC}$ sobre $\mathbf{C}$ e $\mathbf{R}$ é definido indutivamente como segue:

- Todo nome de conceito é uma descrição de conceito;
- $\top$ e $\perp$ são descrições de conceito;
- Se C e D são descrições de conceito e r é um nome de papel, então os seguintes também são descrições de conceito:

$C \sqcap D$ (conjunção),

$C \sqcup D$ (disjunção),

$\neg C$ (negação),

$\exists r.C$ (restrição existencial) e

$\forall r.C$ (restrição de valor).

Exemplo 2. Confira graficamente os elementos de uma ontologia:

Conceitos e suas instâncias:

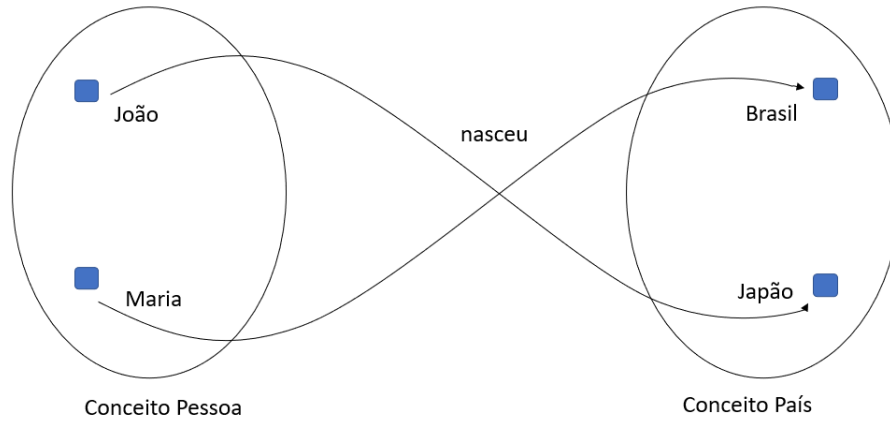
Figura 2 – Instâncias dos conceitos: Pessoa(João), Pessoa(Maria); País(Brasil), País(Japão).



Fonte: Elaborada pelo autor (2022)

Relação nasceu:

Figura 3 – Instâncias da relação nasceu: nasceu(João, Japão), nasceu(Maria, Brasil).



Fonte: Elaborada pelo autor (2022)

Definição 7. (Axiomas): Sejam A e D conceitos. Então $A \equiv D$ e $A \sqsubseteq D$ são *axiomas terminológicos*, respectivamente de equivalência e subsunção (classificação, que funciona como a implicação material clássica), em que $(A \sqsubseteq D) \sqcap (D \sqsubseteq A)$ corresponde a $A \equiv D$. Sejam, agora, C uma descrição de conceito, r um nome de papel e x, y nomes de instâncias. Um *axioma assertivo* possui as seguintes formas² respectivamente: $C(x)$ ou $r(x, y)$. As instanciações são axiomas assertivos.

Exemplo 3.

$\text{Animal} \sqsubseteq \text{CoisaViva}$

$\text{Burro} \equiv \text{Animal} \sqcap \forall \text{temPai.Burro}$

$\text{Cavalo} \equiv \text{Animal} \sqcap \forall \text{temPai.Cavalo}$

$\text{Mula} \equiv \text{Animal} \sqcap \exists \text{temPai.Cavalo} \sqcap \exists \text{temPai.Burro}$

$\exists \text{temPai.Mula} \equiv \perp$

Instância de conceito:

Mula(Heitor)

Neste caso, "Heitor" pertence ao conceito "Mula".

² Outra sintaxe para axiomas assertivos $x : C$ e $(x, y) : r$.

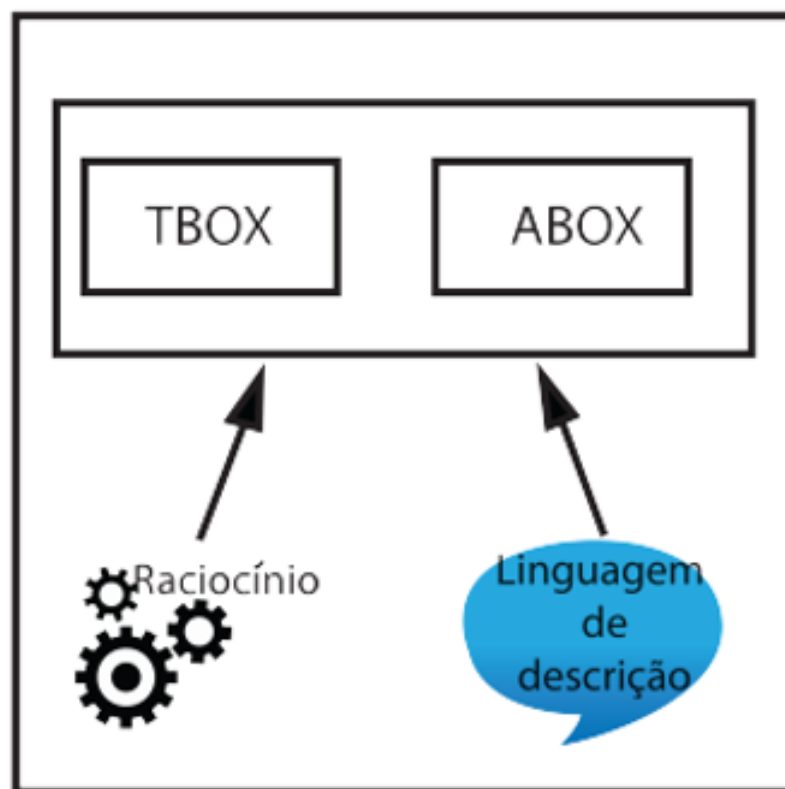
Instância de relação:

temPai(Heitor, Hélder)

Definição 8. (Tbox): Uma terminologia (Tbox) é um conjunto finito de axiomas terminológicos, possivelmente vazio, com a restrição adicional de que nenhum nome de conceito pode aparecer mais de uma vez como o lado esquerdo de um axioma de equivalência. O conjunto de axiomas das Tboxes geram uma taxonomia. Os chamados *conceitos definidos* ocorrem no lado esquerdo de um axioma de equivalência. Os outros conceitos são chamados de *conceitos primitivos*; para a linguagem $\mathcal{ALC}$, os papéis são sempre primitivos, pois não há definições de papéis.

Definição 9. (Abox): Um conjunto finito de axiomas assertivos caracteriza uma Abox.

Figura 4 – *Framework* de DL



Fonte: Baader et al. (2010)

2.2 SEMÂNTICA

Baseada na teoria dos modelos, construída sobre a teoria axiomática dos conjuntos de Zermelo-Fraenkel (HRBACEK; JECH, 1999), a semântica de $\mathcal{ALC}$ deve ser compreendida como:

- Δ sendo o universo de discurso;
- As instancias de conceitos são elementos deste Δ ;
- Os conceitos são subconjuntos deste Δ ;
- As relações binárias são subconjuntos de $\Delta \times \Delta$;

A relação "subclasse" (subsunção) entre classes é interpretada como inclusão de conjuntos. A partir do domínio de interpretação, Δ^I , é formalizada a semântica:

Definição 10. (Interpretação e modelo): Uma interpretação é definida como o par $\langle \Delta^I, *^I \rangle$ em que uma função de interpretação atribui a cada conceito atômico A um conjunto $A^I \subseteq \Delta^I$ e a cada papel atômico r uma relação binária $r^I \subseteq \Delta^I \times \Delta^I$. Os conjuntos A^I e r^I são chamados extensões de A , r com respeito a I :

Tabela 2 – Semântica dos operadores como em Baader et al. (2010)

$\top^I$	$:=$	Δ^I
$\perp^I$	$:=$	$\emptyset$
$(\neg A^I)$	$:=$	$\Delta^I \setminus A^I$
$(C \sqcap D)$	$:=$	$C^I \cap D^I$
$(C \sqcup D)$	$:=$	$C^I \cup D^I$
$\forall r.C$	$:=$	$a \in \Delta^I \mid \forall b.(a, b) \in r^I \rightarrow b \in C^I$
$\exists r.C$	$:=$	$a \in \Delta^I \mid \exists b.(a, b) \in r^I, b \in C^I$

Fonte: Elaborada pelo autor (2022)

Uma Tbox admite axiomas de subsunção e equivalência, como mencionado na Definição 8, e uma Abox, asserções de conceitos e papéis. Uma interpretação I é um **modelo** da T-box T se satisfaz $A^I = D^I$ para todos os axiomas terminológicos $A \equiv D$ em T , e $A^I \subseteq D^I$ para todos os axiomas terminológicos $A \sqsubseteq D$ em T . Da mesma forma, uma interpretação I é modelo de um axioma assertivo $x : C$ se $x^I \in C^I$ e I é modelo de um axioma assertivo $(x, y) : r$ se $\langle x, y \rangle^I \in r^I$; I é modelo para uma Abox se for modelo para todo axioma desta Abox. Ela é, portanto, modelo de uma ontologia, se satisfaz ambas Tbox e Abox.

2.3 RACIOCÍNIO EM DL

O uso das DLs se dá através de ontologias. As linguagens são usadas para descrevê-las e permitir se gerar conhecimento a partir de tarefas de raciocínio (BAADER et al., 2010). As tarefas de raciocínio: classificação de conceitos e instâncias, equivalência, verificação de satisfatibilidade e consistência são descritas a seguir nas definições 11, 12, 14 e 15.

Definição 11. (Classificação (subsunção) de conceitos e instâncias): Seja T uma Tbox e A , B nomes de conceitos

$$A \sqsubseteq_T B \text{ sse } A^I \subseteq B^I \text{ para todos os modelos } I \text{ de } T.$$

Neste caso, todos os A estão em B , diz-se que B **classifica** A em T . Na possibilidade de haver mais de uma Tbox para a ontologia, a letra T em $\sqsubseteq_T$ identifica a qual Tbox se está referindo.

Exemplo 4. Considere as sentenças:

1. Motoristas dirigem veículos;
2. Motoristas de ônibus dirigem ônibus;
3. Um ônibus é um veículo;
4. Veículos precisam de licença.
5. Para serem motoristas, é necessário ter uma CNH

Tbox:

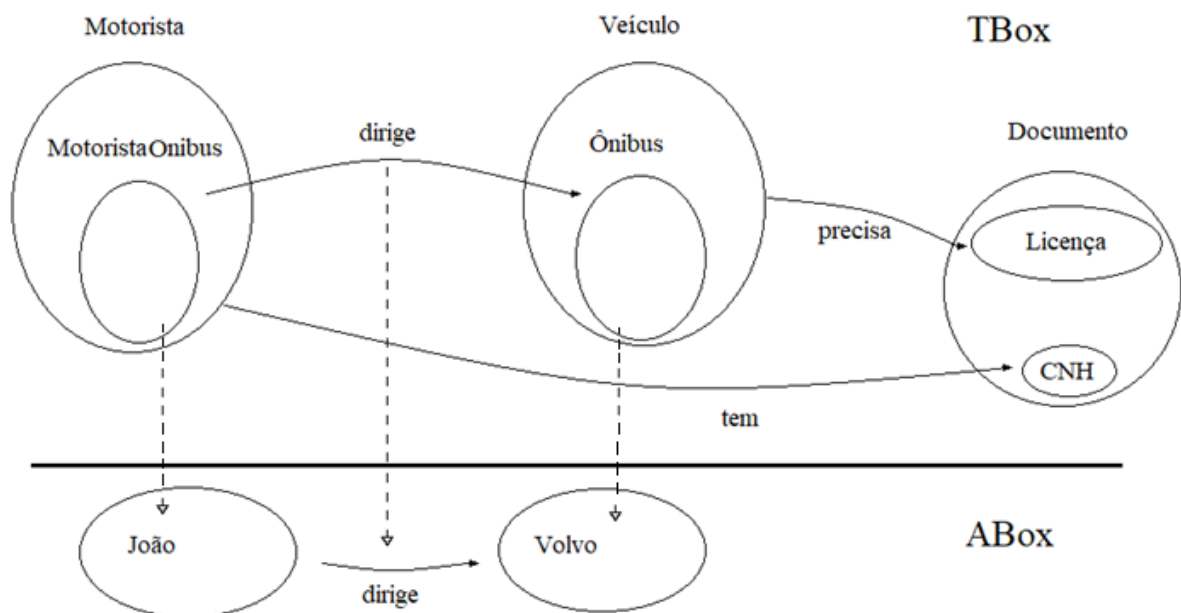
1. Motorista $\equiv \exists \text{dirige.Veículo}$
2. MotoristaOnibus $\equiv \exists \text{dirige.Ônibus}$
3. Ônibus $\sqsubseteq$ Veículo
4. Licença $\sqcap$ CNH $\sqsubseteq$ Documento
5. Veículo $\equiv \exists \text{precisa.Licença}$
6. Motorista $\equiv \forall \text{tem.CNH}$

Abox:

1. dirige(João, Ônibus)
2. MotoristaOnibus(João)
3. Ônibus(Volvo)

A partir da Tbox, a tarefa de classificação permite inferir que todos os motoristas de ônibus são motoristas, ou $\text{MotoristaOnibus} \sqsubseteq \text{Motorista}$. O axioma 6 da Tbox não teria a necessidade de existir, mas foi listado de propósito para melhor ilustrar um axioma com restrição universal. Neste exemplo, se houver o papel "tem", estará relacionado com uma instância de CNH.

Figura 5 – Representação gráfica da ontologia do Exemplo 4.



Fonte: Elaborada pelo autor (2022)

Definição 12. (Equivalência de conceitos): Dois conceitos C e D são equivalentes sse a interpretação de ambos for idêntica em todos os casos, ou seja, $C \equiv D \leftrightarrow C^I = D^I$. Em DL, isto significa que D classifica C e vice-versa, ou $C \sqsubseteq D \sqcap D \sqsubseteq C$.

Exemplo 5. Considere as sentenças:

1. Mulheres são pessoas fêmeas;
2. Homens são pessoas diferentes de mulheres;
3. Masculino são pessoas que não são fêmeas.

Em DL:

Mulher $\equiv$ Pessoa $\sqcap$ Fêmea

Homem $\equiv$ Pessoa $\sqcap \neg$ Mulher

Masculino $\equiv$ Pessoa $\sqcap \neg$ Fêmea

Para melhor entendimento, é adequado expandir a Tbox, substituindo os axiomas pelos conceitos equivalentes e simplificando as sentenças:

Homem $\equiv$ Pessoa $\sqcap \neg$ Mulher

Homem $\equiv$ Pessoa $\sqcap \neg$ (Pessoa $\sqcap$ Fêmea)

Homem $\equiv$ Pessoa $\sqcap (\neg$ Pessoa $\sqcup \neg$ Fêmea)

Homem $\equiv$ (Pessoa $\sqcap \neg$ Pessoa) $\sqcup$ (Pessoa $\sqcap \neg$ Fêmea)

Homem $\equiv$ Pessoa $\sqcap \neg$ Fêmea

A inferência a partir desta Tbox é a equivalência a seguir, em que é deduzido que Homens são do sexo masculino, em outras palavras, o axioma abaixo:

Homem $\equiv$ Masculino

Definição 13. (Verificação de Instanciação): Uma ontologia está instanciada se existe uma Abox que a instancia. Ou seja, todos os predicados unários (conceitos) contêm um nome de uma instância e todos os predicados binários (papéis) contêm dois nomes de instâncias.

Exemplo 6. Considere a base de conhecimento do Exemplo 4. O axioma Motorista(Joao) é deduzido por esta base.

Definição 14. (Satisfatibilidade): Uma ontologia satisfaz um conceito C se existe uma interpretação que o satisfaça. O conceito C é satisfatível nesta interpretação. O conceito C é insatisfatível se não existe interpretação que o satisfaça. Uma ontologia é satisfatível se admite um modelo, i.e., existe pelo menos uma interpretação que satisfaça ao mesmo tempo todos os conceitos e seus axiomas.

Exemplo 7. Sabe-se que comida vegana não conta com ingredientes de origem animal. Veja o exemplo da pizza a seguir:

$\text{PizzaVegana} \sqsubseteq$

$\text{Pizza} \sqcap \exists \text{temCobertura.Tomate} \sqcap \exists \text{temCobertura.Queijo} \sqcap \neg \exists \text{temCobertura.AlimentoOrigemAnimal};$

$\text{Queijo} \sqsubseteq \text{AlimentoOrigemAnimal};$

$\text{Tomate} \sqsubseteq \neg \text{AlimentoOrigemAnimal};$

$\models$

$\text{PizzaVegana} \sqsubseteq \perp$

Portanto, tanto o conceito PizzaVegana quanto a ontologia são insatisfatíveis.

Definição 15. (Consistência): Uma ontologia é inconsistente se apresenta instâncias de conceitos insatisfatíveis (equivalentes a um conjunto vazio, não podem ter nenhum indivíduo) ou seja, se a Abox ou a Tbox forem insatisfatíveis.

Exemplo 8. Neste exemplo, é possível notar uma instanciação que deixa a Abox inconsistente, porque o domínio de Pai não admite instâncias do sexo feminino. No Exemplo anterior (7) tem-se uma inconsistência na Tbox.

Tbox:

$\text{Mulher} \equiv \text{Pessoa} \sqcap \text{Fêmea};$

$\text{Homem} \equiv \text{Pessoa} \sqcap \neg \text{Mulher};$

$\text{Pai} \equiv \text{Homem} \sqcap \exists \text{temFilho.Pessoa};$

Abox:

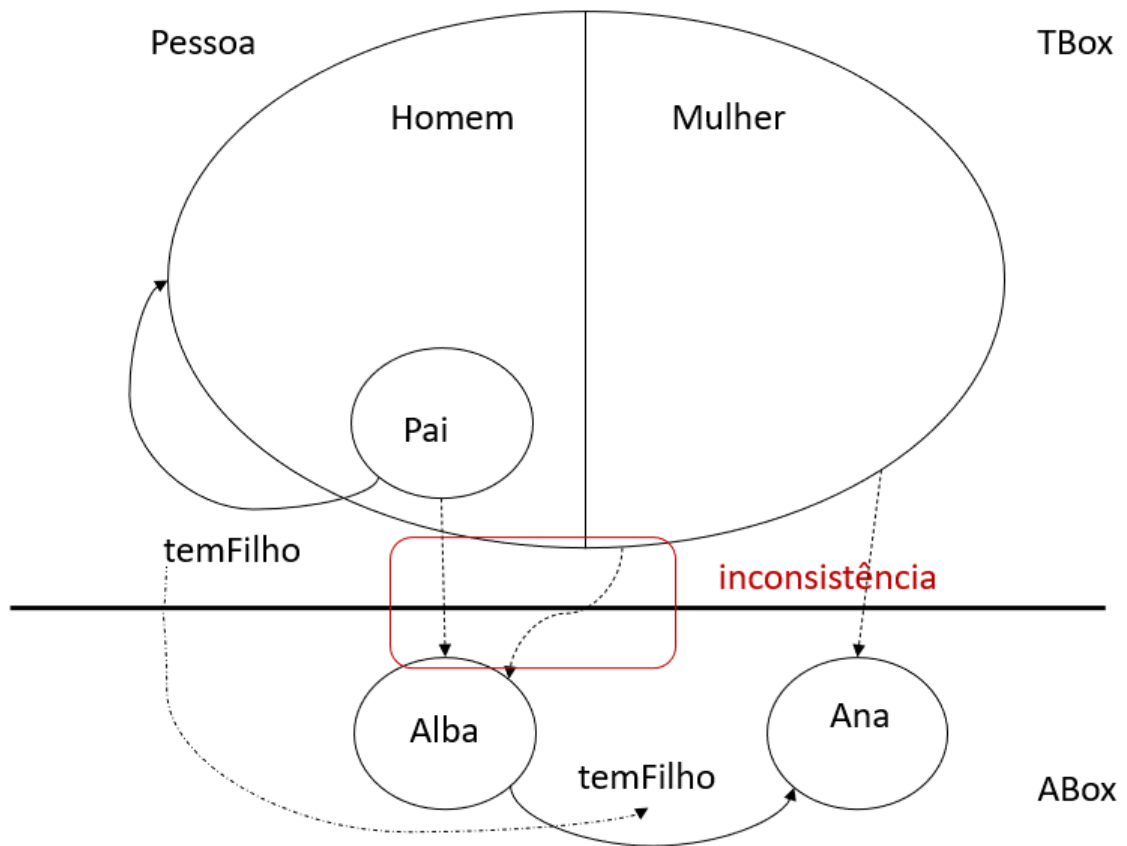
$\text{Mulher}(\text{Alba});$

$\text{Pai}(\text{Alba});$

$\text{temFilho}(\text{Alba}, \text{Ana});$

A ontologia pode ser representada graficamente como na Figura 6:

Figura 6 – Representação gráfica da ontologia do Exemplo 8



Fonte: Elaborada pelo autor (2022)

3 LÓGICA MODAL

A Lógica Modal é a lógica que estuda os modos de qualificar a verdade. Ela pertence à família de lógicas classificadas como não-clássicas ou não-padrão. Esta classe de lógicas difere da lógica padrão por conterem extensões linguísticas ou axiomáticas desta. Isto significa que, *inter alia*, os conectivos padrão, ou seja, a disjunção, conjunção e negação, são inteiramente preservados na Lógica Modal, o que quer dizer que não precisam de qualquer desvio quanto ao tratamento que recebem na lógica padrão clássica. A origem da ideia dos "modos de qualificar a verdade" é atribuída a Aristóteles³ (FELIX FILHO, 2019).

É comum encontrar na literatura o termo Lógica Modal Proposicional (do inglês, *Propositional Modal Logic*, PML) como intercambiável com o termo somente Lógica Modal quando a não abordar uma modalidade específica, mas a sintaxe e semântica comum a todas elas. As lógicas que apresentam mais de um construtor modal acabaram recebendo a alcunha de lógicas multimodais (BAADER, 2009), (CARNIELLI; PIZZI, 2008).

Na Lógica Modal, a semântica dos "mundos possíveis" de Leibniz, definida e detalhada mais adiante, explica como a noção de necessidade difere da simples verdade, φ : sejam s e t "mundos", em que s seja o mundo atual (sobre o qual se raciocina no momento), certa proposição φ em s pode ser verdade nos padrões clássicos, mas é necessariamente verdade somente se for verdade em todos os "mundos" t diferentes de s (despreocupadamente, pois não está sendo considerada ainda a relação de acessibilidade).

Esta nomenclatura se manteve como legado, mas, a depender do modelo que descrevem, estes "mundos possíveis" podem estar designando situações, estágios de informação ou processos, localizações num espaço, ou até simplesmente pontos num grafo direcionado.

Como discorre Marcus (1981), "A Lógica e semântica Modais preocupam-se fundamentalmente com as noções de necessidade, possibilidade e conceitos relacionados". No texto, ele afirma que a versão proposicional foi proposta por Lewis. Ela é a lógica clássica proposicional enriquecida com os operadores $\Box$ e $\Diamond$. À noção de mundos possíveis, está atrelada a noção de acessibilidade, de um "mundo ser acessível a partir de outro mundo". Em se tratando do trabalho de Lewis, a interpretação alternativa para os operadores modais é sobre inferência.

³ Esta preocupação não ficou na história. Na Lógica Modal contemporânea, à intuição de Aristóteles foi atribuída uma modalidade: a modalidade Alética. Raciocinar sob esta modalidade é se preocupar em como entender o motivo do valor verdade do objeto de estudo. Por exemplo, imagine que rochas sejam o objeto de estudo num domínio que engloba o planeta. É verdade que a maioria das rochas são invisíveis. Trata-se de um fato. Mas elas são invisíveis por serem transparentes (o que tornaria, na verdade, todas elas invisíveis), ou por serem opacas e, portanto, impedirem a observação de todas que estão numa camada inferior àquelas visíveis?

A definição dos conceitos de implicação material e implicação estrita foi dada por Lewis e Langford:

C. I. Lewis e C. H. Langford em *Symbolic Logic*, quando o desenvolvimento de um sistema modal de "implicação estrita" é usado para a interpretação da lógica do "se ... então".[...] o significado extensional, conhecido como "implicação material" de Russel (do inglês, Material Implication, ou MI), é a forma com que as fórmulas eram tratadas até então, enquanto que o significado intensional, ou a "implicação estrita" (do inglês, Strict Implication, ou SI), seria o entendimento de que o conseqüente deriva logicamente do antecedente.

Ou seja, sendo A e B fórmulas:

$$MI : A \supset B.$$

Na implicação material, aqui denotada por $\supset$, tem-se que A é falso, ou B é verdadeiro.

$$SI : A \prec B.$$

Na implicação estrita, aqui denotada por $\prec$, B é inferido a partir de A .

É válido trazer a este trabalho tais conceitos por serem comuns a ambas as lógicas, a Modal e a de Descrições. De fato, em DL, a implicação material é conhecida como conhecimento de instância (do inglês, *instance knowledge*) e a implicação estrita conhecida como conhecimento de esquema (do inglês, *schema knowledge*).

Resumidamente, a semântica modal pode ser entendida da seguinte forma:

1. *Mundos possíveis*. Um conjunto não vazio de pontos, que representam as situações pertinentes. Qualquer um deles pode ser o atual, sobre o qual se raciocina.
2. *Relação de acessibilidade*. Uma relação entre dois pontos indica que situações são alternativas a quais.
3. *Função de valoração*. Uma interpretação que estabelece quais enunciados são verdadeiros e quais falsos para cada situação.

Se ao leitor desperta a curiosidade de aprofundamento nas lógicas modais, o trabalho de Felix Filho (2019) traz muito acerca da história, com possível origem em Aristóteles (com a modalidade Alética), e Bull e Segerberg (2001) entrega um texto muito didático e mais detalhado das contribuições modernas até os anos 1970. Em sequência, o livro de Carnielli e Pizzi (2008) vai auxiliar na compreensão das modalidades, assunto discutido mais adiante. Aos leitores interessados em aprofundar esta introdução à Lógica Modal, recomenda-se o livro de, Cocchiarella

e Freund (2008). O *handbook* (BLACKBURN; BENTHEM; WOLTER, 2006) contém trabalhos que merecem ser conferidos, como Moss e Tiede (2007), que traz aplicações em linguística.

3.1 SINTAXE E SEMÂNTICA

Tendo em vista que a Lógica Modal aqui apresentada e discutida é Proposicional, as definições sintática e semântica a respeito da Lógica Proposicional Clássica serão propositalmente omitidas com o intuito da celeridade, sem que comprometa a compreensão. A seguir, serão apresentadas as definições dos dois operadores que caracterizam a Lógica como sendo Modal, o $\Box$, ou "necessário", e o $\Diamond$, ou "possível".

Definição 16. (Sintaxe dos operadores $\Box$ e $\Diamond$): Seja $\wp = \{p, q, r, \dots\}$ o conjunto de variáveis proposicionais; o conjunto de fórmulas bem-formadas sobre $\wp$ é definido como:

$$\varphi \stackrel{def}{=} p \mid \neg\varphi \mid \varphi \wedge \varphi' \mid \Box\varphi \mid \Diamond\varphi$$

Todos os conectivos restantes da lógica clássica podem ser obtidos através da álgebra relacional.

Definição 17. (Modelo de Kripke e semântica dos símbolos $\Box$ e $\Diamond$): Um modelo de Kripke é definido como uma terna ordenada:

$$\langle \mathcal{W}, \mathfrak{R}, \nu \rangle ,$$

em que $\mathcal{W}$ é um conjunto não-vazio, cujos membros são comumente referidos como mundos possíveis, $\mathfrak{R}$, uma relação binária entre os mundos possíveis ($\mathfrak{R} \subseteq \mathcal{W} \times \mathcal{W}$) chamada de relação de acessibilidade e ν , uma função binária, chamada de função de valoração (ou avaliação, do inglês, *valuation*), definida pela assinatura:

$$\nu : \mathcal{W} \times \wp \rightarrow \{\mathbf{0}, \mathbf{1}\}.$$

Semelhante à Definição 16, $\wp$ diz respeito ao conjunto de variáveis proposicionais. Os elementos da imagem de ν são as constantes "verdadeiro" e "falso". Esta notação foi admitida aqui pois concorda com a notação comumente usada nas lógicas a serem discutidas mais adiante.

Assim, assumindo que $\mathfrak{w}$ e $\mathfrak{w}'$ são elementos de $\mathcal{W}$, $\Box$ e $\Diamond$ são satisfatíveis quando:

$$(\text{ Satisfatibilidade } \Box) \quad \mathfrak{w} \models \Box p \text{ sse } \forall \mathfrak{w}' ((\mathfrak{w}' \in \mathcal{W}, (\mathfrak{w}, \mathfrak{w}') \in \mathfrak{R}) \rightarrow \mathfrak{w}' \models p);$$

Ou seja, lendo a fórmula, sendo p uma proposição, $\mathfrak{w}' \models p$ quando $\Box p$ é verdade num mundo possível $\mathfrak{w}$ sse p é verdade em todo mundo possível que é acessível a partir de $\mathfrak{w}$. Isto significa que todos os mundos possíveis acessíveis a partir do atual, do mundo em questão analisado, devem deduzir (valor verdade **1**) a fórmula sob a qual opera o operador $\Box$. Além do mais, que pode não haver dupla que satisfaça a relação de acessibilidade (assim como permite a implicação nesta definição $(\mathfrak{w}' \in \mathcal{W}, (\mathfrak{w}, \mathfrak{w}') \in \mathfrak{R}) \rightarrow \mathfrak{w}' \models p$), mas se houver, todos devem deduzi-la. Uma interpretação alternativa seria: ou não há acessibilidade de $\mathfrak{w}$ para $\mathfrak{w}'$, ou $\mathfrak{w}'$ deduz p . Na prática, mesmo sendo uma interpretação logicamente válida, ela pode não servir dependendo da aplicação, pois pode não ser prático que não haja relação de acessibilidade entre os mundos analisados.

$$(\text{ Satisfatibilidade } \Diamond) \quad \mathfrak{w} \models \Diamond p \text{ sse } \exists \mathfrak{w}' ((\mathfrak{w}' \in \mathcal{W}, (\mathfrak{w}, \mathfrak{w}') \in \mathfrak{R}) \wedge \mathfrak{w}' \models p).$$

Portanto, $\mathfrak{w}'$ deduz p quando $\Diamond p$ é verdade num mundo possível $\mathfrak{w}$ sse p é verdade em algum mundo possível que é acessível a partir de $\mathfrak{w}$. Esta definição indica que vai haver tal mundo que pertença à relação e que deduza p , diferentemente da possibilidade de não haver da definição do operador de necessidade modal.

Seja $\mathcal{A} = \langle \mathcal{W}, \mathfrak{R}, v \rangle$ um modelo de Kripke: diz-se que $\mathcal{A}$ satisfaz p se existe algum mundo $\mathfrak{w} \in \mathfrak{M}$ tal que $\mathfrak{w} \models p$. Diz-se que p é satisfatível se existe algum modelo que o satisfaça. Se p é satisfatível em $\mathcal{A}$ para todo mundo $\mathfrak{w} \in \mathfrak{M}$, $\mathfrak{w} \models p$, então $\mathcal{A}$ é um modelo para p e p é válido em $\mathcal{A}$. Se $\mathcal{A} \models p$ para todo modelo de Kripke, então p é válido⁴. É simples entender validade e satisfatibilidade enquanto duais como os quantificadores universais: uma proposição é válida se e somente se sua negação não for satisfatível.

⁴ Vale lembrar que os conceitos de corretude e completude fazem uso da definição de validade. Ou seja, um sistema dedutivo $\vdash$ é correto em relação à semântica $\models$ se todo teorema for válido e, por sua vez, se toda sentença válida é um teorema, $\vdash$ é completo em relação a $\models$.

Definição 18. (Fórmulas Equivalentes):

1. $\Box p \equiv \neg \Diamond \neg p$
2. $\Diamond p \equiv \neg \Box \neg p$
3. $\Box \neg p \equiv \neg \Diamond p$
4. $\neg \Box p \equiv \Diamond \neg p$

Nota-se que as equivalências 1 e 2 são semelhantes às clássicas da Lógica de Primeira Ordem para os quantificadores universais. No contexto modal, a equivalência 1 deve ser entendida como: sendo p necessário, não é possível que haja a negação de p , que seu contrário também seja verdade (e vice-versa). Remetendo à Definição 16 é possível constatar a veracidade desta leitura: há a possibilidade de que no modelo não exista uma relação de acessibilidade entre dois mundos e, mesmo assim, uma fórmula $\Box p$ ser verdade; entretanto, é impossível que, sendo $\neg p$ verdade, p necessariamente também seja. Além destas equivalências encaradas como clássicas, existem as equivalências 3 e 4. Considerando a 3, segue da discussão anterior: se necessariamente $\neg p$, então não é possível que p seja verdade e vice-versa. Da mesma forma, 4 lista a condição de que: não sendo necessário p , é possível que $\neg p$ seja verdade.

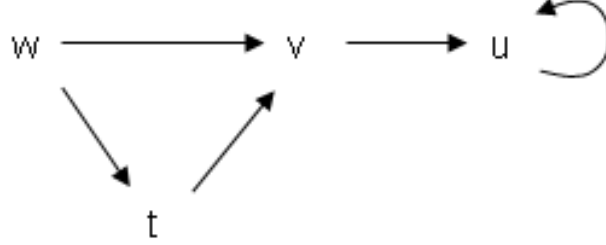
Dedução modal não passa de raciocínio com quantificadores numa notação perspicua livre de variáveis. Com efeito, a Lógica Modal em sua forma mais básica é uma miniatura decidível da LPO, como esperado, em se tratando de uma equivalente das Lógicas de Descrições. Há vários métodos de decisão para validade ou satisfatibilidade que exploram características especiais das fórmulas modais, como seleção, redução e filtragem (MARX, 2007). Neste trabalho, o método da filtragem serve ao propósito e é detalhado tanto neste capítulo quanto no reservado à Lógica Dinâmica.

Comumente, é preferível trabalhar com uma dupla $\langle \mathcal{W}, \mathfrak{R} \rangle$ em vez da terna $\langle \mathcal{W}, \mathfrak{R}, v \rangle$. Sem a função de valoração, a dupla recebe o nome de *frame* de Kripke. É comum encontrar definições para Kripke *frame* na literatura. A utilização de um Kripke *frame* é aconselhável quando nas definições semânticas (HAREL; KOZEN; TIURYN, 2000):

Definição 19. (Kripke *frame*): Um Kripke *frame*, também chamado de estrutura de Kripke, é uma dupla $F = \langle \mathcal{W}, \mathfrak{R} \rangle$ tal que $\mathcal{W}$ e $\mathfrak{R}$ são os mesmos da Definição 17.

Exemplo 9. Seja um *frame* tal que $\mathcal{W} = \{t, u, v, w\}$ e $\mathfrak{R} = \{(w, v), (w, t), (t, v), (v, u) \text{ e } (u, u)\}$

Figura 7 – Visualização do *frame* do Exemplo 9.



Fonte: Elaborada pelo autor (2022)

Definição 20. (Semântica dos Operadores nos Kripke *frames*): Dados $\wp$, um conjunto de proposições, um Kripke *frame* $F = \langle \mathcal{W}, \mathfrak{R} \rangle$ tal que $\mathcal{W}$ é o conjunto não-vazio de mundos possíveis, $\mathfrak{R}$ a relação de acessibilidade e ν_F é a função de valoração que atribui um subconjunto de $\mathcal{W}$ a cada proposição atômica, define-se:

1. $\nu_F(p \vee q) \stackrel{def}{=} \nu_F(p) \cup \nu_F(q)$
2. $\nu_F(p \wedge q) \stackrel{def}{=} \nu_F(p) \cap \nu_F(q)$
3. $\nu_F(\neg p) \stackrel{def}{=} \mathcal{W} \setminus \nu_F(p)$
4. $\nu_F(p \rightarrow q) \stackrel{def}{=} (\mathcal{W} \setminus \nu_F(p)) \cup \nu_F(q)$
5. $\nu_F(\Diamond p) \stackrel{def}{=} \{u \mid \exists v \in \mathcal{W}, (u, v) \in \mathfrak{R} \text{ e } v \in \nu_F(p)\}$
 $= \nu_F(\mathfrak{R}) \circ \nu_F(p)$
6. $\nu_F(\Box p) \stackrel{def}{=} \mathcal{W} \setminus (\nu_F(\mathfrak{R}) \circ (\mathcal{W} \setminus \nu_F(p)))$
 $= \{u \mid \forall v \in \mathcal{W}, \text{ se } (u, v) \in \mathfrak{R}, \text{ então } v \in \nu_F(p)\}$
7. $\nu_F(\mathbf{1}) \stackrel{def}{=} \mathcal{W}$
8. $\nu_F(\mathbf{0}) \stackrel{def}{=} \emptyset$

É importante que, ao se deparar com seções de grande carga notacional, os leitores não sejam repelidos, ou relutem a entender: a Definição 20 está compilando tudo o que foi discutido anteriormente. Aqui, para que a função avalie em verdadeiro ou falso (tal qual sua definição em 17), os elementos avaliados precisam pertencer ao domínio. É a isto que a notação conjuntista do lado direito quer remeter. Os itens de 1 a 4 dizem respeito ao comportamento da função de valoração com expressões da Lógica Proposicional Clássica. Por exemplo, a avaliação de uma disjunção entre p e q é definida como a avaliação da união dos conjuntos que contêm ambos. O item 3 mostra que o valor verdade obtido da avaliação da negação de um p é oriundo de um *frame*

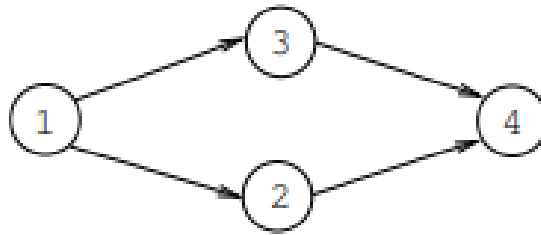
que tem excluído este elemento. Os itens 5 e 6 seguem da definição dos operadores modais: para avaliar a expressão "é possível p " assim como em 5, como a proposição p é avaliada no mundo v ($v \in \nu_F(p)$) que já é elemento de $\mathcal{W}$, deve haver um mundo diferente de v que o acesse. Esta sentença inteira é o mesmo que uma possível composição da "avaliação" do conjunto $\mathfrak{R}$ com o conjunto que contém p . As constantes dos itens 7 e 8 concluem o estudo: será tautologia uma fórmula satisfatível em qualquer mundo do conjunto $\mathfrak{R}$. Da mesma forma, o conjunto vazio quer dizer que não há mundo que a satisfaça.

Exemplo 10. Seja o *frame* de Kripke $F = \langle \mathcal{W}, \mathfrak{R} \rangle$ dado por:

$$F = (\{1, 2, 3, 4\}, \{(1, 2), (1, 3), (2, 4), (3, 4)\}).$$

É possível construir um grafo a partir desse modelo como o a seguir:

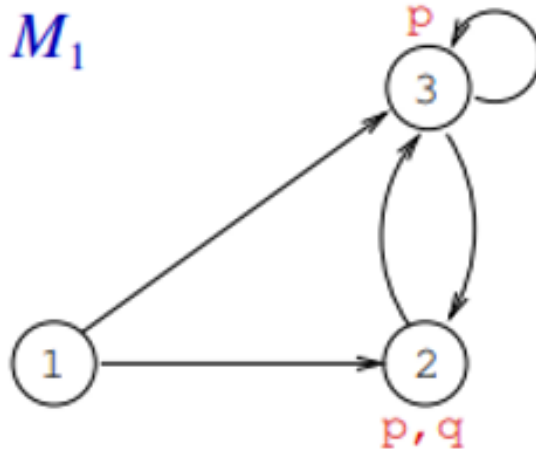
Figura 8 – Representação gráfica do frame do Exemplo 10.



Fonte: Elaborada pelo autor (2022)

Exemplo 11. Sendo os mundos possíveis do modelo $M_1 = 1, 2, 3$, quais fórmulas são válidas a partir de 1, ou seja $1 \models \varphi$?

Figura 9 – Representação gráfica do modelo do Exemplo 11.



Fonte: Elaborada pelo autor (2022)

É possível construir fórmulas como as que seguem: $1 \models \Diamond p$, $1 \models \Box p$, $1 \models \Diamond q$, $1 \models \Box p \wedge \Diamond q$

Em contrapartida, fórmulas como $\Box q$ e $\Box q \vee \Diamond q$ não podem ser deduzidas em 1, assim como em 2 não se deduz $\Box q$, pois a fórmula q é deduzida somente em 2, não em 3, como pede a definição, bem como em 3 ela também não é deduzida: existe uma indicação de reflexividade neste mundo possível.

Exemplo 12. Considere o modelo de Kripke $\langle \mathcal{W}, \mathfrak{R}, \nu \rangle$ tal que:

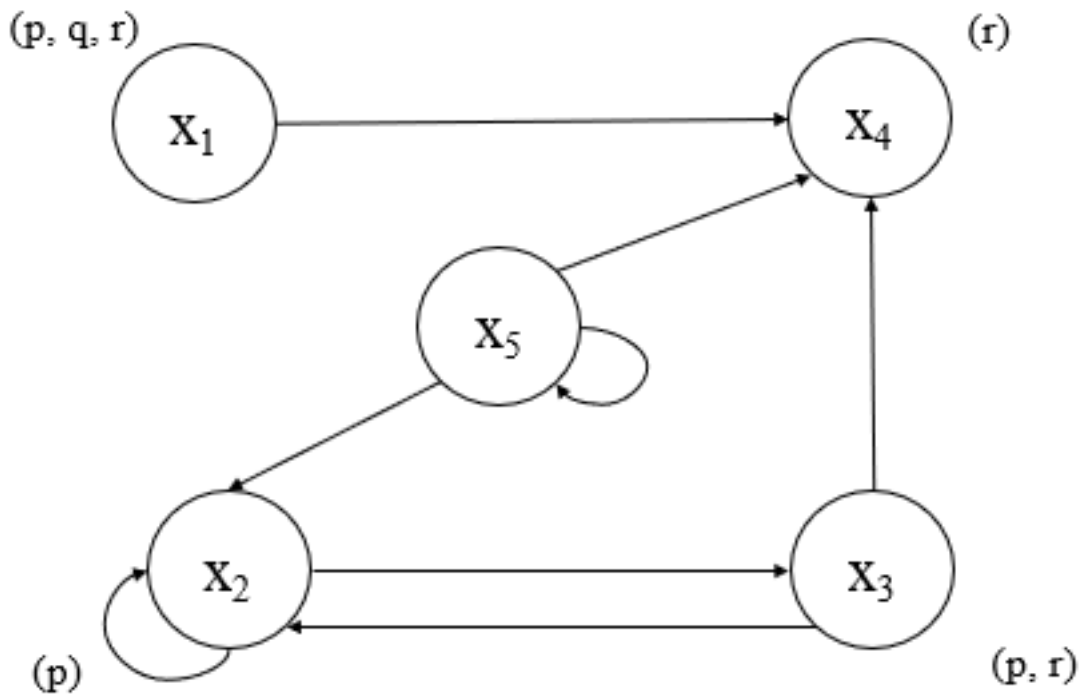
$$\mathcal{W} = \{x_1, x_2, x_3, x_4, x_5\};$$

$$\mathfrak{R} = \{(x_1, x_2), (x_1, x_4), (x_2, x_2), (x_2, x_3), (x_3, x_2), (x_3, x_4), (x_5, x_5), (x_5, x_2), (x_5, x_4)\};$$

$$\nu_i = \begin{cases} (p, q, r) & i = 1; \\ (p) & i = 2; \\ (p, r) & i = 3; \\ (r) & i = 4; \\ \emptyset & i = 5; \end{cases}$$

de modo que pode ser representado graficamente como na Figura 10. É possível conferir que a fórmula $\Diamond(r \wedge \Box q)$ pode ser deduzida no mundo x_1 , ou $x_1 \models \Diamond(r \wedge \Box q)$.

Figura 10 – Modelo de Kripke do Exemplo 12



Fonte: Elaborada pelo autor (2022)

Adicionalmente, define-se fórmulas válidas para a Lógica Modal Proposicional (HAREL; KOZEN; TIURYN, 2000, p. 129). Elas podem ser constatadas também em (CARNIELLI; PIZZI, 2008, p. 39) como "K-teoremas". A prova deste Teorema 1 pode ser conferida no apêndice C.

Teorema 1. (*Fórmulas Válidas*): São fórmulas válidas da PML:

1. $\Diamond(p \vee q) \equiv \Diamond p \vee \Diamond q$
2. $\Box(p \wedge q) \equiv \Box p \wedge \Box q$
3. $\Diamond p \wedge \Box q \rightarrow \Diamond(p \wedge q)$
4. $\Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q)$
5. $\Diamond(p \wedge q) \rightarrow \Diamond p \wedge \Diamond q$
6. $\Box p \vee \Box q \rightarrow \Box(p \vee q)$
7. $\Diamond \mathbf{0} \equiv \mathbf{0}$

4 CORRESPONDÊNCIA DEMONSTRADA

Neste capítulo, serão exibidas as funções que mapeiam as sintaxes, bem como as correspondências semânticas e de raciocínio automático entre as lógicas Modal e de Descrições. Assim, como motivação para a primeira seção, sem muito se ater a uma definição mais formalizada, repare no mapeamento 4.1 entre DL $\mathcal{ALC}$ e o sistema $\mathbf{K}$:

$$\pi(\exists r.C) = \Diamond(\pi(C)) \quad (4.1)$$

Tal mapeamento se verifica quando houver apenas um nome de papel na ontologia (r for único) em $\mathcal{ALC}$ sendo traduzida. Este papel se tornará a relação de acessibilidade em PML. O Exemplo 8 conta com apenas um papel na TBox. Se nela fosse adicionado um papel qualquer, talvez temConjuje, a tradução ficaria impossível de ser estabelecida.

Desta forma, fica evidente que urge um enriquecimento da PML para que acompanhe a natureza não unitária do conjunto de papéis de $\mathcal{ALC}$. Para tanto, a seção seguinte trabalha justamente a multimodalização do sistema $\mathbf{K}$: a versão multimodal $\mathbf{K}_m$ que possui mais de uma relação de acessibilidade. Assim, cada nome de papel de $\mathcal{ALC}$ pode ser mapeado individualmente para a relação de acessibilidade em $\mathbf{K}_m$ que o corresponda.

Em sequência, nas duas seções 4.2.2 e 4.3, as correspondências propriamente ditas serão exibidas: respectivamente, a correspondência limitada à Tbox e, em seguida, a correspondência com respeito à Abox, com a apresentação de uma linguagem formal específica chamada Lógica Híbrida. Esta lógica é apresentada como enriquecimento da PML, que contempla a natureza das instâncias de $\mathcal{ALC}$. Por fim, uma seção curta lista o mapeamento entre os dois formalismos e é seguida pela seção que demonstra as correspondências em termos de raciocínio automático através do método de *tableaux*, fechando assim o capítulo.

4.1 ENRIQUECIMENTO DO SISTEMA $\mathbf{K}$: A MULTIMODALIZAÇÃO $\mathbf{K}_M$

Uma lógica multimodal $\mathbf{K}_m$ é tal que a componente "relação de acessibilidade" da tripla $\langle \mathcal{W}, \mathfrak{R}, \nu \rangle$ seja entendida como um conjunto contendo as relações propriamente ditas (possivelmente $\mathfrak{R}_i$, sendo i cada um dos diferentes objetos em $\mathfrak{R}$), sendo assim quantificadas conforme sua nomenclatura⁵. Conforme (HAREL; KOZEN; TIURYN, 2000, p. 130), a Definição

⁵ É adotada a simbologia com índice em algumas nomenclaturas encontradas na literatura, usando os operadores chamados multimodais $\Box_i$ e $\Diamond_i$. Aqui será adotada a sintaxe que corresponde à Lógica Dinâmica

21 generaliza o sistema **K** para sua versão multimodal. O que o autor chama no texto de "modalidades", entenda-se conforme explicado anteriormente acerca da relação de acessibilidade.

Logo a seguir, serão definidas a sintaxe e semântica desta versão. A sintaxe é idêntica à do sistema **K**, sendo o operador de necessidade $\Box$ substituído pelo $[]$ e o $\Diamond$ (de possibilidade) pelo $\langle \rangle$. Já na semântica, o modelo de Kripke do sistema **K** $\langle \mathcal{W}, \mathfrak{R}, v \rangle$ tem a relação $\mathfrak{R}$ substituído por um conjunto A de relações (ou transições). Similarmente, funcionando apenas como uma alternativa sintática, sem a necessidade de definir um conjunto A , os operadores $\Box$ e $\Diamond$ podem ganhar índices ($\Box_i$ e $\Diamond_i$), indicando a propriedade multimodal do sistema.

Definição 21. (Sintaxe de $\mathbf{K}_m$): Sejam $\wp = \{p, q, r, \dots\}$ o conjunto de variáveis proposicionais e $A = \{a, b, c, \dots\}$ um conjunto de modalidades. As fórmulas bem-formadas do sistema são definidas como:

$$\wp \stackrel{def}{=} p \mid \neg \wp \mid \wp \wedge \wp' \mid [a]\wp$$

A fórmula da forma $\langle a \rangle \wp$ é definida como sendo obtida a partir do primitivo $[]$: $\langle a \rangle \wp = \neg[a]\neg\wp$.

A semântica herda todas as propriedades da Definição 17:

Definição 22. (Semântica de $\mathbf{K}_m$): O modelo de Kripke do sistema **K** $\langle \mathcal{W}, \mathfrak{R}, v \rangle$ tem a relação $\mathfrak{R}$ substituída pelo conjunto $A = a, b, c, \dots$. A satisfatibilidade da necessidade no sistema multimodal **K** é definida como:

$$(\text{Satisfatibilidade do } []) \quad \mathfrak{w} \models [a]p \text{ sse } \forall \mathfrak{w}' ((\mathfrak{w}' \in \mathcal{W}, (\mathfrak{w}, \mathfrak{w}') \in a) \rightarrow \mathfrak{w}' \models p);$$

Ou, simplesmente, se para todos os mundos $\mathfrak{w}'$ há uma transição chamada a saindo de $\mathfrak{w}$, $[a]p$ é satisfatível em $\mathfrak{w}$. Equivalentemente obtêm-se para a possibilidade na fórmula $\langle a \rangle \wp$.

Da mesma forma, as equivalências da Definição 18 têm a sintaxe a seguir:

1. $[a]\wp \equiv \neg \langle a \rangle \neg \wp$
2. $\langle a \rangle \wp \equiv \neg [a] \neg \wp$
3. $[a] \neg \wp \equiv \neg \langle a \rangle \wp$
4. $\neg [a] \wp \equiv \langle a \rangle \neg \wp$

No capítulo reservado à Lógica Dinâmica, esta sintaxe será bem mais trabalhada, pois define uma linguagem formal consolidada com aplicações bem definidas. Através dos exemplos ilustrativos, os conceitos apresentados serão melhor compreendidos.

4.2 CORRESPONDÊNCIA LIMITADA À TBOX

Em relação à parte terminológica, $\mathcal{ALC}$ corresponderia ao sistema $\mathbf{K}$, mapeando-se cada papel como uma relação de acessibilidade. Porém, $\mathbf{K}$ conta somente com uma relação de acessibilidade, que é única. Logo, havendo mais de um papel, $\mathbf{K}$ não alcança o mapeamento completo, provocando, então, a necessidade de um enriquecimento para este sistema. Será mostrado adiante (4.2.1) como é alcançado o sistema multimodal $\mathbf{K}_m$, que consegue corresponder à pluralidade de nomes de papéis de $\mathcal{ALC}$.

A seguir, a versão adaptada do exemplo de Ohlbach e Koehler (1999, p. 2) para $\mathcal{ALC}$ é ilustrada.

4.2.1 Correspondência com $\mathbf{K}$

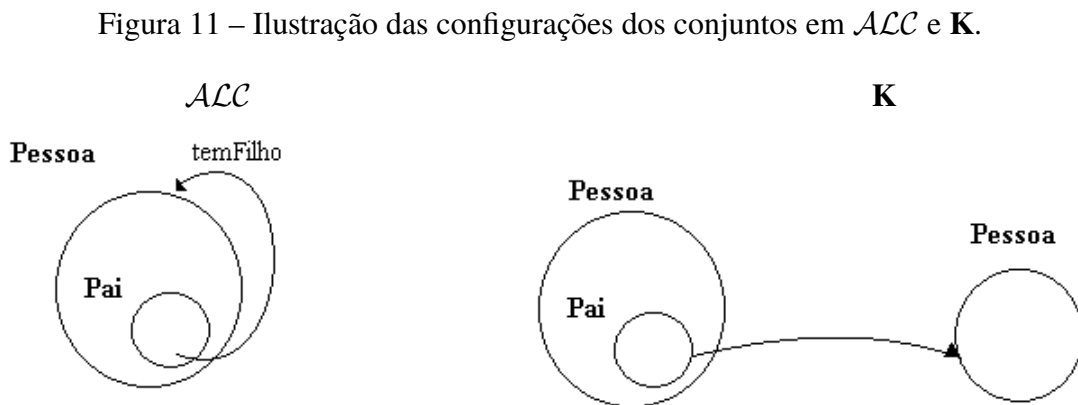
Como mencionado anteriormente, repare no Exemplo 13:

Exemplo 13.

Em DL: $Pai \equiv Pessoa \sqcap \exists temFilho.Pessoa$

Em PML: $Pai \leftrightarrow (Pessoa \wedge \Diamond Pessoa)$

Repare na figura abaixo:



Fonte: Elaborada pelo autor (2022)

O papel *temFilho* está impícito sob o operador modal $\Diamond$, denotando a relação (única) de acessibilidade.

Considere o Exemplo 3. Ele apresenta somente um papel. Há a possibilidade de traduzir sem ônus. Considere o seguinte axioma:

$$Mula \equiv Animal \sqcap \exists temPai.Cavalo$$

tal que Mula = M, Animal = A e Cavalo = C. Pela sintática modal, a sentença corresponde a esta:

$$M \leftrightarrow A \wedge \Diamond C$$

4.2.2 Correspondência com $\mathbf{K}_m$

Nesta seção, repare o Exemplo 4. Tantos são os nomes de papéis, que torna-se imprescindível o sistema $\mathbf{K}$ em sua forma multimodal para que aconteça a correspondência adequadamente. Logo em seguida (14), confira o mapeamento π entre as linguagens.

Exemplo 14. Considere a Tbox do Exemplo 4.

Tbox

1. $Motorista \equiv \exists dirige.Veículo$
2. $MotoristaOnibus \equiv \exists dirige.Ônibus$
3. $Ônibus \sqsubseteq Veículo$
4. $Licença \sqcap CNH \sqsubseteq Documento$
5. $Veículo \equiv \exists precisa.Licença$
6. $Motorista \equiv \forall tem.CNH$

Considere agora $A = \{a, b, c, \dots\}$ conforme a Definição 21 como sendo o conjunto das transições. Os papéis desta Tbox têm sua tradução apresentada a seguir, assim como os conceitos serão em $\mathbf{K}_m$ representados pelas iniciais maiúsculas do seu nome:

CONCEITOS	PAPÉIS
Motorista = M	dirige = d
Veículo = V	precisa = p
MotoristaOnibus = MO	tem = t
Ônibus = O	
Licença = L	
CNH = C	
Documento = D	

Portanto, as sentenças enumeradas sob a sintaxe modal têm a forma como apresentadas a seguir:

1. $M \leftrightarrow \langle d \rangle V$
2. $MO \leftrightarrow \langle d \rangle O$
3. $O \rightarrow V$
4. $L \wedge C \rightarrow D$
5. $V \leftrightarrow \langle p \rangle L$
6. $M \leftrightarrow [t] C$

A correspondência Lógica Modal $\times$ Lógica de Descrições foi descrita em (BAADER et al., 2007) como uma tradução π de conceitos $\mathcal{ALC}$ usando nomes de papéis quantificados r_i para $\mathbf{K}_m$:

Definição 23. (Correspondência na Tbox):

$$\begin{aligned}
 \pi(A) &= A \\
 \pi(C \sqcap D) &= \pi(C) \wedge \pi(D) \\
 \pi(C \sqcup D) &= \pi(C) \vee \pi(D) \\
 \pi(\neg(C)) &= \neg(\pi(C)) \\
 \pi(\exists r_i.C) &= \Diamond_i(\pi(C)) \\
 \pi(\forall r_i.C) &= \Box_i(\pi(C))
 \end{aligned}$$

Ou simplesmente, sem tanta carga notacional, como em (BAADER; HORROCKS; SATTler, 2008):

$$\begin{aligned}
A &= a \\
C \sqcap D &= C \wedge D \\
C \sqcup D &= C \vee D \\
\neg C &= \neg C \\
\exists r.C &= \langle r \rangle C \\
\forall r.C &= [r]C
\end{aligned}$$

Ou seja, a é uma instância de um conceito C de $\mathcal{ALC}$ numa interpretação I , sse sua tradução $\pi(C)$ é verdade num a de uma estrutura de Kripke correspondente à interpretação I . Como auxílio visual, a correspondência é representada como:

nome de conceito A	=	variável proposicional a
nome de papel r	=	parâmetro modal r
interpretação I	=	frame de Kripke $K = \langle \mathcal{W}, \mathfrak{R} \rangle$
conjunto de instâncias $dom(I)$	=	conjunto de mundos possíveis $\mathcal{W}$
interpretação de papéis r^I	=	relação de acessibilidade $\mathfrak{R}_r$
interpretação de conceitos A^I	=	mundos em que a é verdade

Como a representação em DL permite asserções (Abox), é fundamental que seja demonstrada também a correspondência com ela. Perceba que, no Exemplo 8, a correspondência, mesmo somente para **K**, ela se daria apenas na TBox, não havendo como estender para a ABox sem a prévia definição de uma estrutura adequada. Para tal, será apresentada na seção seguinte uma lógica chamada **híbrida**, conforme o trabalho de Areces, Blackburn e Marx (1999), que conta com o operador @ como primitivo, assim como o $\Box$, que contempla completamente a noção de uma instanciação em DL.

4.3 CORRESPONDÊNCIA PARA ABOX: A LÓGICA HÍBRIDA DO OPERADOR @

Diferente da Tbox, a Abox não tem correspondência direta na Lógica Modal tradicional. O enriquecimento híbrido para as Lógicas Modais datam dos anos 60 tardios, com o trabalho de Prior, como cita Blackburn (2006). Somente na Lógica Modal Híbrida, que contém um

construtor modal chamado **nominal**⁶. Os nominais são variáveis proposicionais especiais, com a característica de ser verificável em exatamente um mundo único.

Formalmente, a Lógica Híbrida é idêntica à PML, tanto em sintaxe, quanto em semântica, adicionada de um operador modal chamado @. Este operador serve para designar um conjunto unitário, o único mundo em que o nominal se verifica. É direto, portanto, perceber que, sendo uma instância um objeto pertencente a um conceito, sua correspondente nesta versão híbrida da Lógica Modal se trata do nominal atrelado àquele mundo possível unitário, o único que o torna verificável. A Definição 24 segue da Definição 16 e respeita a definição da sintaxe híbrida do trabalho de Areces, Blackburn e Marx (1999).

Definição 24. (Sintaxe da Lógica Híbrida): Sejam $\wp = \{p, q, r, \dots\}$ o conjunto de variáveis proposicionais e $\mathcal{N} = \{i, j, k, \dots\}$ o conjunto de nominais disjunto de $\wp$; O conjunto ATOM é definido como $\text{ATOM} = \wp \cup \mathcal{N}$ e denota o conjunto de átomos. O conjunto de fórmulas bem-formadas sobre ATOM é definido como:

$$\varphi \stackrel{\text{def}}{=} a \mid \neg\varphi \mid \varphi \wedge \varphi' \mid \Box\varphi \mid @_i\varphi$$

O elemento a pertence a ATOM. Fórmulas que não contêm símbolos de $\wp$ são chamadas puras.

Da mesma forma, sendo idêntica, exceto pelo operador @, a semântica segue da Definição 17:

Definição 25. (Semântica da Lógica Híbrida): Seja um modelo definido por $\langle \mathcal{W}, \mathfrak{R}, \nu \rangle$, em que $\mathcal{W}$ é um conjunto não-vazio de mundos, aqui referidos como estados, $\mathfrak{R}$, uma relação binária de acessibilidade entre os estados ($\mathfrak{R} \subseteq \mathcal{W} \times \mathcal{W}$) chamada de relação de transição e ν a função de valoração de assinatura $\nu : \text{ATOM} \rightarrow \mathcal{W}$ tal que para todo $i \in \mathcal{N}$, $\nu(i)$ é um subconjunto unitário de $\mathcal{W}$. A relação de satisfatibilidade para @ é definida como:

Seja M um modelo e $(m, n) \in \mathcal{W}$:

$$\begin{aligned} M, m \models a & \quad \text{sse} \quad m \in \nu(a), a \in \text{ATOM}; \\ M, m \models @_i\varphi & \quad \text{sse} \quad M, n \models \varphi, \nu(i) = n, i \in \mathcal{N}. \end{aligned}$$

Toda esta notação quer definir algo simples: como as funções de avaliação atribuem conjuntos unitários aos nominais, então cada nominal é satisfatível em exatamente um estado em qualquer modelo. Ou seja, para avaliar $@_i\varphi$, avalie φ no único estado denominado por i .

⁶ Há a necessidade em DLs em alguns casos para os nominais: Brasileiro $\equiv$ nasceu.{Brasil}, tal que este axioma da TBox demanda a presença da instância "Brasil". Mais sobre eles poderá ser conferido no apêndice D.

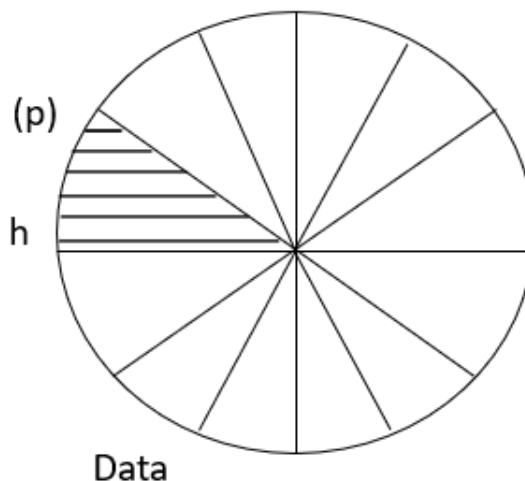
O Exemplo 15 aplica um conceito de uma situação do mundo real em Lógica Híbrida.

Exemplo 15. Imagine que seja preciso raciocinar sobre o fato de estar chovendo. Diretamente, nota-se que é um fato dependente de uma condição variante no tempo: será verdade em momentos específicos. Note também que a frase "são 10h em 23 de fevereiro de 2022" tem uma característica semelhante: será verdade somente neste momento específico. Agora que a intuição de nominais foi despertada, torna-se simples organizar estas frases numa sentença da Lógica Híbrida: $@_h p$ tal que o nominal se refere àquela hora do dia, daí a escolha do índice "h", e a proposição denota o fato de "estar chovendo". Portanto, "às 10h em 23 de fevereiro de 2022, está chovendo." é a fórmula $@_h p$, tal que:

$@_h =$ às 10h em 23 de fevereiro de 2022;

$p =$ está chovendo.

Figura 12 – Ilustração do Exemplo 15 em que "Data" resume a data por extenso, "h" representa a hora de tal data com a seção hachurada e "(p)" representa a proposição analisada em tal nominal, que é verdade apenas nele.



Fonte: Elaborada pelo autor (2022)

Por fim, é válido estender a Definição 23 para uma que contemple também a Abox (ARECES; RIJKE, 2000).

Definição 26. (Correspondência na Abox): Sejam as instâncias como na Definição 7 e a um elemento de ATOM como na Definição 24, a função π do mapeamento conforme a Definição 23 opera na Abox como definido abaixo:

$$\begin{aligned}\pi(a : C) &= @_a \pi(C) \\ \pi((a, b) : r) &= @_a \langle r \rangle . b\end{aligned}$$

Agora, repare nesta modificação do exemplo 4, em que a Abox será reescrita em termos da Lógica Híbrida.

Exemplo 16. A Abox do Exemplo 4 é composta das seguintes sentenças:

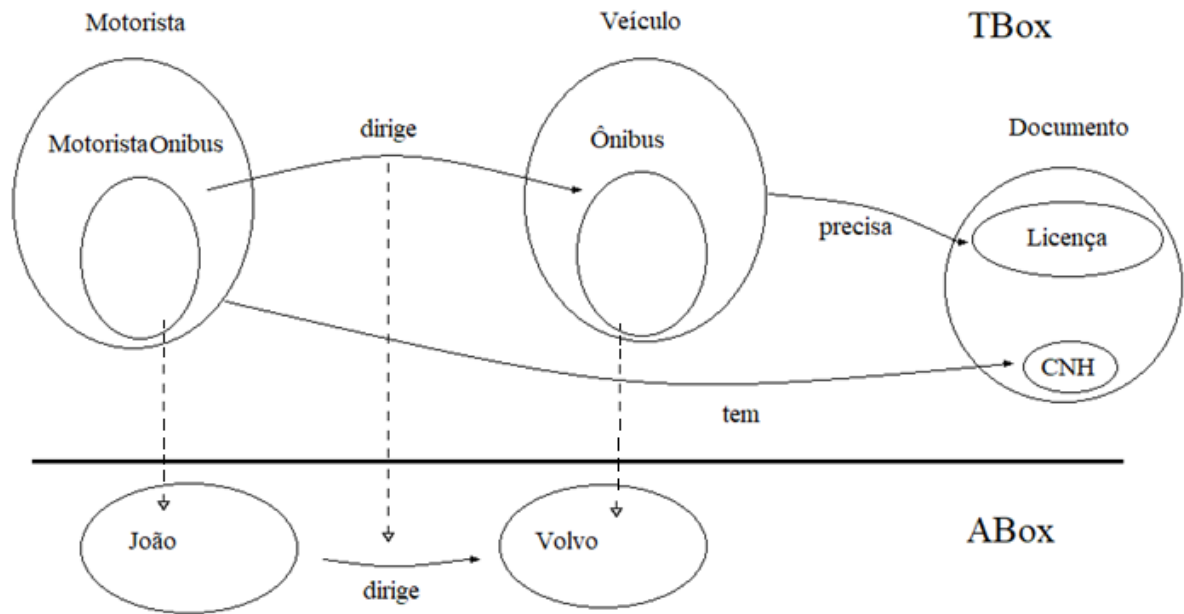
- João dirige Volvo;
- João é motorista de ônibus;
- Volvo é um ônibus;

Tabela 3 – Comparação entre sentenças da Abox e fórmulas puras da LH.

DL $\mathcal{ALC}$	LH
dirige(João, Volvo)	$@_J \langle d \rangle . V$
MotoristaOnibus(João)	$@_J MO$
Ônibus(Volvo)	$@_V O$

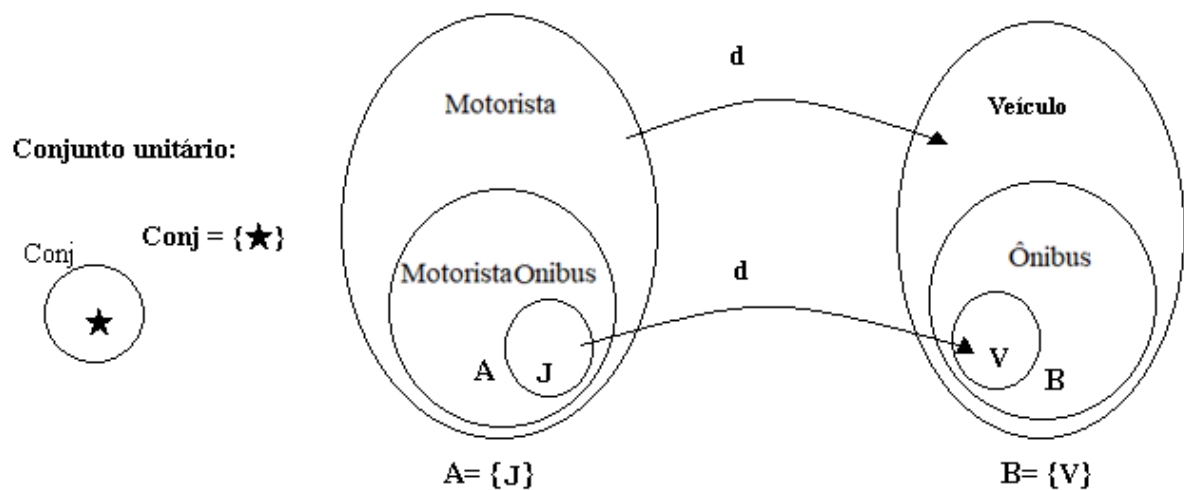
Fonte: Elaborada pelo autor (2022)

Figura 13 – Representação gráfica da ontologia do Exemplo 4 em DL $\mathcal{ALC}$.



Fonte: Elaborada pelo autor (2022)

Figura 14 – Representação gráfica de parte da mesma ontologia do Exemplo 4 em Lógica Híbrida do Exemplo 16; Observe-se que as instâncias estão representadas por conjuntos unitários e que a relação de acessibilidade *dirige* (*d*) liga tanto instâncias quanto classes.



Fonte: Elaborada pelo autor (2022)

5 CORRESPONDÊNCIA SOB A PERSPECTIVA DE RACIOCÍNIO AUTOMÁTICO

Com a intenção de aproximar ainda mais o leitor do entendimento e adicionar um viés mais prático, de raciocínio automático, nesta seção será apresentado e discutido o método de *tableau* ou tablô, como se encontra na literatura em português. O *tableau* semântico da teoria da prova se trata de um método de prova para satisfatibilidade de fórmulas. O *tableau* analítico é a forma que toma este método: uma árvore binária. Como escrito em (CARNIELLI; CONIGLIO; BIANCONI, 2006), o método foi introduzido por Beth e aperfeiçoado por Hintikka. É encontrado também que Smullyan simplificou posteriormente o método para a lógica clássica sob a alcunha de “one-sided tableau”. Em (CARNIELLI, 1991) é possível conferir o estabelecimento formal da simetria entre o método de *tableau* e o cálculo de seqüentes. Aqui o *tableau* é apresentado brevemente e utilizado como ferramenta de demonstração de equivalência: são usados *tableaux* como tarefa de raciocínio tanto em DL como em PML:

- Em DL $\mathcal{ALC}$, é essencialmente utilizado para raciocinar se algum conceito pode ser deduzido dos axiomas existentes, ou seja, se determinado conceito pertence à ontologia;
- Em PML $\mathbf{K}$, é essencialmente utilizado para raciocinar se um mundo deduz uma proposição, ou seja, se um mundo possível pertence ao conjunto de mundos possíveis daquele modelo;

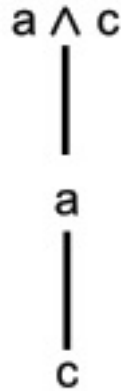
Assim, uma fórmula em $\mathcal{ALC}$ provada pertencente a uma ontologia através do *tableau* será mapeada a uma em $\mathbf{K}$ pertencente ao modelo equivalente à ontologia através do mapeamento. Em seqüência, serão apresentadas as regras de análise para os conectivos desde os clássicos, depois para os da Lógica de Descrições e, por fim, para o operador modal de necessidade, visto que os demais podem ser obtidos a partir dos apresentados. As regras na Definição 27 são inspiradas em (CARNIELLI; CONIGLIO; BIANCONI, 2006) para a Lógica Clássica, (GORÉ, 1999) para a Lógica Modal, (BAADER; SATTTLER, 2000) para a $\mathcal{ALC}$ e (BALBIANI; BOUDOU, 2015) para a Lógica Dinâmica Proposicional, mas apenas pela semelhança notacional, visto que (GORÉ, 1999) não trata especificamente da natureza multimodal.

Definição 27. (Regras de análise dos operadores clássicos): Sejam a , b e c proposições:

Regra do ($\wedge$): A ramificação correspondente continua sequencialmente, proposição por proposição. Esta regra se resume em considerar proposição por proposição, fórmula por fórmula, geralmente separadas por (,) ou (;) quando evidenciadas individualmente escritas como elementos do conjunto de proposições.

$$(\wedge): \frac{a \wedge c}{\frac{a}{c}}$$

Figura 15 – Ilustração da análise de comportamento da conjunção.

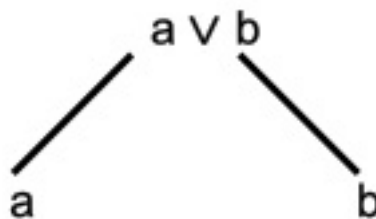


Fonte: Elaborada pelo autor (2022)

Regra do $\vee$: A ramificação correspondente bifurca em 2 ramos: o lado esquerdo do conectivo ramifica respectivamente, assim como o lado direito.

$$(\vee): \frac{a \vee b}{a | b}$$

Figura 16 – Ilustração da análise de comportamento da disjunção.



Fonte: Elaborada pelo autor (2022)

Regra da dupla negação ($\neg\neg$): A ramificação correspondente continua sequencialmente, removendo os operadores.

$$(\neg\neg): \frac{\neg\neg a}{a}$$

Figura 17 – Ilustração da análise de comportamento da dupla negação.

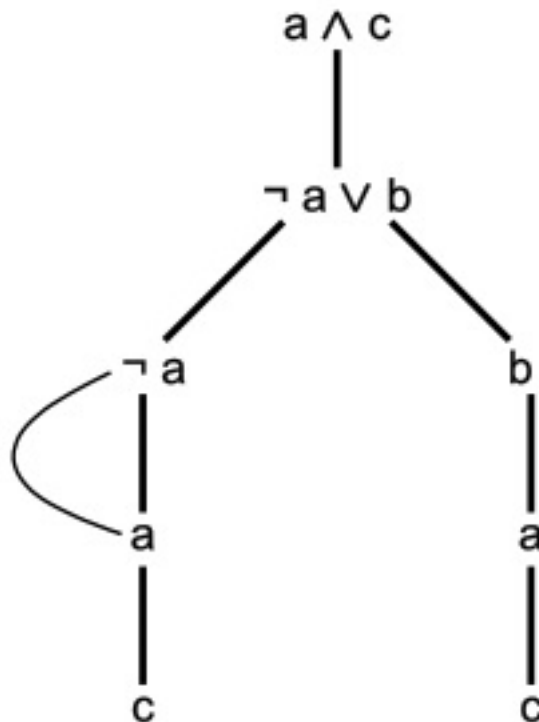


Fonte: Elaborada pelo autor (2022)

Definição 28. (Fechamento de Ramificação): Um ramo é dito fechado se, a qualquer passo do *tableau*, ocorrer uma contradição. Um ramo é dito aberto se todas as regras forem exauridas e ele não se tornar fechado. Para o teste de satisfatibilidade, o conjunto de fórmulas submetidas ao *tableau* é dito satisfatível se todos os ramos do *tableau* gerado terminam fechados.

Exemplo 17. Considere o conjunto de fórmulas $\{a \wedge c, \neg a \vee b\}$.

Figura 18 – Ilustração da análise de comportamento de um *tableau* simples.



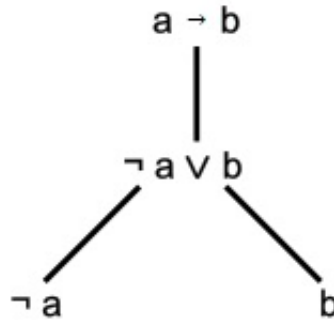
Fonte: Elaborada pelo autor (2022)

Os operadores complexos, formados como uma composição dos operadores que geraram as regras anteriores, não geram regras particulares por si só, mas demandam que sejam destrinchados em formas normais antes de operados sob o *tableau*. De fato, sequer a regra da disjunção tem necessidade de existir, visto que $(a \vee c)$ é congruente a $\neg(\neg a \wedge \neg c)$. Como menção e para comparação com regra posterior, é apresentada a regra de análise da implicação:

Definição 29. (Regra do $\rightarrow$): Assim como o operador de implicação tem a mesma tabela verdade que a disjunção da negação do primeiro com o segundo, no *tableau* é criada uma bifurcação em que o ramo da esquerda é negado e o da direita preservado.

$$(\rightarrow): \frac{a \rightarrow b}{\neg a \mid b}$$

Figura 19 – Ilustração da análise de comportamento do operador de implicação.



Fonte: Elaborada pelo autor (2022)

A Definição 30 apresenta as regras de análise para os operadores na Lógica de Descrições.

Definição 30. (Regras de análise para DL $\mathcal{ALC}$): Sejam A, B, C nomes de conceito, r, s nomes de papéis, x, y instâncias, todas as operações clássicas são válidas para DL, sendo as fórmulas tratadas como conceitos. Os operadores $\wedge$ e $\vee$ são trocados por $\sqcap$ e $\sqcup$ respectivamente e o operador de subsunção obedece a regra da implicação material. Os demais são definidos como segue:

Regra do $\sqsubseteq$: Assim como o operador de implicação, no *tableau* é criada uma bifurcação em que o ramo da esquerda é negado e o da direita preservado.

$$(\sqsubseteq): \frac{A \sqsubseteq B}{\neg A \mid B}$$

Regra do $\equiv$: O operador $\equiv$ funciona como o $\longleftrightarrow$ da lógica clássica. É criada uma bifurcação em que cada ramificação contenha uma conjunção dos termos analisados, uma preservados os termos e outra com a negação deles.

$$(\equiv): \frac{A \equiv B}{A \sqcap B \mid \neg A \sqcap \neg B}$$

Regra de instanciação: Os conceitos são instanciados e o *tableau* preservado.

$$(\sqcap): \frac{A \sqcap B(x)}{\frac{A(x)}{B(x)}}$$

Regra do $\exists$: O *tableau* segue a ramificação sequencialmente. Os conceitos são instanciados de modo que x representa instanciação já presente na Abox e y representa instanciação nova, não pertencente à Abox.

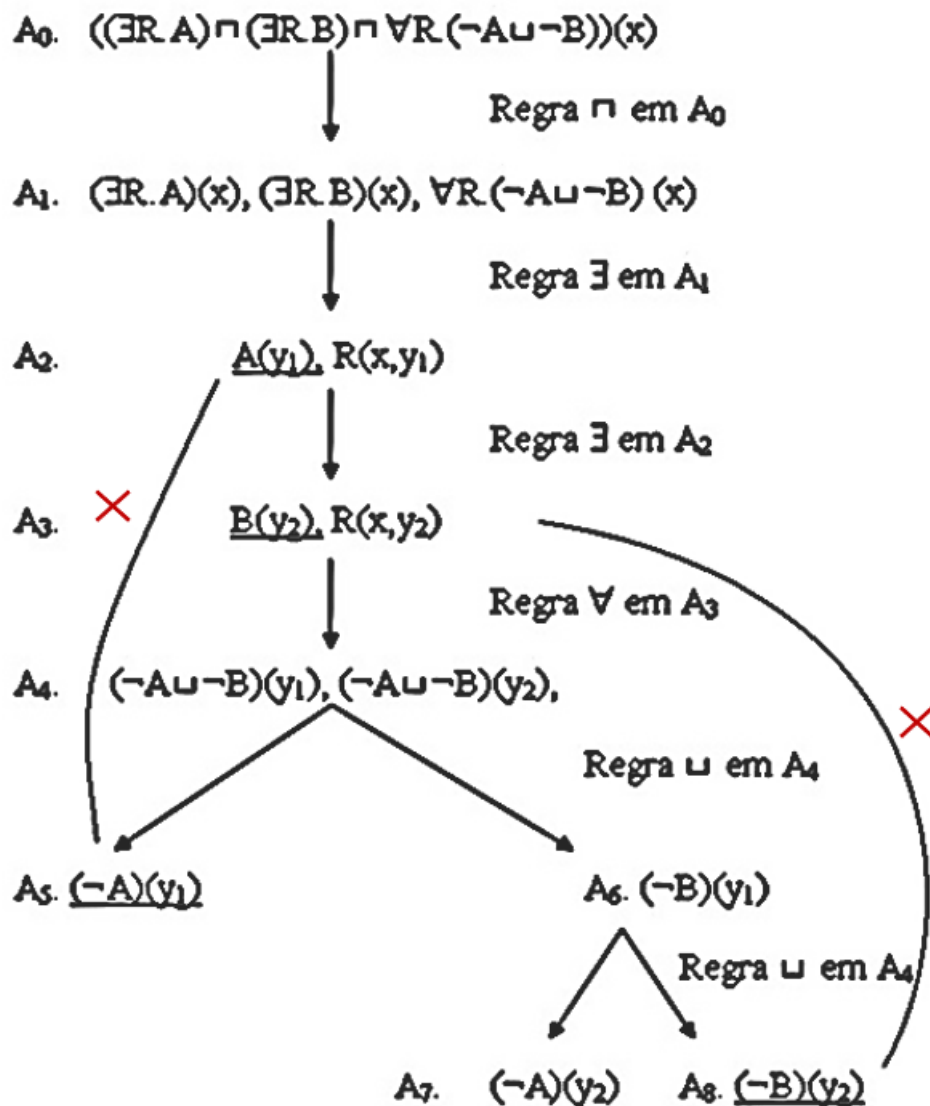
$$(\exists): \frac{\exists r.C(x)}{\frac{C(y)}{r(x,y)}}$$

Regra do $\forall$: O *tableau* segue a ramificação sequencialmente. O conceito universal precisa ser analisado juntamente com os papéis que quantifica.

$$(\forall): \frac{\forall r.C(x) \sqcap r(x,y)}{C(y)}$$

Exemplo 18. Sejam A e B nomes de conceitos e R o nome de um papel. Analisar se $(\exists R.A) \sqcap (\exists R.B)$ é subclasse de $\exists R.(A \sqcap B)$, ou seja se $(\exists R.A) \sqcap (\exists R.B) \sqsubseteq \exists R.(A \sqcap B)$. O primeiro passo é negar a expressão dada, ou seja, colocá-la na forma $C \sqcap \neg D$, tal que C equivale a $(\exists R.A) \sqcap (\exists R.B)$ e $\neg D$ equivale a $(\neg \exists R.(A \sqcap B))$. Após a negação, obtêm-se a expressão $(\exists R.A) \sqcap (\exists R.B) \sqcap \forall R.(\neg A \sqcup \neg B)$. Os outros passos da prova são representados na próxima figura.

Figura 20 – *Tableau* correspondente ao Exemplo 18.

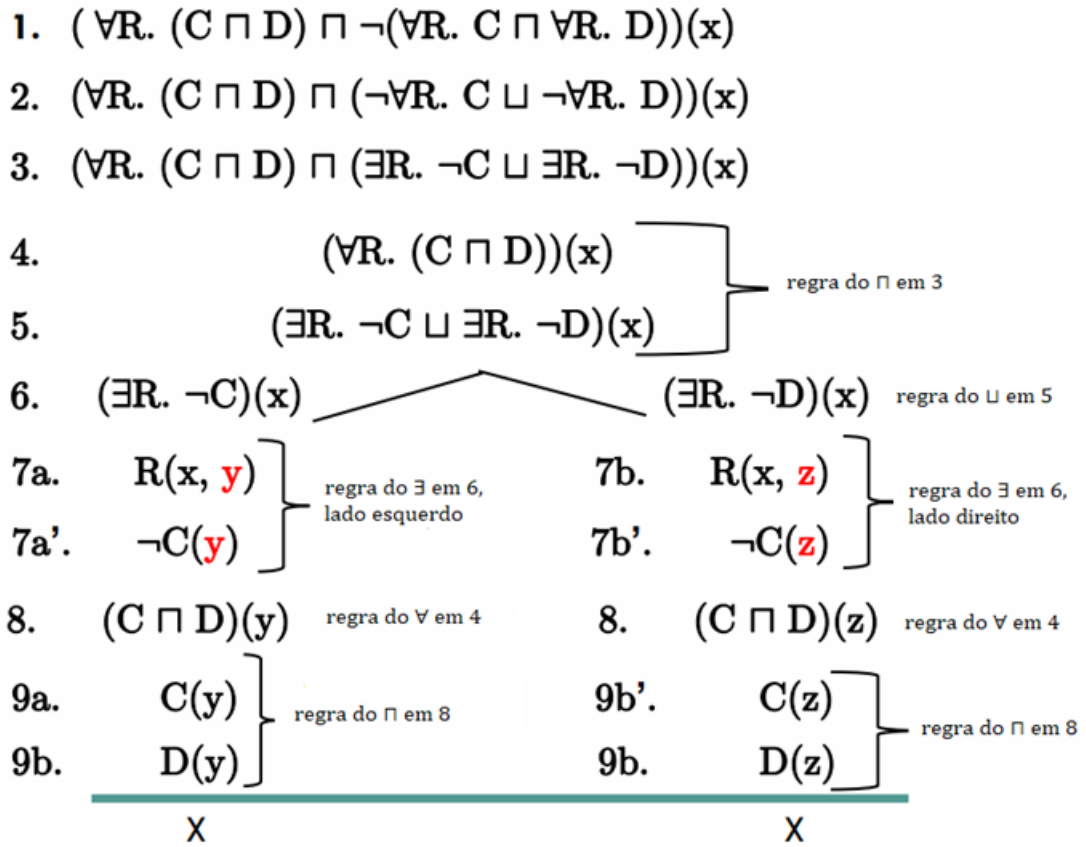


Fonte: Elaborada pelo autor (2022)

Como pode ser observado, há um ramo que não fecha, portanto $(\exists R.A) \sqcap (\exists R.B)$ não é subclasse de $\exists R.(A \sqcap B)$.

Exemplo 19. Analisar se $\forall R.(C \sqcap D) \sqsubseteq (\forall R.C) \sqcap (\forall R.D)$: primeiro, organizar as fórmulas da forma $\forall R.(C \sqcap D) \sqcap \neg(\forall R.C \sqcap \forall R.D)$. O algoritmo pode ser conferido como segue:

Figura 21 – *Tableau* correspondente ao Exemplo 19.



Fonte: Elaborada pelo autor (2022)

Como pode ser observado, todos os ramos fecham, portanto $\forall R.(C \sqcap D)$ é subclasse de $(\forall R.C) \sqcap (\forall R.D)$.

Goré (1999) traz um detalhamento interessante sobre como os Kripke *frames*, modelos, ou seja, a semântica de Kripke pode ser trabalhada com o método de *tableau*. Com efeito, a regra da necessitação da Definição 33 é citada e, posteriormente, usada como regra de análise, a única que trabalha com o operador modal. No texto que escreve, ele chama tal regra de transicional (K), enquanto as demais, dos operadores clássicos, de estáticas, e trata o fechamento de ramo como uma regra, a do ($\perp$). O Exemplo 20 ilustra como elas são usadas na prática. No presente trabalho, esta regra (K) é chamada de simplesmente regra do $\Box$. A regra (K) diz respeito ao axioma **K**, da Lógica Modal normal mais básica, sendo os demais axiomas tratados como regra individualmente. A Figura 33 a seguir ilustra tais regras. Nela, X, Y são conjuntos de fórmulas

bem formadas e P conjunto de proposições.

A Definição 31 a seguir diz respeito ao operador modal $\Box$, tradicionalmente lido como necessidade. Como exemplo, o axioma de Aristóteles é ilustrado inspirado em Goré (1999). No apêndice 33, os leitores interessados podem visualizar uma lista com as regras de análise para os axiomas definidores das modalidades.

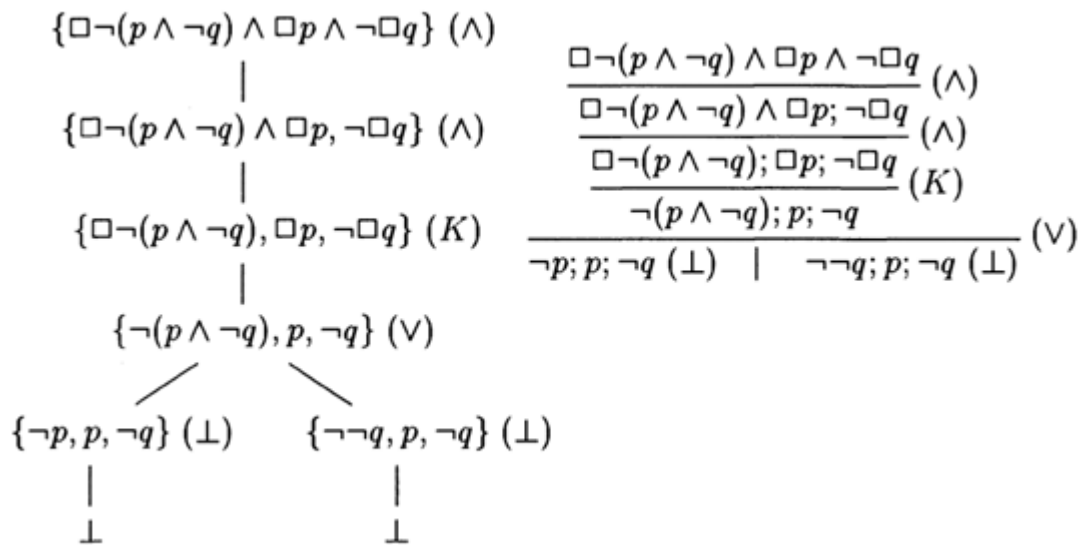
Definição 31. (Regra de análise do operador modal $\Box$): Seja p uma proposição ou P uma fórmula, a regra do $\Box$ é definida como segue:

Regra do $\Box$: o *tableau* continua sequencialmente, eliminando o operador $\Box$. Esta regra se dá pela análise da definição semântica do operador: ou não existe relação, ou existe e p é deduzido. Como a primeira possibilidade não serve ao propósito, a segunda é assumida em análise.

$$(\Box): \frac{\Box p}{p}; \frac{\neg \Box p}{\neg p}$$

Exemplo 20. A fórmula $K = \{\Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q)\}$ tem o *tableau* construído da seguinte forma: primeiro, as implicações podem ser destrinchadas como anteriormente citado, ou trabalhadas como na Definição 29. Seguindo como por deduzir as implicações, a fórmula ganha a forma de $K = \{\Box(\neg(p \wedge \neg q)) \wedge \Box p \wedge \neg \Box q\}$

Figura 22 – *Tableau* correspondente à fórmula K.



Fonte: Goré (1999)

Em sequência, são apresentadas as regras básicas para os operadores conforme a sintaxe multimodal de $\mathbf{K}_m$.

Definição 32. (Regras de análise em $\mathbf{K}_m$): Sejam α, β programas e φ, ψ fórmulas, as regras de análise de *tableau* são definidas como:

Regra do $[]$: Assim como a regra do $\forall$, o *tableau* segue a ramificação sequencialmente.

$$([]): \frac{[\alpha] \varphi(x) \wedge \alpha(x, y)}{\varphi(y)}$$

Regra do $\langle \rangle$: Assim como a regra do $\exists$, o *tableau* segue a ramificação sequencialmente.

$$(\langle \rangle): \frac{\langle \alpha \rangle \varphi(x)}{\frac{\alpha(x, y)}{\varphi(y)}}$$

O Exemplo 21 ilustra a correspondência (considerando as regras dos operadores traduzidos para $\mathbf{K}_m$).

6 CONCLUSÕES

Esta dissertação se propôs a explicar a problemática da correspondência entre as lógicas de Descrições e Modais de maneira didática para que pudesse ser lida por vários tipos de audiência. Nesse sentido, uma densa revisão da literatura revelou a carência de um texto único que descrevesse minuciosamente os formalismos em questão, bem como onde e de quais formas as equivalências ocorrem. A correspondência mais simples, mas principal, está descrita até o capítulo 5, passando por sintaxe e semântica, inclusive semelhanças em raciocínio, usando para tal o método de *tableaux* de cada uma das lógicas. Espera-se, com este texto, preencher uma lacuna, visto que na literatura não se encontra tanto material com viés didático a este respeito e desta forma como este foi construído. O texto trouxe bastante exemplos, procurando tornar a leitura ainda mais palatável.

No apêndice B há uma correspondência alternativa entre Lógica de Descrições e Lógica Dinâmica. Esta parte do texto, sendo matematicamente mais densa pela própria natureza do assunto, foi apresentada com menor preocupação didática. A tentativa de estabelecer correspondências entre as Lógicas de Descrições e Lógica Modal e Dinâmica se mostrou como sendo interessante por fazerem correlatos campos não relacionados. Foi possível compreender que as correspondências completas ocorreram apenas em ramificações com níveis mais altos de complexidade, neste caso entre $\mathcal{ALC}_{reg}$ e PDL e entre $\mathcal{ALCT}_{reg}$ e *converse*-PDL.

Como apresentado, foi necessário enriquecer o sistema **K** monomodal para sua versão multimodal **K_m** permitindo que a correspondência com $\mathcal{ALC}$ em nível de TBox ocorresse de maneira completa. Devido a sua natureza distinta das Lógicas de Descrições, a PML não contém em sua definição um mecanismo para tratar instanciações, intrínseco da linguagem $\mathcal{ALC}$ com a ABox. Logo, a apresentação da Lógica Híbrida se mostrou necessária e concluiu a correspondência entre a Lógica Modal Proposicional multimodal enriquecida com o operador @ de nominais e a DL $\mathcal{ALC}$. Em contrapartida, ao se tentar estabelecer uma correspondência entre $\mathcal{ALC}$ e PDL, o inverso ocorre: $\mathcal{ALC}$ é trivialmente menos expressiva que PDL. Portanto, ocorreu a necessidade de demonstrar uma versão de $\mathcal{ALC}$ que pudesse corresponder com as operações de iteração e execução sequencial. Obtida através de um enriquecimento da linguagem $\mathcal{FL}_0$, foi apresentada a $\mathcal{ALC}_{reg}$ e sua correspondência com PDL.

A análise sobre a correspondência visou ajudar o leitor na aquisição de conteúdo. Mesmo assim, desde o início do texto, tanto para DL quanto MDL (ou PML) é exibida a equivalência das expressões para a LPO. Ora, uma vez tradutíveis para uma terceira linguagem em comum, o

indivíduo pode estabelecer a correspondência entre ambas as linguagens iniciais. Esta é uma possível continuação deste trabalho.

Fica como proposta para adição futura a demonstração das correspondências de mais ramificações dos formalismos, como é o caso da linguagem $\mathcal{SR}$. Também em nível algorítmico, uma comparação entre os algoritmos de *tableaux* para verificação de satisfatibilidade para as linguagens que já o possuem e a confecção do sistema de *tableaux* para aquelas que ainda o carecem pode ser uma proposta interessante. Por fim, conceber o uso de uma correspondência a nível de modalidades, ou seja, a correspondência entre uma ramificação de DL com a Lógica Deôntica, ou Lógica Tensa promete ser um trabalho interessante.

REFERÊNCIAS

- AHO, A.; ULLMAN, J. Universality of data retrieval languages. **Proceedings of the 6th ACM SIGACT-SIGPLAN symposium on Principles of programming languages**, 1979. Citado 2 vezes nas páginas 102, 103.
- ARECES, C.; BLACKBURN, P.; MARX, M. A Road-Map on Complexity for Hybrid Logics. In: ISBN 978-3-540-66536-6. DOI: 10.1007/3-540-48168-0_22. Citado 2 vezes nas páginas 51, 52.
- ARECES, C.; RIJKE, M. Description and/or Hybrid Logics. **Journal of Applied Physics - J APPL PHYS**, jan. 2000. Citado 1 vez na página 53.
- ASSIS, M. de. **O Punhal de Martinha**. [S.l.]: SESC, 1894. Disponível em: <<https://zdocs.mx/doc/o-punhal-de-martinha-y01rjnzerlpg>>. Citado 1 vez na página 6.
- BAADER, F.; SATTTLER, U. Tableau Algorithms for Description Logics. In: DYCKHOFF, R. (Ed.). **AUTOMATED REASONING WITH ANALYTIC TABLEAUX AND RELATED METHODS. TABLEAUX 2000. Lecture Notes in Computer Science**. Berlin, Heidelberg: Springer, 2000. v. 1847. DOI: 10.1007/10722086_1. Citado 2 vezes nas páginas 19, 56.
- BAADER, F. et al. **The Description Logic Handbook: Theory, Implementation and Applications**. en. 2a. Nova York, NY, EUA: Cambridge University Press, 2010. ISBN: 0521150116, 9780521150118. Citado 3 vezes nas páginas 18, 20, 29–31.
- BAADER, F. A Formal Definition for the Expressive Power of Terminological Knowledge Representation Languages. **Journal of Logic and Computation**, v. 6, n. 1, p. 33–54, fev. 1996. ISSN 0955-792X. DOI: 10.1093/logcom/6.1.33. eprint: <https://academic.oup.com/logcom/article-pdf/6/1/33/3039314/6-1-33.pdf>. Citado 1 vez na página 109.
- _____. Augmenting Concept Languages by Transitive Closure of Roles: An Alternative to Terminological Cycles. In: **PROCEEDINGS of the 12th International Joint Conference on Artificial Intelligence - Volume 1**. Sydney, New South Wales, Australia: Morgan Kaufmann Publishers Inc., 1991. (IJCAI'91), p. 446–451. ISBN 1558601600. DOI: 10.5555/1631171.1631238. Citado 7 vezes nas páginas 18, 22, 25, 100, 102, 107, 109.

BAADER, F. Description Logics. In: **Reasoning Web. Semantic Technologies for Information Systems: 5th International Summer School**. Edição: Sergio Tessaris. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. P. 1–39. ISBN 978-3-642-03754-2. DOI: 10.1007/978-3-642-03754-2_1. Citado 1 vez na página 36.

_____. Terminological Cycles in KL-ONE-Based Knowledge Representation Languages. In: **PROCEEDINGS of the Eighth National Conference on Artificial Intelligence**. Boston, Massachusetts: AAAI Press, 1990. (AAAI'90), p. 621–626. ISBN 026251057X. Citado 4 vezes nas páginas 102, 103, 105.

BAADER, F.; HORROCKS, I.; SATTler, U. Description Logics. In: **Handbook of Knowledge Representation**. Edição: Frank van Harmelen, Vladimir Lifschitz e Bruce Porter. [S.l.]: Elsevier, 2008. v. 3, p. 135–179. (Foundations of Artificial Intelligence). DOI: 10.1016/S1574-6526(07)03003-9. Citado 7 vezes nas páginas 18, 19, 25, 50, 103, 104.

BAADER, F. et al. **An Introduction to Description Logic**. [S.l.]: Cambridge University Press, 2017. DOI: 10.1017/9781139025355. Citado 1 vez na página 26.

BAADER, F. et al. **The Description Logic Handbook: Theory, Implementation and Applications**. 2. ed. [S.l.]: Cambridge University Press, 2007. DOI: 10.1017/CBO9780511711787. Citado 13 vezes nas páginas 18, 19, 25, 50, 98, 116, 117, 125.

BALBIANI, P.; BOUDOU, J. TABLEAUX METHODS FOR PROPOSITIONAL DYNAMIC LOGICS WITH SEPARATING PARALLEL COMPOSITION. en. In: **INTERNATIONAL Conference on Automated Deduction (CADE 2015)**. Berlin, Germany: [s.n.], 2015. 539–554–01650183. Citado 2 vezes nas páginas 56, 123.

BAŞKENT, C. **A Quick Introduction to Dynamic Logic**. [S.l.]: Istanbul Kultur University, 2010. Citado 2 vezes nas páginas 76, 91.

BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. The Semantic Web. en. **Scientific American**, v. v. 284, p. 34–43, 2001. Citado 0 vez na página 23.

BEZERRA, C. **Uma abordagem de modularização de ontologias baseada na satisfação lógica de questões de competência**. 2019. Tese (Doutorado) – Curso de Ciência da Computação — Universidade Federal de Pernambuco, Recife. Citado 2 vezes nas páginas 18, 23, 25.

BLACKBURN, P. Arthur prior and hybrid logic. English. **Synthese**, Springer Netherlands, v. 150, n. 3, p. 329–372, jun. 2006. ISSN 0039-7857. DOI: 10.1007/s11229-005-5512-y. Citado 1 vez na página 51.

BLACKBURN, P.; BENTHEM, J. F. A. K. v.; WOLTER, F. **Handbook of Modal Logic, Volume 3 (Studies in Logic and Practical Reasoning)**. USA: Elsevier Science Inc., 2006. ISBN 0444516905. Citado 1 vez na página 38.

BORGIDA, A. On the relative expressiveness of description logics and predicate logics. **Artificial Intelligence**, v. 82, n. 1, p. 353–367, 1996. ISSN 0004-3702. DOI: 10.1016/0004-3702(96)00004-5. Citado 1 vez na página 109.

BRACHMAN, R. J.; SCHMOLZE, J. G. An overview of the KL-ONE Knowledge Representation System. **Cognitive Science**, v. 9, n. 2, p. 171–216, 1985. ISSN 0364-0213. DOI: [https://doi.org/10.1016/S0364-0213\(85\)80014-8](https://doi.org/10.1016/S0364-0213(85)80014-8). Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0364021385800148>>. Citado 1 vez na página 18.

BULL, R.; SEGERBERG, K. Basic Modal Logic. In: GABBAY, D. M.; GUENTHNER, F. (Ed.). **Handbook of philosophical logic**. 2nd. Dordrecht, Boston: Kluwer Academic Publishers, 2001. v. 2. P. 1–88. Citado 8 vezes nas páginas 37, 76, 80, 83, 95, 119.

CARNIELLI, W. On sequents and tableaux for many-valued logics. en. **The Journal of Non-Classical Logics**, v. 8, n. 1, p. 59–76, 1991. Citado 1 vez na página 56.

CARNIELLI, W.; CONIGLIO, M.; BIANCONI, R. **Lógica e aplicações: matemática, ciência da computação e filosofia. Versão preliminar: capítulos 1 a 5**. pt. Campinas: Universidade Estadual de Campinas, Unicamp, 2006. Disponível em: disponível em: <<http://www.cle.unicamp.br/prof/coniglio/LIVRO.pdf>>. Citado 2 vezes na página 56.

CARNIELLI, W.; PIZZI, C. **Modalities and Multimodalities**. [S.l.]: Springer, 2008. Citado 5 vezes nas páginas 36, 37, 45, 77, 79.

CC BY-SA 4.0, C. C. **Creative Commons. Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)**. [S.l.: s.n.]. Disponível em: <<https://creativecommons.org/licenses/by-sa/4.0/>>. Citado 0 vez na página 78.

COCCHIARELLA, N. B.; FREUND, M. A. **Modal Logic: An Introduction to its Syntax and Semantics**. [S.l.]: Oxford e New York: Oxford University Press, 2008. Citado 1 vez na página 37.

DAVEY, B.; PRIESTLEY, H. **Introduction to Lattices and Order**. 2a. Cambridge: Cambridge University Press, 2002. DOI: 10.1017/CBO9780511809088. Citado 1 vez na página 127.

DIVROODI, A. R. Comparing the Expressiveness of Description Logics. In: **DESCRIPTION Logics**. [S.l.: s.n.], 2014. Citado 4 vezes nas páginas 109–111.

FELIX FILHO, H. P. **Lógica Modal: Uma abordagem introdutória**. 2019. Monografia (Bacharelado em Engenharia da Computação) – Universidade Federal de Pernambuco, Recife. Citado 3 vezes nas páginas 18, 36, 37.

FISCHER, M. J.; LADNER, R. E. Propositional dynamic logic of regular programs. **Journal of Computer and System Sciences**, v. 18, n. 2, p. 194–211, 1979. ISSN 0022-0000. DOI: [https://doi.org/10.1016/0022-0000\(79\)90046-1](https://doi.org/10.1016/0022-0000(79)90046-1). Disponível em: <<https://www.sciencedirect.com/science/article/pii/0022000079900461>>. Citado 3 vezes nas páginas 84, 86.

GABBAY, D. M.; GUENTHNER, F. **HANDBOOK OF PHILOSOPHICAL LOGIC**. [S.l.: s.n.], 2005. ISBN 978-1-4020-3520-3. Citado 2 vezes na página 84.

GABBAY, D. M. A general filtration method for modal logics. In: DOI: 10.1007/BF00649988. Citado 3 vezes nas páginas 95, 118.

GORANKO, V.; RUMBERG, A. Temporal Logic", The Stanford Encyclopedia of Philosophy. In: EDITION), E. N. (Ed.). URL =. [S.l.: s.n.], 2021. Disponível em: <<https://plato.stanford.edu/archives/fall2021/entries/logic-temporal/>>. Citado 2 vezes nas páginas 19, 78.

GORÉ, R. Tableaux Method for Modal and Temporal Logics. In: D'AGOSTINO, M. et al. (Ed.). **HANDBOOK OF TABLEAU METHODS**. [S.l.]: Springer, 1999. ISBN: 978-94-017-1754-0. P. 297–396. DOI: 10.1007/978-94-017-1754-0.. Citado 6 vezes nas páginas 19, 56, 62, 63, 77, 124.

HANSSON, B. An Analysis of some Deontic Logics. **Noûs**, Wiley, v. 3, n. 4, p. 373–398, 1969. ISSN 00294624, 14680068. Disponível em: <<http://www.jstor.org/stable/2214372>>. Citado 1 vez na página 80.

HAREL, D.; KOZEN, D.; TIURYN, J. **Dynamic Logic. Foundations of Computing**. [S.l.]: Cambridge, MA: MIT Press, 2000. Citado 27 vezes nas páginas 40, 45, 46, 84–88, 91–93, 95, 96, 112, 114, 115, 119, 121, 129.

HENNESSY, M.; MILNER, R. **On observing nondeterminism and concurrency. Automata, Languages and Programming. Lecture Notes in Computer Science**. es. Berlin, Heidelberg: Springer, 1980. P. 299–309. ISBN: 978-3540100034. DOI: 10.1007/3-540-10003-2_79.. Citado 1 vez na página 85.

HOARE, C. An axiomatic basis for computer programming. **Commun. ACM**, v. 12, p. 576–580, 1969. Citado 0 vez na página 84.

HOLLIDAY, W. **Knowing What Follows, Epistemic Closure and Epistemic Logic, Dissertation**. [S.l.]: Department of Philosophy, Stanford University, 2012. Citado 1 vez na página 80.

HRBACEK, K.; JECH, T. **Introduction to Set Theory**. en. Revised and Expanded. 3a. Reino Unido: CRC Press, 1999. ISBN: 9780824779153. Citado 3 vezes nas páginas 30, 118, 127.

INDUCTION., M. O. P. **Postado em 21/03/2013 by j2kun**. pt. [S.l.: s.n.]. Disponível em: disponível em: <<https://jeremykun.com/2013/03/21/methods-of-proof-induction/>>. Citado 1 vez na página 95.

KFOURY, A.; PARK, D. On the termination of program schemas. **Information and Control**, v. 29, n. 3, p. 243–251, 1975. ISSN 0019-9958. DOI: [https://doi.org/10.1016/S0019-9958\(75\)90415-5](https://doi.org/10.1016/S0019-9958(75)90415-5). Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0019995875904155>>. Citado 1 vez na página 84.

KOZEN, D. **Results on the Propositional μ -Calculus**. en. v. 11. [S.l.], 1982. DOI: DOI: 10.7146/dpb.v11i146.7420.. Disponível em: <<https://doi.org/10.7146/dpb.v11i146.7420.>>. Citado 2 vezes na página 114.

KOZEN, D. C. **Automata and Computability**. 1st. Berlin, Heidelberg: Springer-Verlag, 1997. ISBN 0387949070. Citado 1 vez na página 129.

LEMMON, E. et al. A general filtration method for modal logics. en. **Journal of Philosophic Logic**, n. 1, p. 1, 1966. Citado 0 vez na página 118.

LEMMON, E. J. **An introduction to modal logic**. Blackwell: Oxford, 1977. in collaboration with D. Scott. Citado 0 vez na página 76.

LIU, F. **Reasoning About Preference Dynamics**. [S.l.]: Dordrecht, Netherland: Springer Verlag, 2011. Citado 1 vez na página 80.

MARCUS, R. B. Modal logic, modal semantics and their applications. In: **Tome 1 Philosophie du langage, Logique philosophique / Volume 1 Philosophy of language, Philosophical logic**. Edição: Guttorm Fløistad e G. H. von Wright. Dordrecht: Springer Netherlands, 1981. P. 279–298. ISBN 978-94-009-8356-4. DOI: 10.1007/978-94-009-8356-4_11. Disponível em: <https://doi.org/10.1007/978-94-009-8356-4_11>. Citado 2 vezes nas páginas 36, 118.

MARX, M. 3 Complexity of modal logic. In: BLACKBURN, P.; VAN BENTHEM, J.; WOLTER, F. (Ed.). **Handbook of Modal Logic**. [S.l.]: Elsevier, 2007. v. 3. (Studies in Logic and Practical Reasoning). P. 139–179. DOI: [https://doi.org/10.1016/S1570-2464\(07\)80006-1](https://doi.org/10.1016/S1570-2464(07)80006-1). Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1570246407800061>>. Citado 1 vez na página 40.

MASSACCI, F. Single Step Tableaux for Modal Logics. **J. Autom. Reasoning**, v. 24, p. 319–364, abr. 2000. DOI: 10.1023/A:1006155811656. Citado 1 vez na página 76.

MCNAMARA, P.; PUTTE, F. V. D. Deontic Logic". en. **The Stanford Encyclopedia of Philosophy**, 2021. Edition), Edward N. Zalta (ed.), URL =. Disponível em: <<https://plato.stanford.edu/archives/spr2021/entries/logic-deontic/>>. Citado 1 vez na página 79.

MEGHINI, C.; STRACCIA, U. A Relevance Terminological Logic for Information Retrieval. In: p. 197–205. DOI: 10.1145/243199.243267. Citado 1 vez na página 18.

MENDES, A. et al. Modality annotation for Portuguese: from manual annotation to automatic labeling. **Linguistic Issues in Language Technology**, v. 14, set. 2016. Citado 1 vez na página 80.

MOSS, L. S.; TIEDE, H.-J. 19 Applications of modal logic in linguistics. In: BLACKBURN, P.; VAN BENTHEM, J.; WOLTER, F. (Ed.). **Handbook of Modal Logic**. [S.l.]: Elsevier, 2007. v. 3. (Studies in Logic and Practical Reasoning). P. 1031–1076. DOI: [https://doi.org/10.1016/S1570-2464\(07\)80022-X](https://doi.org/10.1016/S1570-2464(07)80022-X). Disponível em: <<https://www.sciencedirect.com/science/article/pii/S157024640780022X>>. Citado 1 vez na página 38.

NEBEL, B. **Reasoning and Revision in Hybrid Representation Systems**. [S.l.]: Springer-Verlag, 1990a. DOI: 10.1007/bfb0016445. Citado 1 vez na página 103.

_____. Terminological reasoning is inherently intractable. **Artificial Intelligence**, v. 43, n. 2, p. 235–249, 1990b. ISSN 0004-3702. DOI: 10.1016/0004-3702(90)90087-G. Citado 1 vez na página 102.

OHLBACH, H. J.; KOEHLER, J. Modal logics, description logics and arithmetic reasoning. **Artificial Intelligence**, v. 109, n. 1, p. 1–31, 1999. ISSN 0004-3702. DOI: [https://doi.org/10.1016/S0004-3702\(99\)00011-9](https://doi.org/10.1016/S0004-3702(99)00011-9). Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0004370299000119>>. Citado 2 vezes nas páginas 20, 48.

PATEL-SCHNEIDER, P. F.; HAYES, P.; HORROCKS, I. **OWL - Web Ontology Language Semantics and Abstract Syntax W3C Recommendation**. 2004. Disponível em: <www.w3.org/tr/2004/rec-owl-semantics-20040210/>. Citado 2 vezes nas páginas 18, 23.

PRATT, V. R. A decidable mu-calculus: Preliminary report. es. In: 22O ANNUAL SYMPOSIUM ON FOUNDATIONS OF COMPUTER SCIENCE. Nashville, TN, EUA: [s.n.], 1981a. P. 421–427. DOI: 10.1109/SFCS.1981.4.. Citado 1 vez na página 114.

_____. Semantical considerations on Floyd-Hoare logic. en. In: PROC. 17o SYMP. FOUND. COMPUT. SCI. [S.l.]: IEEE, 1976. P. 109–121. Citado 1 vez na página 84.

PRATT, V. R. A decidable μ -calculus: Preliminary report. In: ANNUAL SYMPOSIUM ON FOUNDATIONS OF COMPUTER SCIENCE, 22., 1981, Nashville. **Anais [...]** Nashville: IEEE, 1981b. DOI: 10.1109/SFCS.1981.4. Citado 1 vez na página 114.

RIJKE, M. D. Modal Logics and Description Logics. In: PROC. DL'98. [S.l.: s.n.], 1998. P. 1–3. Citado 3 vezes nas páginas 18–20.

RUDOLPH, S. Foundations of Description Logics. In: **Reasoning Web. Semantic Technologies for the Web of Data: 7th International Summer School 2011, Galway, Ireland, August 23-27, 2011, Tutorial Lectures**. Edição: Axel Polleres. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. P. 76–136. ISBN 978-3-642-23032-5. DOI: 10.1007/978-3-642-23032-5_2. Disponível em: <https://doi.org/10.1007/978-3-642-23032-5_2>. Citado 1 vez na página 25.

SCHILD, K. A correspondence theory for terminological logics: Preliminary report. en. In: PROC. IJCAI-91. Sydney, Australia: Morgan Kaufmann, 1991a. P. 466–471. Citado 1 vez na página 117.

SCHILD, K. A Correspondence Theory for Terminological Logics: Preliminary Report. In: _____. **Proceedings of the 12th International Joint Conference on Artificial Intelligence - Volume 1**. Sydney, New South Wales, Australia: Morgan Kaufmann, 1991b. (IJCAI'91), p. 466–71. ISBN 1558601600. Disponível em: <<http://ijcai.org/Proceedings/91-1/Papers/072.pdf>>. Citado 2 vezes na página 116.

SCHMIDT-SCHAU β , M.; SMOLKA, G. Attributive concept descriptions with complements. **Artificial Intelligence**, v. 48, n. 1, p. 1–26, 1991. ISSN 0004-3702. DOI: [https://doi.org/10.1016/0004-3702\(91\)90078-X](https://doi.org/10.1016/0004-3702(91)90078-X). Disponível em: <<https://www.scie>

ncedirect.com/science/article/pii/000437029190078X>. Citado 0 vez na página 25.

SCOTT, D. S.; J., B. W. A theory of programs. sv. **IBM Vienna**, 1969. Citado 1 vez na página 114.

SEGERBERG, K. A completeness theorem in the modal logic of programs. pt. **Banach Center Publications**, v. 9, n. 1, p. 31–46, 1982. Disponível em: disponível em: <<http://eudml.org/doc/209235>>. Citado 2 vezes nas páginas 88, 114.

SIPSER, M. **Introduction to the theory of computation**. [S.l.]: Boston : PWS Pub. Co., 1997. Citado 2 vezes nas páginas 105, 126.

STREETT, R. S. Fixpoints and program looping: reductions from the propositional μ -calculus into propositional dynamic logics of looping. In: PRAC. WORKSHOP ON LOGICS OF PROGRAMS, Lect. Notes in Comput. Sci. v. 193. [S.l.]: Springer-Verlag, 1985. P. 359–372. Citado 5 vezes nas páginas 84, 113–115.

TROQUARD, N.; BALBIANI, P. **Propositional Dynamic Logic**, **The Stanford Encyclopedia of Philosophy**. 2019. Disponível em: <<https://plato.stanford.edu/archives/spr2019/entries/logic-dynamic/>>. Citado 4 vezes nas páginas 84, 85, 91, 93.

VARDI, M. Y.; WOLPER, P. Automata-Theoretic Techniques for Modal Logics of Programs. **J. Comput. Syst. Sci.**, Academic Press, USA, v. 32, n. 2, p. 183–221, abr. 1986. ISSN 0022-0000. DOI: 10.1016/0022-0000(86)90026-7. Citado 1 vez na página 116.

APÊNDICE A – LÓGICAS MODAIS NORMAIS E MAPEAMENTO PARA LPO

Com um número grande de lógicas desenvolvidas, pode ocorrer um problema com a nomenclatura. Aquela adotada neste trabalho segue de Lemmon (1977 apud BULL; SEGERBERG, 2001, p. 18), cujo menor sistema é designado **K**, homenagem a Saul Kripke que, segundo Bull e Segerberg (2001, p. 18), nunca lidou com este sistema particular.

Definição 33. (O Sistema K): O sistema **K** pode ser definido como contendo:

- Axioma da distributividade⁷: $\Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q)$;
- Regra da necessitação: se p é teorema de **K**, $\Box p$ também é.

Na Lógica Modal, o enriquecimento da linguagem formal se dá pela adição de enunciados como axiomas. Algumas modalidades ganham nomenclaturas específicas, como é o caso das modalidades Epistêmica, Deôntica e Tensa, detalhadas mais adiante. Cada um desses enunciados representa a definição matemática daquilo que era uma demanda semântica. Por exemplo, a modalidade Deôntica lida com a interpretação do que é permitido ou proibido. O "obrigatório" é a interpretação nesta modalidade do operador de necessidade e o "permitido", de possibilidade. Ela tem como axioma a fórmula $\Box\varphi \rightarrow \Diamond\varphi$, ou seja, tudo aquilo que é obrigatório é permitido. Além desta fórmula como axioma, ela tem o sistema **K** como base. Todas as Lógicas Modais que têm o sistema **K** como base de seus axiomas são chamadas de Lógicas Modais normais.

No Quadro 1 podem ser verificadas lógicas modais normais. A iniciativa da nomenclatura de siglas para os inúmeros sistemas é de Lemmon, como em Bull e Segerberg (2001, p. 18). Em Massacci (2000) pode ser verificada um quadro semelhante.

Quadro 1 – Sistemas Modais Normais.

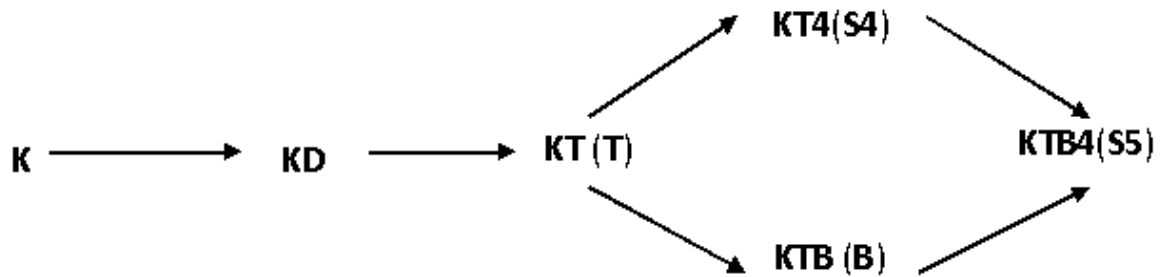
Nome	Axioma	Relação de Acessibilidade $\mathfrak{R}$
K	K	Qualquer relação
D	K + $\Box\varphi \rightarrow \Diamond\varphi$	Serial
CT	K + $\Diamond\varphi \rightarrow \Box\varphi$	<i>Frames</i> com Unicidade
T	K + $\Box\varphi \rightarrow \varphi$	Reflexiva
B	K + $\Diamond\Box\varphi \rightarrow \varphi$	Simétrica
4	K + $\Box\varphi \rightarrow \Box\Box\varphi$	Transitiva
5	K + $\Diamond\varphi \rightarrow \Box\Diamond\varphi$	Euclideana
S4	T + $\Box\varphi \rightarrow \Box\Box\varphi$	Preordenada (reflexiva e transitiva)
S5	S4 + $\Diamond\Box\varphi \rightarrow \varphi$	Equivalência

Fonte: Elaborada pelo autor (2022)

⁷ Também conhecido por Axioma de Aristóteles ou axioma de Kripke (BAŞKENT, 2010)

É certo que nem todas as modalidades (ou, neste caso, os sistemas modais) recebam uma palavra distinta. Os sistemas de Lewis (S1, ..., S5, não trabalhados neste texto) são um exemplo disso. Em Carnielli e Pizzi (2008) é possível constatar um diagrama (24), cujas setas indicam inclusão própria, de uma hierarquização de sistemas modais normais por poder de expressividade. Portanto, como se trata de inclusão própria, o **K** das siglas dos sistemas poderia ser omitido e o **KD**, por exemplo, ser lido como apenas **D**. Confira na Figura 25 um gráfico mais granular de hierarquia dos sistemas modais, esmiuçado da seguinte forma: na vertical, os sistemas modais são organizados por nível de reflexividade, ou seja, a menos que o sistema em questão contenha o sistema **T** (que introduz o axioma da reflexividade), eles serão "menos reflexivos" na hierarquia. O mesmo serve para a diagonal organizando por transitividade e a horizontal organizando a simetria. É necessário interpretar a leitura desta figura como orientado para o diagrama na figura 24. Para aprofundar a compreensão dos diferentes sistemas, o trabalho de Goré (1999) conta com uma lista maior deles.

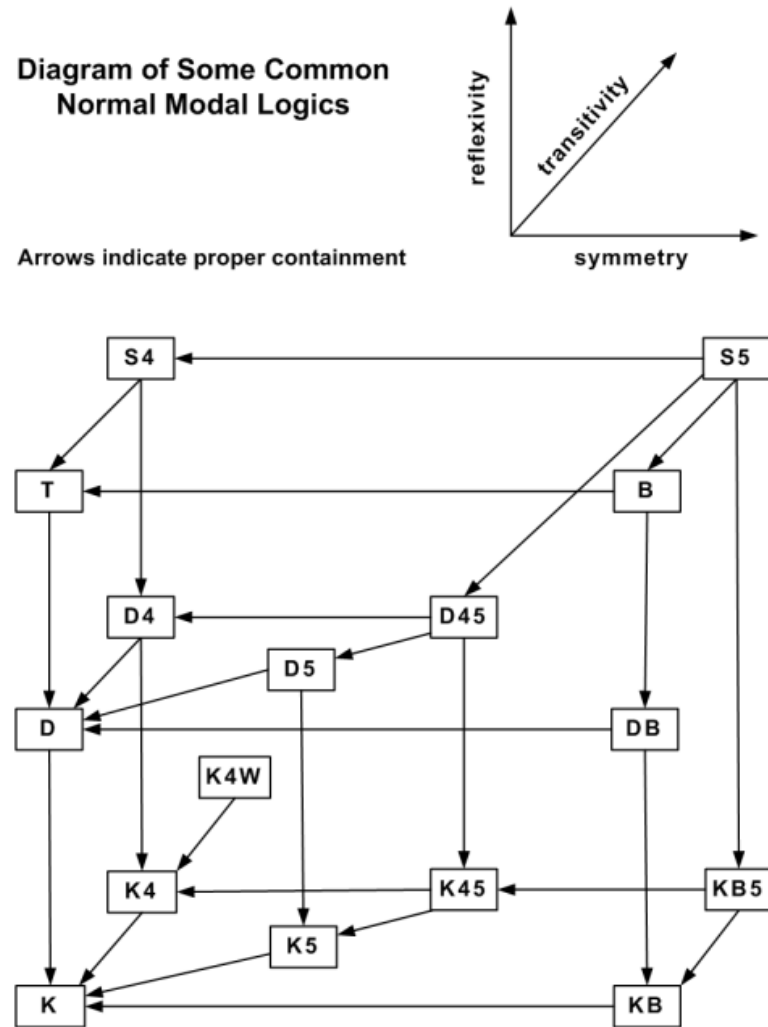
Figura 24 – Diagrama que organiza algumas lógicas modais normais por inclusão própria



Fonte: (CARNIELLI; PIZZI, 2008)

A Figura 25 destaca a hierarquia dos sistemas modais normais conforme mencionado anteriormente no texto.

Figura 25 – Diagrama que organiza lógicas modais normais



Fonte: (CC BY-SA 4.0, s.d.)

Modalidade Tensa

O nome que recebeu esta modalidade foi cunhado por Arthur Prior na década de 1960 (GORANKO; RUMBERG, 2021). Esta modalidade nasceu com o intuito de raciocinar sobre tempo e informação temporal. Os operadores clássicos são P , F , H , G , "certa vez no passado", "em algum momento futuro", "sempre no passado" e "sempre no futuro" respectivamente. Nota-se já a dualidade modal de necessidade e possibilidade. De fato:

- $P\varphi \equiv \neg H\neg\varphi$
- $H\varphi \equiv \neg P\neg\varphi$
- $F\varphi \equiv \neg G\neg\varphi$
- $G\varphi \equiv \neg F\neg\varphi$

O axioma **T** recebe esta sigla devido a esta modalidade. É possível notar diretamente que se necessariamente φ ocorre ($\Box\varphi$), tem-se φ como um fato ($\Box\varphi \rightarrow \varphi$).

Assim como na Definição 18, as equivalências definidas aqui têm sabor semelhante. É importante notar que as noções de "passado" e "futuro" não se entrelaçam: todas as equivalências são idênticas à dos quantificadores universais da LPO, entretanto concorrem apenas P e H na mesma equivalência, ou F e G , que denotam, respectivamente, o raciocínio sobre o passado e o raciocínio sobre o futuro.

Modalidade Deontica

O axioma **D** é consequência de **T** (CARNIELLI; PIZZI, 2008, p. 30). Ele remete à noção de que tudo aquilo que é obrigatório é permitido. Em McNamara e Putte (2021), é possível entender as equivalências modais dos operadores que compõem esta modalidade. São eles "obrigatório"(**O**), "permitido"(**A**), "proibido"(**P**), "omissível"(**S**), "opcional"(**F**) e "não-opcional"(**N**). Eles se organizam desta forma:

1. $A\varphi \equiv \neg O\neg\varphi$
2. $P\varphi \equiv O\neg\varphi$
3. $S\varphi \equiv \neg O\varphi$
4. $F\varphi \equiv (\neg O\neg\varphi \wedge \neg O\varphi)$
5. $N\varphi \equiv (O\varphi \vee O\neg\varphi)$

O operador **O** é tido como primitivo e, a partir dele, são definidos os restantes. Além disso, estas equivalências podem ser confusas a princípio, contudo não requerem mais que uma escrita por extenso dos operadores. Perceba: sendo φ um afazer, se o contrário deste afazer (sua negação) não for obrigatória, então ele próprio é permitido e vice-versa, como lista a primeira definição; a segunda rende uma leitura direta tanto quanto a terceira: o que é proibido, é obrigatório que não se faça (e vice-versa) e se não é obrigatório, é omissível. Esta noção de omissão é mais fraca que a noção de algo ser facultativo, como lista a quarta equivalência: se um afazer é opcional, ele precisa ser tanto permitido, quanto omissível. A última equivalência também tem compreensão direta: se um afazer é não-opcional, qualquer que seja sua natureza, ela será obrigatória.

Em Liu (2011, p. 141), é possível entender mais profundamente o estudo da lógica Deôntica, que "estuda conceitos normativos como obrigação, proibição, permissão e comprometimento", numa semântica baseada em preferência. Hansson (1969) é reconhecido como o trabalho inicial sobre o raciocínio deôntico.

Modalidade Epistêmica

Um estudo sobre a linguagem epistêmica pode ser encontrado na dissertação de Holliday (2012), que define (a partir da página 15) a tal linguagem, o grau modal e usa um operador chamado K . Ele é o equivalente ao clássico $\Box$. O motivo da escolha desta letra e não qualquer outro símbolo é simples: é a inicial de conhecimento em inglês (*Knowledge*). Esta é a modalidade que lida com a noção de conhecimento e crença, ou seja, de saber algo, algum determinado fato, ou acreditar nisso. O equivalente ao operador $\Diamond$ é o B (*Belief*).

Exemplo 22. Considere a situação em que se chega em casa depois de participar de uma palestra e vê que uma janela está quebrada, a porta está aberta, as coisas do armário estão espalhadas pelo chão e algo está faltando. Deduz-se que se fora roubado. Se um policial perguntar se há algo mais que possa ter acontecido e responde-se com razão "não", não há outra possibilidade de que um ladrão entrou na casa e roubou algum pertence, isso deve ser o que aconteceu, não pode ser de outra forma.

Na lógica Epistêmica, a sentença $K_i \phi$ representa "o agente i sabe que ϕ é o caso".

Exemplo 23. Repare a sentença a seguir: $K_i \phi \vee K_i \neg \phi$

Ela representa a noção de que o agente i sabe se ϕ é o caso, ou seja, tem inteira noção da natureza assertiva de ϕ .

No trabalho de Mendes et al. (2016), em que é construída uma inteligência artificial para Processamento de Linguagem Natural (do inglês, *Natural Language Processing*, NLP), a Tabela 1 (pág. 11) lista a taxa de frequência que as sentenças em linguagem natural eram classificadas conforme as modalidades.

Reescrita em LPO dos axiomas modais

Cada modalidade representa uma aplicação específica, uma necessidade que surgiu e foi concebido um modelo matemático de raciocínio. Todos os axiomas e, assim, todos os sistemas podem ser reescritos pela Lógica de Primeira Ordem, ou, como em Bull e Segerberg (2001),

são as condições em $\mathfrak{R}$, ou seja, a propriedade da relação de acessibilidade que caracteriza a estrutura. A tradução tanto pode seguir a Definição 34, como também podem ser listados em equivalência sob os conceitos já conhecidos, como por exemplo o axioma que define o sistema **T**, que reflete a noção de reflexividade. Repare no Exemplo 12, se o mundo x_2 não acessasse o mundo x_3 , o axioma **T** seria o único satisfatível nele.

Definição 34. (Tradução Modal para LPO): Sejam os elementos do conjunto $\mathcal{W}$ definidos como equivalentes às variáveis em LPO, as proposições na PML abstraídas na LPO e a relação de acessibilidade, implícita os operadores modais, equivalente à relação R da LPO. A tradução Π de uma sentença na Lógica Modal para sua equivalente na Lógica de Primeira Ordem, escrita com índice " x "(Π_x) quando avaliada no mundo atual x , é definida indutivamente como:

- $\Pi_x(p) \equiv P(x) \mid p$ atômico, $P(x)$ é verdade quando $x \models p$;
- $\Pi_x(\Diamond\varphi) \equiv \exists y(R(x, y) \wedge \Pi_y(\varphi))$;
- $\Pi_x(\Box\varphi) \equiv \forall y(R(x, y) \rightarrow \Pi_y(\varphi))$.

Seguindo esta definição (34), repare o Exemplo 24 a seguir:

Exemplo 24. A tradução da fórmula $\Box\Box p$ é dada por:

$$\begin{aligned}\Box\Box p &\equiv \forall y(R(x, y) \rightarrow \Pi_y(\Box p)) \\ &\quad (\forall z(R(y, z) \rightarrow P(z))) \\ \Box\Box p &\equiv \forall y\forall z(R(x, y) \rightarrow ((R(y, z) \rightarrow P(z))))\end{aligned}$$

Os axiomas apresentados adiante são reescritos em LPO conforme sua intuição, não seguindo estritamente a Definição 34, mas podendo ser deduzidos dela juntamente com um pouco de álgebra relacional.

(T): Axioma da Reflexividade:

$$\Box p \rightarrow p \equiv \forall x(xRx);$$

Este axioma, como mencionado no texto, é trivialmente visualizado através do Exemplo 12. A Definição 34 garante que:

1. Todo x deduz p ;
2. $\forall y(R(x, y) \rightarrow (P(y) \rightarrow P(y)))$;
3. Ou não há relação do mundo x para y , ou avaliar p num y implica em avaliá-lo no mesmo y , garantindo a intuição da reflexividade.

(D): Axioma da Serialidade:

$$\Box p \rightarrow \Diamond p \equiv \forall x \exists y (xRy);$$

(CD): Axioma da Unicidade:

$$\Diamond p \rightarrow \Box p \equiv \forall x \forall y \forall z (xRy \wedge xRz \rightarrow y = z);$$

(B): Axioma da Simetria:

$$p \rightarrow \Box \Diamond p \equiv \forall x \forall y (xRy \rightarrow yRx);$$

Este axioma garante a intuição da simetria, pois, pela definição semântica dos operadores (17), o operador $\Box$ não garante a existência do mundo possível, da relação que leva até ele, mas o $\Diamond$ sim. Combinados, avaliar p no mundo atual é satisfatório apenas se este mundo tiver uma relação de simetria com outro qualquer (confira os mundos x_2 e x_3 do Exemplo 12).

(4): Axioma da Transitividade:

$$\Box p \rightarrow \Box \Box p \equiv \forall x \forall y \forall z (xRy \wedge yRz \rightarrow xRz);$$

Seguindo da explicação do primeiro axioma, bem como do Exemplo 24, é direta a compreensão desta equivalência.

(5): Axioma da Euclidianidade:

$$\Diamond p \rightarrow \Box \Diamond p \equiv \forall x \forall y \forall z (xRy \wedge xRz \rightarrow yRz);$$

(**GL**): Axioma de Gödel-Löb:

$$\Box(\Box p \rightarrow p) \rightarrow \Box p \equiv \forall x \forall y \exists z (x \neq y \wedge xRy \rightarrow (z \neq x \wedge z \neq y \wedge (xRy \wedge zRy))).$$

Outros nomes de autores homenageados nos sistemas que conceberam merecem ser citados também, todos listados em Bull e Segerberg (2001) e as lógicas que representam.

McKinsey **M**: $\Box \Diamond p \rightarrow \Diamond \Box p$;

Geach **G**: $\Diamond \Box p \rightarrow \Box \Diamond p$;

Hintikka **H**: $(\Diamond p \wedge \Diamond q) \rightarrow (\Diamond(p \wedge q) \vee \Diamond(p \wedge \Diamond q) \vee \Diamond(q \wedge \Diamond p))$;

Grzegorzczak **Grz**: $\Box(\Box(p \rightarrow \Box p) \rightarrow p) \rightarrow p$;

Dummett **Dum**: $\Box(\Box(p \rightarrow \Box p) \rightarrow p) \rightarrow (\Diamond \Box p \rightarrow p)$.

APÊNDICE B – MAPEAMENTO ENTRE EXTENSÕES MAIS RICAS

LÓGICA DINÂMICA

Ao contrário das lógicas de paradigma estático, o foco da Lógica Dinâmica (do inglês, *Dynamic Logic*, aqui MDL, para referenciar a natureza modal e diferenciar da Lógica de Descrições) reside na noção de mudança, como o nome sugere. Com aplicações em ciência da computação, teoria dos jogos, linguística computacional, inteligência artificial e filosofia formal, MDL se preocupa com como o conhecimento muda, como ações produzem certas consequências. O que podem significar "ações"? No contexto da MDL, ações podem ser entendidas como computação, ou como estratégia de jogo, ou ainda como atualizações epistêmicas. Eis a distinção clara das MDL para as lógicas clássicas: em MDL os valores verdade podem mudar, daí a noção de lógica da mudança, em contrapartida à propriedade estática destes valores verdade nas lógicas clássicas (HAREL; KOZEN; TIURYN, 2000).

Lógica Dinâmica aqui neste trabalho é uma linguagem formal que permite raciocinar sobre programas. Alternativamente, como em Troquard e Balbiani (2019), "Lógicas Dinâmicas são lógicas modais que representam estados e eventos de sistemas dinâmicos." Existem duas abordagens para a lógica modal de programas: a abordagem endógena (GABBAY; GUENTHNER, 2005) e a abordagem exógena, que teve como precursor o cálculo de Hoare (1969 apud GABBAY; GUENTHNER, 2005, p. 111), de onde a semântica é importada e deriva a semântica de programas (FISCHER; LADNER, 1979). A MDL aplica os conceitos da Lógica Modal na semântica relacional de programas para entregar sistemas de raciocínio sobre os comportamentos antes e depois de programas. A LPO é inadequada para formalizar o raciocínio sobre programas (STREETT, 1985). Conceitos como terminação requerem lógicas estritamente mais expressivas que de primeira ordem (KFOURY; PARK, 1975 apud STREETT, 1985, p. 359). Construções análogas à necessidade e possibilidade serão estudadas adiante.

Pratt trouxe em 1976 a primeira versão da Lógica Dinâmica, uma "lógica modal de primeira ordem" (PRATT, V., 1976) e, em 1979, Fischer e Ladner (1979) entregaram a Lógica Dinâmica Proposicional. A diferença é que, nas fórmulas de Pratt, quantificação e variáveis de primeira ordem são permitidas, enquanto para a PDL de Fischer e Ladner somente o são as proposicionais. Raciocinar sobre programas significava formalizar especificações de corretude que deveriam ser alcançadas por eles. Além disso, determinar a equivalência de programas, comparar o poder de expressividade de construtores para eles e sintetizá-los a partir das especificações

fazia parte da proposta.

A nomenclatura intuitivamente descreve a característica principal desta linguagem formal: a lógica de predicados clássica é estática, ou seja, o valor de uma fórmula é determinado pela valoração de suas variáveis sobre uma estrutura, enquanto em MDL existem operadores sintáticos, os chamados programas, que operam num determinado domínio de computação, cujo papel principal é mudar o valor das variáveis, mudando assim o valor das fórmulas. MDL pode ser entendida como uma mistura de LPO, PML e a álgebra de eventos regulares ou LTS (*Labeled Transition Systems*) como em Troquard e Balbiani (2019). LTSs podem ser entendidos como a generalização dos modelos de Kripke, em que a transição entre os mundos possíveis, aqui sendo chamados de estados, é rotulada como um "programa atômico". Em Harel, Kozen e Tiuryn (2000, p. 163), a nomenclatura intercambia de programa atômico para "programa primitivo", que são interpretados como relações binárias arbitrárias num conjunto arbitrário K de estados. Do mesmo modo, "asserções primitivas" são apenas proposições atômicas e são interpretadas como subconjuntos arbitrários de K (HAREL; KOZEN; TIURYN, 2000). Uma transição π entre estados w e v indica que existe uma execução possível de π que parte de w e para em v . Caso uma proposição p que seja avaliada em v seja verdade, então a fórmula $\langle \pi \rangle$ é verdade em w .

São várias as versões da MDL. Especificamente para LST, em 1980, Matthew Hennessy e Robin Milner introduzem o que ficaria posteriormente conhecida como a lógica de Hennessy-Milner (do inglês, Hennessy-Milner *logic*, HML). Ela é usada para especificar as propriedades dos LTS, cuja estrutura é similar a um autômato (HENNESSY; MILNER, 1980).

Em MDL, os programas, podem ser entendidos como uma receita em linguagem formal que computa dados. Programas usam variáveis para guardar valores de entrada e saída, além dos intermediários durante a execução. Estes valores podem ser acessados a qualquer instante, sendo um "estado" uma função de valoração de variáveis que muda seus valores, ou seja, que atribui um valor a cada variável de programa. Assim, a partir de um instante arbitrário, o programa deve "estar" num estado de execução determinado pelo valor instantâneo de todas as variáveis. A sequência de estados que ocorrem da execução de um programa dado um estado inicial, ou um valor de entrada, até um estado terminal que resulta num valor de saída é chamada de rastro ou, do inglês, *trace*. O conjunto de todos os pares de estados inicial e final dos *traces* é chamado de relação de entrada/saída. Em geral, programas são construídos indutivamente a partir de programas atômicos e testes são construídos usando operadores de programa (HAREL; KOZEN; TIURYN, 2000).

Construtores de Programas

Os construtores de programas são operadores na lógica que carregam a intuição de raciocinarem sobre este domínio, todos programas algorítmicos.

Programas Regulares

Dado um conjunto de programas e testes, os programas regulares da MDL guardam muita semelhança com expressões regulares e podem ser definidos como a seguir. Posteriormente, os testes serão entendidos em PDL como proposições.

- Qualquer programa atômico é um programa;

Os programas chamados atômicos executam num passo apenas e não podem, como se deve esperar, ser decompostos.

- Se φ é um teste, então $\varphi?$ é um programa;

O operador $?$ testa se a propriedade φ é válida no estado corrente. Continua a execução sem mudar de estado caso φ seja verdade, caso contrário realiza um bloqueio, ou *block*, do inglês. Definido pela constante 0, o bloqueio empata a execução do *trace* sem terminar. Em Fischer e Ladner (1979) para o *block* se usa o θ .

- Se α e β são programas, então $\alpha ; \beta$ é um programa;

O operador $;$ é a composição sequencial. Significa que depois que executar α , execute β .

- Se α e β são programas, então $\alpha \cup \beta$ é um programa;

O operador $\cup$ é a escolha não-determinística. Significa "escolha não-deterministicamente entre α e β e execute.

- Se α é um programa, então α^* é um programa;

O operador $*$ é o operador de iteração. Significa "execute α um número não-determinístico finito de vezes."

Mais sobre esse texto pode ser conferido em Harel, Kozen e Tiuryn (2000). As operações com $*$, $\cup$ e $;$ são típicas da teoria de linguagens formais e autômatos. A título de lembrança, os leitores são convidados a consultar as operações com $*$ no apêndice D.

Programas *While*

A abstração das linguagens imperativas tradicionais pode ser vista como a família dos construtores *while* em MDL. Dependendo da escolha dos tipos de teste permitidos ou da inclusão de não-determinismo, é concebida uma versão ou outra. Os programas *while* são definidos indutivamente a partir de testes e programas atômicos.

Os operadores de programas *while* determinísticos podem ser definidos como a seguir:

1. se φ então α senão $\beta \stackrel{def}{=} \varphi? ; \alpha \cup \neg\varphi? ; \beta$
2. while φ faça $\alpha \stackrel{def}{=} (\varphi? ; \alpha^*) ; \neg\varphi?$

Recursão

O conhecido artifício da recursão é trivialmente demonstrado ser equivalentemente expressivo ao da iteração sobre os números naturais. De fato, os programas **while** correspondem ao que é chamado de recursão final, ou *tail recursion* no inglês, e é por isso que sobre certos domínios, os programas **while** são estritamente inferiores.

Uma sequência finita de computação é uma string finita de programas e testes que termina. Abreviada como seq, é válido notar que é um conceito distinto de *trace*. Ainda é válido notar também que pode existir uma seq que sequer executa, sob qualquer interpretação, mantendo a característica de ser uma seq. O conjunto de seqs de um programa α pode ser denotado por $CS(\alpha)$. A linguagem mais geral no contexto de MDL para programas, que subsume (classifica) todas as demais concebidas pelos construtores mencionados em termos de poder de expressividade, seria a classe de programas recursivamente enumeráveis, ou um conjunto recursivamente enumerável de seqs. Os programas **while** formam os conjuntos regulares de seqs e os recursivos, os conjuntos livres-de-contexto (HAREL; KOZEN; TIURYN, 2000).

PDL

A versão proposicional da Lógica Dinâmica tem como característica principal abstrair a natureza de um domínio de computação. Ela estuda a interação pura das assertivas (proposições) com programas. Tendo isto em mente, a noção de atribuição de variável deve ser abstraída. Ademais, programas podem ser entendidos como relações binárias num conjunto abstrato de estados.

Dentro da PDL, existem várias versões, dependendo dos operadores que compõem a sintaxe. A versão apresentada a seguir é conhecida na literatura como PDL regular, sendo esta a versão mais básica.

Sintaxe

Respeitando a notação simbólica de Harel, Kozen e Tiuryn (2000), a sintaxe da PDL regular consiste em expressões de dois tipos: símbolos proposicionais ($\varphi, \psi, \dots$), que representam as assertivas e programas ($\alpha, \beta, \dots$), que representam relações binárias de transição entre estados. O conjunto de proposições atômicas $\{p, q, r, \dots\}$ é denotado por φ_0 e o conjunto de programas atômicos $\{a, b, c, \dots\}$ é denotado por π_0 . É ideal que, tendo os leitores interesse de aprofundar PDL, comece a consulta por Segerberg (1982).

Definição 35. (Alfabeto da Lógica Dinâmica Proposicional): O alfabeto da PDL é composto pela união dos seguintes conjuntos de símbolos:

1. O operador de implicação material $\rightarrow$,
2. As constantes booleana **0**, **1**,
3. O operador de composição sequencial $;$,
4. O operador de escolha não-determinística $\cup$,
5. O operador de iteração $*$,
6. O operador de necessidade modal $[]$ (equivalente ao $\Box$),
7. O operador de teste $?$,
8. O conjunto Φ_0 de proposições atômicas e
9. O conjunto π_0 de programas atômicos.

Tais que:

Definição 36. (Os conjuntos φ e π): Os conjuntos φ e π da PDL são formados indutivamente entrelaçados⁸ segundo as regras a seguir:

⁸ Os programas e as fórmulas definidas indutivamente estão entrelaçadas e não podem ser concebidas separadamente. A definição das proposições depende da definição de programas por causa do construtor $[\alpha]\varphi$ e a definição de programas depende da definição de proposições por causa do construtor φ (HAREL; KOZEN; TIURYN, 2000, p. 165).

- $\Phi_0 \subseteq \Phi$;
- $\pi_0 \subseteq \pi$;
- Se $\varphi, \psi \in \Phi$, então $\varphi \rightarrow \psi \in \Phi$ e $\mathbf{0} \in \Phi$;
- Se $\alpha, \beta \in \pi$, então $\alpha;\beta, \alpha \cup \beta$ e $\alpha^* \in \pi$;
- Se $\alpha \in \pi$ e $\varphi \in \Phi$, então $[\alpha]\varphi \in \Phi$;
- Se $\varphi \in \Phi$, então $\varphi? \in \pi$.

Por motivos de descarga notacional, os símbolos de ‘(’ e ‘)’ são omitidos de modo que os operadores unários têm precedência sobre os binários e o ; tem precedência sobre o $\cup$.

Exemplo 25. As seguintes expressões são equivalentes:

$$[\alpha; \beta * \cup \gamma^*]\varphi \vee \psi \equiv ([(\alpha; (\beta^*)) \cup (\gamma^*)]\varphi) \vee \psi$$

Os demais símbolos, tanto da Lógica Proposicional Clássica, quanto da Lógica Modal, podem ser deduzidos a partir dos símbolos definidos.

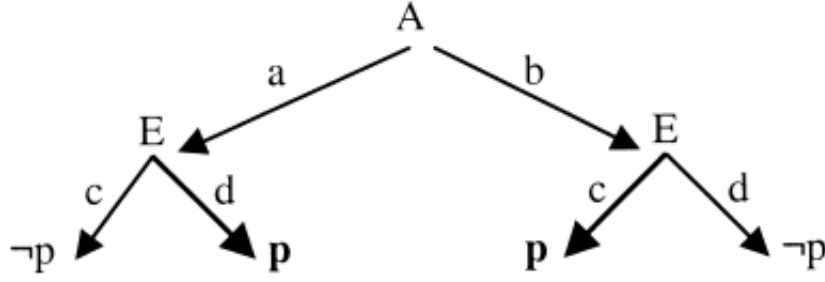
Exemplo 26. A dualidade dos operadores modais necessidade e possibilidade é preservada:

$$\langle \alpha \rangle \varphi \equiv \neg[\alpha]\neg\varphi$$

É válido notar, contudo, uma diferença importante entre $\langle \rangle$ e $[]$ na PDL, que, pela própria definição modal de necessidade e possibilidade, $\langle \alpha \rangle \varphi$ implica que α termina, enquanto que $[\alpha]\varphi$ não. Intuitivamente, $[\alpha]\varphi$ pode ser lido como "é necessário que depois que execute α , φ seja verdade" da mesma forma que $\langle \alpha \rangle \varphi$ pode ser lido como "há uma execução de α tal que φ é verdade após ela".

Exemplo 27. Imagine um jogo cujas ações sejam movimentos disponíveis para os jogadores. Cada estado é representado pelo nome do jogador cujo movimento está disponível atualmente. O jogador E necessita que o resultado satisfaça p contra qualquer jogada do jogador A. Assim, sua estratégia tem o formato da fórmula: $[a \cup b]\langle c \cup d \rangle p$.

Figura 26 – Árvore binária que ilustra o Exemplo 27



Fonte: Elaborada pelo autor (2022)

Exemplo 28. Seja $\mathbb{N}$ o conjunto dos naturais e V um conjunto de variáveis de primeira ordem.

$\varphi 0 = \{x = y, x = 0, x = y + 1, x = y - 1, x = y + z \mid x, y, z \in V\}$, $\pi 0 = \{x := \text{aleatório}, x := y + 1, x := y - 1 \mid x, y \in V\}$.

Considere o modelo $\mathcal{A} = \langle S, \pi, \varphi \rangle$, semelhante ao modelo de Kripke apresentado anteriormente (17), após simples troca de símbolos, mas com o mesmo valor semântico:

$S = V \rightarrow \mathbb{N}$ (o conjunto de atribuições de variáveis),

$\pi(x = y) = \{s \mid s(x) = s(y)\}$,

$\pi(x = 0) = \{s \mid s(x) = 0\}$,

$\pi(x = y + 1) = \{s \mid s(x) = s(y) + 1\}$,

$\pi(x = y - 1) = \{s \mid s(x) = s(y) - 1\}$,

$\pi(x = y + z) = \{s \mid s(x) = s(y) + s(z)\}$,

$\varphi(x := \text{aleatório}) = \{(s, t) \mid s(y) = t(y) \ \forall y \neq x\}$,

$\varphi(x := y + 1) = \{(s, t) \mid t(x) = s(y) + 1 \wedge t(z) = s(z) \ \forall z \neq x\}$,

$\varphi(x := y - 1) = \{(s, t) \mid t(x) = s(y) - 1 \wedge t(z) = s(z) \ \forall z \neq x\}$.

O programa " $x := \text{aleatório}$ " pode ser entendido como $\exists x$, ou seja, "atribua qualquer valor a x não-deterministicamente".

Sejam:

$$p \stackrel{def}{=} (w = x \wedge z = y) \rightarrow [a](z = x + y)$$

e

$$a \stackrel{def}{=} (\neg w = 0?; (w := w - 1); (z := z + 1))^*; (w = 0?).$$

A proposição p declara que o programa a computa corretamente a adição. É trivial verificar que $s \models p$ para qualquer $s \in S$, $\mathcal{A}$ uma vez que se tenha a noção dos construtores, que foram mencionados anteriormente.

Axiomatização

Munidos da sintaxe, agora é adequado listar os axiomas da PDL. Segundo Troquard e Balbiani (2019), a axiomatização de PDL foi proposta por Segerberg em 1977. Para tanto, em sua natureza modal, PDL é uma modalidade normal e a menor que contém:

Definição 37. (Axiomas da PDL)

1. $[\alpha;\beta]\varphi \leftrightarrow [\alpha][\beta]\varphi$
2. $[\alpha \cup \beta]\varphi \leftrightarrow [\alpha]\varphi \wedge [\beta]\varphi$
3. $[\alpha^*]\varphi \leftrightarrow \varphi \wedge [\alpha][\alpha^*]\varphi$
4. $[\varphi?]\psi \leftrightarrow (\varphi \rightarrow \psi)$

E é fechada sob a regra de invariância de loop, sendo ela trivialmente dedutível dos axiomas listados:

$$\varphi \rightarrow [\alpha]\varphi \vdash \varphi \rightarrow [\alpha^*]\varphi.$$

$$[\alpha^*](\varphi \rightarrow [\alpha]\varphi) \rightarrow (\varphi \rightarrow [\alpha^*]\varphi) \text{ (Axioma da indução)}$$

O axioma da indução, também escrito como $\varphi \wedge [\alpha^*](\varphi \rightarrow [\alpha]\varphi) \rightarrow [\alpha^*]\varphi$ em Harel, Kozen e Tiuryn (2000), é o equivalente ao famoso Axioma da indução de Peano

$$\forall y(\Phi(0, y) \wedge \forall x(\Phi(x, y) \rightarrow \Phi(S(x), y)) \rightarrow \forall x\Phi(x, y)).$$

Ou, como em Harel, Kozen e Tiuryn (2000), $\varphi(0) \wedge \forall n (\varphi(n) \rightarrow \varphi(n + 1)) \rightarrow \forall n \varphi(n)$.

Em BAŞKENT (2010) é possível encontrar as nomenclaturas "axioma da união"(A2), "axioma da composição"(A1), "axioma do teste"(A4), "axioma do ponto fixo"(A3), contendo também o axioma de Aristóteles como "axioma de Kripke" e o último, o da indução, com mesmo nome.

Semântica

Graças a sua natureza modal, a semântica de PDL é oriunda da semântica modal tradicional de Kripke. A fim de facilitar a leitura e correlação, é admitida a notação de Harel,

Kozen e Tiuryn (2000). Além do mais, a mudança de notação facilitará o entendimento intuitivo das estruturas. Assim, seguindo da Definição 19, um *frame* de Kripke é definido para PDL como:

Definição 38. (Kripke *frame*): Um Kripke *frame* é um par da forma

$$\mathfrak{K} = (K, m_{\mathfrak{K}}),$$

tal que K é o conjunto de estados da forma $\{u, v, w\}$ e $m_{\mathfrak{K}}$ é chamada função de significado, que atribui um subconjunto de K a cada proposição e uma relação binária em K a cada programa. Para o caso atômico, tem-se:

$$\begin{aligned} m_{\mathfrak{K}}(p) &\subseteq K, & p &\in \Phi_0 \\ m_{\mathfrak{K}}(a) &\subseteq K \times K, & a &\in \Pi_0 \end{aligned}$$

de modo que é extensível para todo elemento de φ e π por indução.

A definição da semântica dos operadores proposicionais para $m_{\mathfrak{K}}$ segue da definição padrão da PML para ν_F . A seguir (39), são definidos os demais operadores típicos da PDL (HAREL; KOZEN; TIURYN, 2000).

Definição 39. (Semântica dos operadores de programa):

$$\begin{aligned} m_{\mathfrak{K}}(\neg\varphi) &\stackrel{def}{=} K - m_{\mathfrak{K}}(\varphi) \\ m_{\mathfrak{K}}(\alpha; \beta) &\stackrel{def}{=} m_{\mathfrak{K}}(\alpha) \circ m_{\mathfrak{K}}(\beta) \\ &= \{(u, v) | \exists w \in K, (u, w) \in m_{\mathfrak{K}}(\alpha) \wedge (w, v) \in m_{\mathfrak{K}}(\beta)\} \\ m_{\mathfrak{K}}(\alpha \cup \beta) &\stackrel{def}{=} m_{\mathfrak{K}}(\alpha) \cup m_{\mathfrak{K}}(\beta) \\ m_{\mathfrak{K}}(\alpha^*) &\stackrel{def}{=} m_{\mathfrak{K}}(\alpha)^* \\ &= \bigcup_{n \geq 0} m_{\mathfrak{K}}^n(\alpha) \\ m_{\mathfrak{K}}(\varphi?) &\stackrel{def}{=} \{(u, u) | u \in m_{\mathfrak{K}}(\varphi)\} \end{aligned}$$

Equivalentemente:

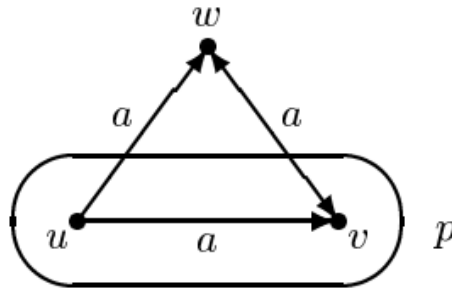
$$\begin{aligned} (u, v) \in m_{\mathfrak{K}}(\alpha \cup \beta) &\iff (u, v) \in m_{\mathfrak{K}}(\alpha) \vee (u, v) \in m_{\mathfrak{K}}(\beta) \\ (u, v) \in m_{\mathfrak{K}}(\alpha^*) &\iff \exists n \geq 0, \exists u_0, \dots, u_n | u = u_0, v = u_n \text{ e} \\ &\quad (u_i, u_{i+1}) \in m_{\mathfrak{K}}(\alpha), 0 \leq i \leq n-1 \\ (u, v) \in m_{\mathfrak{K}}(\varphi?) &\iff u = v \wedge u \models \varphi \end{aligned}$$

É válido notar que o primeiro símbolo⁹ é justamente o operador de iteração (antes da igualdade), enquanto o segundo (depois da igualdade) é o fecho reflexivo transitivo em relações binárias.

Exemplo 29. Sejam K , p e a tais que

$$\begin{aligned} K &= u, v, w \\ m_{\mathcal{R}}(p) &= u, v \\ m_{\mathcal{R}}(a) &= (u, v), (u, w), (v, w), (w, v) \end{aligned}$$

Figura 27 – Diagrama que ilustra o exemplo 29.



Fonte: (HAREL; KOZEN; TIURYN, 2000)

Nessa estrutura, tem-se que $u \models \langle a \rangle \neg p \wedge \langle a \rangle p$, $v \models [a] \neg p$ e $w \models [a] p$. A fórmula

$$\langle a^* \rangle [(aa)^*] \neg p \wedge \langle a \rangle [(aa)^*] p$$

é satisfatível para todo estado deste $\mathcal{R}$.

As regras de inferência para PDL, por sua natureza modal, seguem da PML, assim como as fórmulas que lá foram provadas válidas.

A seguir (teorema 2), são apresentadas fórmulas válidas em PDL, com seus operadores particulares. Diferente de Troquard e Balbiani (2019), aqui, segundo Harel, Kozen e Tiuryn (2000), as fórmulas definidas na Definição 37, tidas como axiomas, são apresentadas como teoremas e respectivamente provados.

⁹ Lembrando que, o fecho reflexivo transitivo de uma relação R é definido como o menor superconjunto de R que é uma relação transitiva e reflexiva. Por exemplo, se R contém elementos do tipo $(x, x+1)$ nos naturais, o fecho reflexivo FR deve conter elementos do tipo (x, x) , ou seja, $FR = \{(x, x+d) \mid d \in \{0, 1\}, x \in \{0, 1, 2, \dots\}\}$. Já para o fecho transitivo, tem-se que, se a relação contiver elementos da forma (x, y) e (y, z) , então (x, z) também pertence à relação. Por exemplo, se $(x, y) = (x, x+1)$ e $(y, z) = (y, y+1)$, então $(x, z) = (x, x+2)$ e, de uma forma geral, o fecho transitivo $FT = \{(x, x+d) \mid d \in \{1, 2, \dots\}, x \in \{0, 1, 2, \dots\}\}$. É, portanto, direto associar que o fecho reflexivo transitivo $*$ é da forma $* = \{(x, x+d) \mid d \in \{0, 1, 2, \dots\}, x \in \{0, 1, 2, \dots\}\}$, ou ainda $* = \{(x, y) \mid (y \geq x) \text{ e } (x, y) \in \{0, 1, 2, \dots\}\}$.

Teorema 2. (*Fórmulas válidas em PDL*):

1. $\langle \alpha \cup \beta \rangle \varphi \leftrightarrow \langle \alpha \rangle \varphi \vee \langle \beta \rangle \varphi$
2. $[\alpha \cup \beta] \varphi \leftrightarrow [\alpha] \varphi \wedge [\beta] \varphi$
3. $\langle \alpha; \beta \rangle \varphi \leftrightarrow \langle \alpha \rangle \langle \beta \rangle \varphi$
4. $[\alpha; \beta] \varphi \leftrightarrow [\alpha][\beta] \varphi$
5. $\langle \varphi? \rangle \psi \leftrightarrow (\varphi \wedge \psi)$
6. $[\varphi?] \psi \leftrightarrow (\varphi \rightarrow \psi)$

Prova: Para todo Kripke frame $\mathcal{K}$:

1. $m_{\mathcal{K}}(\langle \alpha \cup \beta \rangle \varphi) = m_{\mathcal{K}}(\langle \alpha \rangle \varphi \vee \langle \beta \rangle \varphi)$:

$$\begin{aligned} & m_{\mathcal{K}}(\langle \alpha \cup \beta \rangle \varphi) \\ &= (m_{\mathcal{K}}(\alpha) \cup m_{\mathcal{K}}(\beta)) \circ m_{\mathcal{K}}(\varphi) \text{ semântica de } \langle \rangle \\ &= ((m_{\mathcal{K}}(\alpha) \circ m_{\mathcal{K}}(\varphi)) \cup (m_{\mathcal{K}}(\beta) \circ m_{\mathcal{K}}(\varphi))) \text{ álgebra relacional} \\ &= m_{\mathcal{K}}(\langle \alpha \rangle \varphi) \cup m_{\mathcal{K}}(\langle \beta \rangle \varphi) \text{ semântica de } \langle \rangle \\ &= m_{\mathcal{K}}(\langle \alpha \rangle \varphi \vee \langle \beta \rangle \varphi) \text{ semântica da lógica proposicional} \end{aligned}$$
2. $m_{\mathcal{K}}([\alpha \cup \beta] \varphi) = m_{\mathcal{K}}([\alpha] \varphi \wedge [\beta] \varphi)$:

$$\begin{aligned} & m_{\mathcal{K}}([\alpha \cup \beta] \varphi) \\ &= K - (m_{\mathcal{K}}(\alpha \cup \beta) \circ (K - m_{\mathcal{K}}(\varphi))) \text{ semântica de } [] \\ &= m_{\mathcal{K}}(\neg(m_{\mathcal{K}}(\alpha \cup \beta)) \circ m_{\mathcal{K}}(\neg \varphi)) \text{ semântica de } \neg \\ &= m_{\mathcal{K}}((\neg \langle \alpha \cup \beta \rangle)(\neg \varphi)) \text{ semântica de } \langle \rangle \\ &= m_{\mathcal{K}}(\neg(\langle \alpha \rangle \neg \varphi \vee \langle \beta \rangle \neg \varphi)) \text{ passo i)} \\ &= m_{\mathcal{K}}(\neg \langle \alpha \rangle \neg \varphi \wedge \neg \langle \beta \rangle \neg \varphi) \text{ semântica da lógica proposicional} \\ &= m_{\mathcal{K}}([\alpha] \varphi \wedge [\beta] \varphi) \text{ equivalência modal} \end{aligned}$$
3. $m_{\mathcal{K}}(\langle \alpha; \beta \rangle \varphi) = m_{\mathcal{K}}(\langle \alpha \rangle \langle \beta \rangle \varphi)$:

$$\begin{aligned} & m_{\mathcal{K}}(\langle \alpha; \beta \rangle \varphi) \\ &= (m_{\mathcal{K}}(\alpha) ; m_{\mathcal{K}}(\beta)) \circ m_{\mathcal{K}}(\varphi) \text{ semântica de } \langle \rangle \\ &= (m_{\mathcal{K}}(\alpha) \circ m_{\mathcal{K}}(\beta)) \circ m_{\mathcal{K}}(\varphi) \text{ semântica de } ; \\ &= m_{\mathcal{K}}(\alpha) \circ (m_{\mathcal{K}}(\beta) \circ m_{\mathcal{K}}(\varphi)) \text{ álgebra relacional} \\ &= m_{\mathcal{K}}(\langle \alpha \rangle \langle \beta \rangle \varphi) \text{ semântica de } \langle \rangle \end{aligned}$$

4. Segue análogo ao iii) por dualidade.

5. $m_{\mathcal{R}}(<\varphi?>\psi) = m_{\mathcal{R}}(\varphi \wedge \psi)$:

$$\begin{aligned} & m_{\mathcal{R}}(<\varphi?>\psi) \\ &= \{(u, u) \mid u \in m_{\mathcal{R}}(\varphi)\} \circ m_{\mathcal{R}}(\psi) \text{ semântica de ?} \\ &= \{u \mid u \in m_{\mathcal{R}}(\varphi)\} \cap m_{\mathcal{R}}(\psi) \text{ álgebra relacional} \\ &= m_{\mathcal{R}}(\varphi) \cap m_{\mathcal{R}}(\psi) \text{ semântica modal} \\ &= m_{\mathcal{R}}(\varphi \wedge \psi) \text{ semântica da lógica proposicional} \end{aligned}$$

6. Segue análogo ao v) por dualidade.

□

Filtragem em PDL

Assim como na PML, por sua natureza modal, a filtragem é usada para prova de corretude e completude na PDL. Agora, mais que a generalidade da Lógica Modal, PDL conta com funções específicas para o mesmo propósito, em prol da eficiência e entendimento. (GABBAY, 1972) afirma que qualquer sistema que admite filtragem tem a propriedade de modelo finito (também em Bull e Segerberg (2001)[p. 39]), mas não vice-versa. Isto é, apesar do caráter infinitário de um conjunto, um sistema dedutivo com esta característica é completo, pois todo conjunto consistente de fórmulas tem um modelo (HAREL; KOZEN; TIURYN, 2000). O que Harel, Kozen e Tiuryn (2000) chama de "a propriedade de modelo pequeno" quer dizer que se uma fórmula φ é satisfatível, então ela o é num *frame* de Kripke em no máximo $2|\varphi|$ estados. Já existem algoritmos mais eficientes que o tradicional para o teste de satisfatibilidade e o autor demonstra um exemplo.

Num cenário em que a prova da filtragem na PML usaria o conjunto de subfórmulas, em PDL é usado o correspondente fecho de Fischer-Ladner de uma dada fórmula. A princípio, são definidas duas funções por indução simultânea¹⁰:

$$\mathcal{FL} : \Phi \rightarrow 2^{\Phi}$$

$$\mathcal{FL}_{\Box} : [\alpha]\varphi \mid \alpha \in \Psi, \varphi \in \Phi \rightarrow 2^{\Phi}$$

O conjunto $\mathcal{FL}(\varphi)$ é chamado de *fecho de Fischer-Ladner de φ* . A seguir, são definidas indutivamente as funções sobre os termos:

¹⁰ Aos leitores interessados, consultar (INDUCTION., s.d.)

1. $\mathcal{FL}(p) \stackrel{def}{=} \{p\}$, p sendo uma proposição atômica;
2. $\mathcal{FL}(\varphi \rightarrow \psi) \stackrel{def}{=} \{\varphi \rightarrow \psi\} \cup \mathcal{FL}(\varphi) \cup \mathcal{FL}(\psi)$;
3. $\mathcal{FL}(0) \stackrel{def}{=} \{0\}$;
4. $\mathcal{FL}([\alpha]\varphi) \stackrel{def}{=} \mathcal{FL}_\square([\alpha]\varphi) \cup \mathcal{FL}(\varphi)$;
5. $\mathcal{FL}_\square([a]\varphi) \stackrel{def}{=} \{[a]\varphi\}$, a sendo um programa atômico;
6. $\mathcal{FL}_\square([\alpha \cup \beta]\varphi) \stackrel{def}{=} \{[\alpha \cup \beta]\varphi\} \cup \mathcal{FL}_\square([\alpha]\varphi) \cup \mathcal{FL}_\square([\beta]\varphi)$;
7. $\mathcal{FL}_\square([\alpha; \beta]\varphi) \stackrel{def}{=} \{[\alpha; \beta]\varphi\} \cup \mathcal{FL}_\square([\alpha][\beta]\varphi) \cup \mathcal{FL}_\square([\beta]\varphi)$;
8. $\mathcal{FL}_\square([\alpha^*]\varphi) \stackrel{def}{=} \{[\alpha^*]\varphi\} \cup \mathcal{FL}_\square([\alpha][\alpha^*]\varphi)$;
9. $\mathcal{FL}_\square([\psi?]\varphi) \stackrel{def}{=} \{[\psi?]\varphi\} \cup \mathcal{FL}(\psi)$.

À primeira vista, pode parecer circular quando observada a regra 8, mas a função $\mathcal{FL}_\square$ tem justamente o propósito de evitar qualquer circularidade. Por ser definida somente para fórmulas do tipo $[\alpha]\varphi$, produz os elementos $\mathcal{FL}([\alpha]\varphi)$ quebrando α e ignorando φ .

As considerações feitas a seguir são lemas cujas provas podem ser conferidas em Harel, Kozen e Tiuryn (2000) e, por motivo de celeridade, não serão detalhadas.

Lema 1.

1. Se $\Sigma \in \mathcal{FL}(\varphi)$, então $\mathcal{FL}(\sigma) \subseteq \mathcal{FL}(\varphi)$.
2. Se $\sigma \in \mathcal{FL}_\square([\alpha]\varphi)$, então $\mathcal{FL}(\sigma) \subseteq \mathcal{FL}_\square([\alpha]\varphi) \cup \mathcal{FL}(\varphi)$.

As consequências das assertivas apresentadas são estas:

Lema 2.

1. Se $[\alpha]\psi \in \mathcal{FL}(\varphi)$, então $\psi \in \mathcal{FL}(\varphi)$;
2. Se $[\rho?]\psi \in \mathcal{FL}(\varphi)$, então $\rho \in \mathcal{FL}(\varphi)$;
3. Se $[\alpha \cup \beta]\psi \in \mathcal{FL}(\varphi)$, então $[\alpha]\psi \in \mathcal{FL}(\varphi)$ e $[\beta]\psi \in \mathcal{FL}(\varphi)$;
4. Se $[\alpha; \beta]\psi \in \mathcal{FL}(\varphi)$, então $[\alpha][\beta]\psi \in \mathcal{FL}(\varphi)$ e $[\beta]\psi \in \mathcal{FL}(\varphi)$;
5. Se $[\alpha^*]\psi \in \mathcal{FL}(\varphi)$, então $[\alpha][\alpha^*]\psi \in \mathcal{FL}(\varphi)$.

Posteriormente, são também lemas as expressões:

Lema 3.

1. Para qualquer fórmula φ , $\#\mathcal{FL}(\varphi) \leq |\varphi|$.
2. Para qualquer fórmula $[\alpha]\varphi$, $\#\mathcal{FL}_\square([\alpha]\varphi) \leq |\alpha|$.

Este lema limita a cardinalidade ($\#$) de $\mathcal{FL}(\varphi)$ como função do tamanho (número de símbolos) de φ .

Por fim, o lema da filtragem em PDL é apresentado e o teorema do modelo pequeno demonstrado como a seguir:

Dada uma proposição φ e um *frame* $\mathfrak{K} = (K, m_{\mathfrak{K}})$, é definido um novo *frame* da forma $\mathfrak{K}/\mathcal{FL}(\varphi) = (K/\mathcal{FL}(\varphi), m_{\mathfrak{K}/\mathcal{FL}(\varphi)})$, a filtragem de $\mathfrak{K}$ por $\mathcal{FL}(\varphi)$, como:

$$u \equiv v := \forall \psi \in \mathcal{FL}(\varphi) (u \in m_{\mathfrak{K}}(\psi) \leftrightarrow v \in m_{\mathfrak{K}}(\psi)).$$

Assim, os estados indistinguíveis por qualquer fórmula de $\mathcal{FL}(\varphi)$ u e v são colapsados.

$$\begin{aligned} u & \stackrel{def}{=} v | v \equiv u \\ K/\mathcal{FL}(\varphi) & \stackrel{def}{=} [u] | u \in K \\ \{\varphi\}\alpha\{\psi\} & \stackrel{def}{=} \varphi \rightarrow [\alpha]\psi \\ m_{\mathfrak{K}}/\mathcal{FL}(\varphi)(p) & \stackrel{def}{=} \{[u] | u \in m_{\mathfrak{K}}(p)\}, \text{ sendo } p \text{ uma proposição atômica.} \\ m_{\mathfrak{K}}/\mathcal{FL}(\varphi)(a) & \stackrel{def}{=} \{([u], [v]) | (u, v) \in m_{\mathfrak{K}}(a)\}, \text{ sendo } a \text{ um programa atômico.} \end{aligned}$$

Eis, portanto o lema da filtragem:

Lema 4. *Seja $\mathfrak{K}$ um frame e u, v estados de $\mathfrak{K}$.*

1. Para todos os $\psi \in \mathcal{FL}(\varphi)$, $u \in m_{\mathfrak{K}}(\psi)$ sse $[u] \in m_{\mathfrak{K}/\mathcal{FL}(\varphi)}(\psi)$.
2. Para todos os $[\alpha]\psi \in \mathcal{FL}(\varphi)$,
 - a) se $(u, v) \in m_{\mathfrak{K}}(\alpha)$, então $([u], [v]) \in m_{\mathfrak{K}/\mathcal{FL}(\varphi)}(\alpha)$;
 - b) se $([u], [v]) \in m_{\mathfrak{K}/\mathcal{FL}(\varphi)}(\alpha)$ e $u \in m_{\mathfrak{K}}([\alpha]\psi)$, então $v \in m_{\mathfrak{K}}(\psi)$.

Teorema 3. *(Teorema do modelo pequeno): Seja φ uma fórmula satisfatível de PDL. φ é satisfatível em não mais que $2^{|\varphi|}$ estados.*

Prova: Se φ é satisfatível, então existe um *frame* $\mathfrak{K}$ e um estado $u \in \mathfrak{K}$, em que $u \in m_{\mathfrak{K}}(\varphi)$. Seja $\mathcal{FL}(\varphi)$ o fecho de Fischer-Ladner de φ . Pelo lema da filtragem, $[u] \models_{\mathfrak{K}/\mathcal{FL}(\varphi)} \varphi$. Além do mais, $\mathfrak{K}/\mathcal{FL}(\varphi)$ não tem mais estados que o número de assertivas em $\mathcal{FL}(\varphi)$, que é no máximo $2^{|\varphi|}$ pelo Lema 3. $\square$

RAMIFICAÇÕES COM CORRESPONDÊNCIA: $\mathcal{ALC}_{\text{reg}}$, *CONVERSE-PDL* E μ -CÁLCULO

Nesta parte do trabalho, são apresentadas as ramificações das linguagens formais Modal Dinâmica e de Descrições que apresentam correspondência, especificamente PDL e enriquecimentos de $\mathcal{ALC}$, como encontrado na literatura.

DL

Nesta seção, a linguagem $\mathcal{ALC}_{\text{reg}}$, uma ramificação mais expressiva que $\mathcal{ALC}$, é construída em paralelo, com o intuito de melhorar a compreensão, a partir de um enriquecimento feito na linguagem $\mathcal{FL}_0$, uma ramificação estritamente menos expressiva que $\mathcal{ALC}$.

Linguagem $\mathcal{FL}_0$

Com o propósito de demonstrar como chegar à linguagem $\mathcal{ALC}_{\text{reg}}$, extensão de $\mathcal{ALC}$, trabalhada mais adiante no texto, equivalente a uma extensão da Lógica Modal Proposicional Dinâmica, é conveniente começar por uma linguagem mais simples em termos de expressividade, a $\mathcal{FL}_0$. Esta extensão de DL é enriquecida de tal maneira que se torna capaz de ponderar sobre ciclos terminológicos, problema presente nas Tboxes na execução do raciocínio de classificação (subsunção). As versões enriquecidas serão chamadas respectivamente por poder de expressividade $\mathcal{FL}_{\text{trans}}$ e $\mathcal{FL}_{\text{reg}}$.

Como escreve Baader et al. (2007), por razões históricas, a linguagem chamada $\mathcal{FL}_0$ (quadro, do inglês, *frame*) é tal que se permite apenas interseção de conceitos e restrição de conceitos por um papel, sem negação atômica, ou qualquer outro construto binário. Enriquecendo a linguagem, de maneira crescente, por permitir quantificação existencial limitada, obtém-se a linguagem $\mathcal{FL}^-$. Portanto, mais ou menos construtores, significa uma linguagem mais ou menos expressiva, com nomenclaturas distintas.

Sintaxe e Semântica

São análogas à $\mathcal{ALC}$ a sintaxe e semântica de $\mathcal{FL}_0$, somente permitindo menos fórmulas, sendo mais restritiva. Seguindo da Definição 6, as fórmulas bem-formadas para $\mathcal{FL}_0$ são tais que:

Definição 40. (Fórmulas bem-formadas $\mathcal{FL}_0$):

- Todo nome de conceito é uma descrição de conceito;
- $\top$ e $\perp$ são descrições de conceito;
- Se C e D são descrições de conceito e r é um nome de papel, então os seguintes também são descrições de conceito:

$C \sqcap D$ (conjunção),
 $\forall r.C$ (restrição de valor),
 $\forall r.\top, \forall r.\perp$.

Assim como a Definição 10, para $\mathcal{FL}_0$ segue:

Definição 41. (Interpretação):

$$\begin{aligned} \top^I &:= \Delta^I \\ \perp^I &:= \emptyset \\ (C \sqcap D)^I &:= C^I \cap D^I \\ \forall r.C &:= a \in \Delta^I \mid \forall b.(a, b) \in r^I \rightarrow b \in C^I \\ \exists r &:= a \in \Delta^I \mid \exists b.(a, b) \in r^I \end{aligned}$$

Raciocínio automático

Para o processo de raciocínio automático nesta linguagem mais simples que $\mathcal{ALC}$, haverá duas etapas. Na primeira, ocorrerá uma normalização, em que várias restrições universais aninhadas se transformarão em apenas uma restrição contendo uma cadeia dos papéis. Aqui, normalização é tratada dentro do raciocínio específico. Se trata de uma reorganização, um agrupamento de conjuntos e não diz respeito à normalização conhecida, com formas normais conjuntivas (ou disjuntivas). Na segunda, cada conceito será representado por uma cadeia e um conceito classificará outro quando a linguagem destas cadeias for gerada pelo superconceito, verificado através de autômatos.

Inspirado em Baader (1991), segue a caracterização para $\mathcal{FL}_0$. Para a normalização da Tbox T , qualquer conceito pode ser transformado numa conjunção infinita de termos da forma $\forall R_1. \forall R_2. \dots \forall R_n.A$. Agora, é possível abreviar o prefixo “ $\forall R_1. \forall R_2. \dots \forall R_n$ ” por simplesmente “ $\forall W$ ”, em que $W = R_1 R_2 \dots R_n$ é uma palavra sobre R_T , o conjunto de papéis em T . Para o caso em que $n = 0$, escreve-se “ $\forall \varepsilon : A$ ”. Numa interpretação I , W^I denota a composição de papéis $R_1^I \circ R_2^I \circ \dots \circ R_n^I$ ¹¹ de relações binárias $R_1^I R_2^I \dots R_n^I$ e o termo ε^I denota a relação de identidade (na literatura é possível encontrar *self* denotando identidade). Assim é possível associar um autômato finito A_T à Terminologia T de $\mathcal{FL}_0$.

Definição 42. (Autômato para $\mathcal{FL}_0$): O autômato não determinístico generalizado A_T é definido como: o alfabeto de A_T é o conjunto R_T de nomes de papéis ocorrendo em T ; os estados de A_T são os nomes de conceito ocorrendo em T ; um axioma terminológico da forma $A = \forall W_1.A_1 \sqcap \dots \sqcap \forall W_k.A_k$ dá origem a k transições, tais que a transição de A para A_i é rotulada pela palavra W_i .

O autômato é chamado generalizado por ter as transições rotuladas por palavras, não somente símbolos do alfabeto. Qualquer autômato finito generalizado pode ser transformado num autômato finito equivalente. Os leitores interessados podem conferir as definições de autômatos no apêndice D. Neste cenário, verificar uma classificação $C \sqsubseteq D$ significa testar se a linguagem de C está contida em D .

Exemplo 30. Considere a verificação de complexidade para a linguagem $\mathcal{FL}_0$: normalização (não permitir conjunções internas, juntar papéis e conceitos) e comparação estrutural.

$$D \equiv \forall R.(\forall S.B \sqcap \forall S.\forall R.A) \sqcap \forall R.(A \sqcap B)$$

$$C \equiv \forall R.(A \sqcap \forall S.(B \sqcap \forall R.A))$$

Passos:

$$\begin{aligned} 1. \quad D &\equiv \forall R.\forall S.B \sqcap \forall R.\forall S.\forall R.A \sqcap \forall R.A \sqcap \forall R.B \\ C &\equiv \forall R.A \sqcap \forall R.\forall S.B \sqcap \forall R.\forall S.\forall R.A \end{aligned}$$

$$\begin{aligned} 2. \quad D &\equiv \forall RS.B \sqcap \forall RSR.A \sqcap \forall R.A \sqcap \forall R.B \\ C &\equiv \forall R.A \sqcap \forall RS.B \sqcap \forall RSR.A \end{aligned}$$

¹¹ Apesar de composição ser melhor detalhada em PDL, eis um exemplo prático: imagine que g signifique calçar meias e f signifique calçar sapatos. Então $f \circ g$ significaria calçar meias depois sapatos, nesta ordem. Assim sendo, g^{-1} seria tirar as meias e f^{-1} tirar os sapatos, de modo que $(f \circ g)^{-1} = g^{-1} \circ f^{-1}$ significa o processo inverso, de descalçar completamente.

$$\begin{aligned}
3. \quad D &\equiv \forall\{RS, R\}.B \sqcap \forall\{RSR, R\}.A \\
C &\equiv \forall\{R, RSR\}.A \sqcap \forall\{RS\}.B
\end{aligned}$$

Agora, feita a normalização *ad hoc*, chega a etapa em que deve-se verificar a subsunção. Neste cenário, verificar uma classificação $C \sqsubseteq D$, como dito anteriormente, significa testar se a linguagem de C está contida em D .

$$4. C \sqsubseteq D \equiv \{R, RSR\} \sqsubseteq \{RSR, R\} \sqcap \{RS\} \sqsubseteq \{RS, R\}$$

Figura 28 – Representação gráfica dos autômatos correspondentes às linguagens mais expressivas, respectivamente $\{RS, R\}$ e $\{R, RSR\}$.



Fonte: Elaborada pelo autor (2022)

A verificação de uma subsunção é polinomial em tempo, enquanto testar se uma KB está em NP equivale a verificar se a linguagem dos autômatos equivalentes se relacionam por continência, ou $L(C) \subseteq L(D)$.

Exemplo 31. Sejam A , B e C conceitos definidos.

$$A \equiv \forall R.A \sqcap \forall S.C \sqcap P$$

$$B \equiv \forall RS.C \sqcap \forall S.Q$$

$$C \equiv \forall S.C \sqcap P$$

Um par de estados p, q de um autômato define uma linguagem regular $L(p, q)$, o conjunto de todas as palavras que são rótulos de caminhos de p até q . No exemplo, $L(A, P) = R^*S^*$ ou $R^nS^m; n, m \geq 0$, $L(B, P) = RSS^*$, $L(C, P) = S^*$, $L(B, Q) = S$ e $L(A, Q) = L(C, Q) = \emptyset$.

Ciclos terminológicos

Um serviço importante que a maioria dos sistemas de representação terminológica fornece é a computar a hierarquia de subsunção. Muitos dos algoritmos de subsunção existentes

para linguagens KR terminológicas não trabalham nas Tboxes (BAADER, 1991), mas nos conceitos que são termos. Por essa razão, eles têm que "desdobrar" a Tbox. O desdobramento de uma Tbox significa substituir conceitos definidos que ocorrem no lado direito de uma definição por seus termos definidores. Este processo deve ser iterado até que restem apenas conceitos primitivos do lado direito das definições. Obviamente, este procedimento termina se e somente se a terminologia for acíclica. Esta foi uma razão para proibir definições cíclicas. Uma terminologia está *desdobrada* sse o lado direito de seus axiomas contém apenas conceitos primitivos. Observe que o tamanho da Tbox desdobrada pode ser exponencial no tamanho Tbox original (NEBEL, 1990b).

Apesar do poder de expressividade de $\mathcal{ALC}$, para certas aplicações seria ideal possuir meios de expressar conceitos como fecho transitivo. Como a maioria das variações desta linguagem não vai além do escopo da LPO e o fecho transitivo em relações binárias não pode ser expresso nela (AHO; ULLMAN, 1979 apud BAADER, 1991, p. 3), há duas maneiras de superar o problema: pode-se adicionar um novo operador de formação de papéis e definir sua semântica como sendo sua interpretação sobre um papel R o fecho transitivo de R. Embora exista exemplo de linguagem que faz uso de um operador dessa natureza, ela não conta com um algoritmo para determinar relações de subsunção entre conceitos, uma etapa canônica nos sistemas de representação KL-ONE; igualmente seria possível adicionar definições de conceitos cíclicos com semântica de ponto-fixo para expressar restrição de valores com respeito ao fecho transitivo de papéis (BAADER, 1990). Mas, como já mencionado também, os algoritmos de inferência existentes não raciocinam ciclos, além de não ter uma semântica clara do ponto de vista teórico.

Exemplo 32. Sejam **Homem**, **Humano**, **Macho**, **Pai** e **PSF** (pai que tem somente filho) nomes de conceitos e **temCriança** um nome de papel. Considere a Tbox com os seguintes axiomas:

$$\text{Homem} \equiv \text{Humano} \sqcap \text{Macho}$$

$$\text{PSF} \equiv \text{Homem} \sqcap \forall \text{temCriança.Homem}$$

$$\text{Pai} \equiv \text{Homem} \sqcap \exists \text{temCriança.Homem}$$

Os dois primeiros axiomas são axiomas de $\mathcal{FL}_0$. Macho e Humano são conceitos primitivos e os demais (à esquerda) são conceitos definidos. Assuma, agora, que seja necessário expressar o conceito "homem que tem somente descendentes machos", MSDM. Seria inadequado adicionar um papel chamado "descendente", pois não teria conexão entre ele e *criança*. A intenção, evidentemente, é que "descendente" seja um fecho transitivo de *criança*. Eis, portanto, a definição cíclica MSDM: um homem que tem somente descendentes machos é tanto um homem

quanto todos os seus filhos são também homens que têm somente descendentes machos.

$$\text{MSDM} \equiv \text{Homem} \sqcap \forall \text{temCriança} . \text{MSDM} \quad (\text{B.1})$$

Como demonstração do processo de desdobramento, PSF pode claramente ser desdobrada através de uma substituição simples do conceito definido Homem, enquanto MSDM nunca chegará ao desdobramento final.

Exemplo 33. Imagine que seja preciso definir objetos como uma árvore binária. Sejam **Árvore** um conceito e **temRamo** um papel. Assim, árvores binárias são árvores com, no máximo, duas sub-árvores, estas também árvores binárias, portanto.

$$\text{ÁrvoreBi} \equiv \text{Árvore} \sqcap \leq 2 \text{ temRamo} \sqcap \forall \text{ temRamo} . \text{ÁrvoreBi}.$$

Ciclos são definidos em Baader, Horrocks e Sattler (2008) como sendo: sejam A, B conceitos atômicos ocorrendo numa terminologia T. A *usa diretamente* B em T se B aparece do lado direito da definição de A. Usa fica definido como o fecho transitivo da relação *usa diretamente*. Se há um conceito que usa a si mesmo dentro destas definições, há um ciclo em T. Há também uma discussão sobre terminologias *definidoras*. Nela, T é definidora se toda base de interpretação tem somente uma extensão que é modelo de T, ou seja, os significados dos símbolos estão completamente determinados. Em resumo, se a terminologia for acíclica, ela é definidora. Semânticas de ponto fixo são motivadas pela necessidade de definir situações em que intuitivamente ciclos são significativos. Esta intuição é capturada pela semântica de maior ou menor ponto fixo. No Exemplo 32, a semântica de maior ponto fixo captura a intuição do ciclo; já no Exemplo 33, ela é capturada pela de menor ponto fixo. Para uma brevíssima introdução sobre a teoria de ponto fixo, os leitores podem endereçar ao apêndice D.

Como já introduzido, ciclos são proibidos na maioria das linguagens de representação de conhecimento, como discorre Baader (1990), pelas razões já citadas. Ainda assim, axiomas recursivos têm um papel crucial em pesquisa de banco de dados, por exemplo. Aho e Ullman (1979) apud Baader (1990, p. 32 **m**) mostraram que o fecho transitivo de relações não pode ser expresso na LPO. Daí propuseram adicionar definições cíclicas, interpretadas pela semântica de menor ponto fixo. Este foi como o ponto de partida para os estudos sobre extensões de ponto fixo na LPO. Nebel (1990a) investiga ciclos em linguagens de representação de conhecimento, considerando as duas semânticas de ponto fixo, além da semântica chamada *descriptive*, específica

para ciclos em sua linguagem NTF¹².

Semântica de ponto fixo

Agora, é detalhada a semântica de ponto fixo, inspirado em Baader, Horrocks e Sattler (2008). Numa terminologia T , todo símbolo de nome A ocorre exatamente uma vez como o lado esquerdo de um axioma $A \equiv C$. Portanto, é possível enxergar T como um mapeamento que associa a descrição do conceito $T(A) = C$ a um símbolo de nome A . Com esta notação, uma interpretação I é um modelo de T se, e somente se $A^I = (T(A))^I$. Com a similaridade a uma equação de ponto fixo, pode-se introduzir uma família de mapeamentos tal que a interpretação seja um modelo de T sse é um ponto fixo de tal mapeamento.

Proposição B.0.1. *Sejam T e I uma terminologia e uma interpretação respectivamente. Se J é a restrição de I para os símbolos da base T , então I é modelo de T sse I é ponto fixo de T_J .*

Sendo J uma interpretação base fixa de T , denota-se Ext_J como sendo o conjunto de todas as extensões de J . Seja, então, $T_J : Ext_J \rightarrow Ext_J$ o mapeamento da extensão I à extensão $T_J(I)$ definida por $AT_J(I) = (T(A))^I$ para cada símbolo de nome A . Agora, I é um ponto fixo de T_J sse $I = T_J(I)$, ou seja, para todo nome de símbolo, $A^I = A_J^T(I)$. Sendo assim, obtém-se o resultado de que para toda definição $A \equiv C$ em T , tem-se que $A^I = A_J^{T(I)} = (T(A))^I = C^I$, o que prova que I é um modelo de T . De acordo com a proposição, a terminologia é definidora sse toda interpretação base J tem uma extensão única que é ponto fixo de T_J .

Com base nisto, é pertinente voltar ao Exemplo 33 sobre a terminologia T^{MSDM} e axioma (B.1). Considere a base de interpretação J definida por

$$\begin{aligned} \Delta^J &= \text{João}_1, \text{João}_2 \dots \cup \text{José}_1, \text{José}_2, \dots, \text{José}_{\text{Último}} \\ \text{Homem} &= \Delta^J \\ \text{TEMCRIANÇA} &= \text{João}_i, \text{João}_{(i+1)} | i \geq 1 \cup \text{José}_i, \text{José}_{(i+1)} | 1 \leq i < \text{Último}. \end{aligned}$$

Com esta configuração, entende-se que a dinastia de João não morre enquanto há um último membro da dinastia de José.

Para identificar os pontos fixos de T_J^{MSDM} , se I é uma extensão ponto fixo de J , então $\text{José}_{\text{Último}}$ está em $(\forall \text{temCriança.MSDM})^I$ e, portanto, em $MSDM^I$. É válido notar que um indivíduo, mesmo sem filhos, sem preenchimento do TEMCRIANÇA , está sempre na interpretação de $\forall \text{temCriança.MSDM}$, independentemente de como $MSDM$ é interpretado.

¹² Esta linguagem permite conjunção de conceitos, restrição de valores, restrição de número e negação de conceitos primitivos.

Conclui-se que José é um *MSDM*. Seja I_1 a extensão de J tal que $MSDM^{I_1}$ abrange exatamente toda a dinastia de José, I_1 é um exemplo de ponto fixo. Considerando I_2 como a extensão que interpreta *MSDM* como o domínio inteiro, obtém-se outro ponto fixo, e único distinto.

Menor e Maior ponto fixo

Para o caso da existência de múltiplos pontos fixos numa terminologia cíclica T , é necessário eleger um ponto fixo particular do mapeamento T_J . Com este fim, define-se uma ordem parcial $\preceq$ sobre as extensões de J . Para todo nome símbolo em T , $I \preceq I'$ se $A^I \preceq A^{I'}$. No exemplo anterior, *MSDM* é o único nome símbolo. Se $MSDM^{I_1} \preceq MSDM^{I_2}$, tem-se que $I_1 \preceq I_2$.

Um ponto fixo I de T_J é o menor ponto fixo (do inglês, *least fixpoint*, lfp) se $I \preceq I'$ para todos os outros pontos fixos I' . É chamado de *menor ponto fixo modelo* de T se I é o menor ponto fixo de T_J para alguma interpretação base J . Sob a semântica de lfp, somente admite-se menores pontos fixos modelos de T como interpretação a ser trabalhada. Analogamente são definidos maior ponto fixo (do inglês, *greatest fixpoint*, gfp), modelos de gfp e a semântica de maior ponto fixo. No exemplo *MSDM*, I_1 é o lfp e I_2 o gfp de T_J .

Semântica de gfp para $\mathcal{FL}_0$

Em Baader (1990) é possível encontrar o tratamento de ciclos para $\mathcal{FL}_0$. O efeito das três semânticas mencionadas anteriormente é capturado e descrito completamente com o conceito de autômatos finitos e linguagens regulares. Para uma introdução mais completa ao assunto, referenciar (SIPSER, 1997). Como consequência, subsunção pode ser reduzida a um problema de decisão em autômatos, à inclusão de linguagens regulares. Para a $\mathcal{FL}_0$ especificamente, a semântica de gfp cai melhor, envolve somente as linguagens regulares sobre o alfabeto de nomes de papéis (BAADER, 1990).

É válido lembrar que as definições de conceito com ciclos e semântica de gfp de $\mathcal{FL}_0$ podem ser obtidas proibindo ciclos simplesmente, permitindo, no lugar, definições de papéis envolvendo união, composição e fecho transitivo de papéis. Como será mostrado adiante, é possível obter as características da semântica de gfp para $\mathcal{FL}_0$ numa linguagem mais expressiva, neste caso $\mathcal{ALC}_{\text{reg}}$ derradeiramente.

No Exemplo 32, C classifica A com respeito a gfp por $L(C, P) = S^*$ ser um subconjunto de $L(A, P) = R^*S^*$ e $L(A, Q) = L(C, Q)$. O teorema 4 pode ser entendido da seguinte forma: a

linguagem $L(A, P)$ significa a (possivelmente) infinita quantidade de restrições da forma $\forall W : P$ que a terminologia impõe a A . Quanto mais restrições impostas, menor é o conceito.

Teorema 4. *Seja T uma terminologia de $\mathcal{FL}_0$ e A_T o autômato correspondente. Sejam também I um modelo gfp de T e A, B nomes de conceito ocorrendo em T :*

1. Para todos os conceitos primitivos P , todas as palavras $W \in L(A, P)$ e todos os indivíduos $y \in \text{dom}(I)$, $(x, y) \in W^I$ implica em $y \in P^I$ sse para todo $x \in \text{dom}(I)$, $x \in A^I$.
2. Subsunção em T pode ser reduzida à inclusão de linguagens regulares definidas por A_T . Mais precisamente, $A \sqsubseteq_{\text{gfp}, T} B$ sse $L(B, P) \subseteq L(A, P)$ para todos os conceitos primitivos P .

A parte (1) do teorema motiva a definição de restrição regular de valor.

Definição 43. (Uma “extensão regular” de $\mathcal{FL}_0$):

1. Seja L uma linguagem regular sobre um dado conjunto finito de nomes de papéis e C um termo conceito já definido: $\forall L : C$ é uma *restrição regular de valor*. Sua semântica é definida portanto como $(\forall L : C)^I := x \in \text{dom}(I); \forall W \in L \forall y \in \text{dom}(I), (x, y) \in W^I$ implica em $y \in C^I$.
2. Na extensão FL_{reg} é permitido o uso de restrições regulares de valor e conjunções de conceitos como operadores de formação de conceitos.

Com respeito à semântica gfp, a parte (1) do Teorema 4 implica que terminologias cíclicas de $\mathcal{FL}_0$ podem ser expressas por terminologias desdobradas de FL_{reg} . Assim, a Tbox cíclica do Exemplo 32 corresponde à Tbox desdobrada de FL_{reg} como segue:

$$\begin{aligned} A &= \forall R^* S^* : P \\ B &= \forall R S S^* : P \sqcap \forall S : Q \\ C &= \forall S^* : P \end{aligned}$$

Como naturalmente deve ser feito, os leitores devem esperar que uma terminologia desdobrada de FL_{reg} possa ser expressa por uma (possivelmente) cíclica de $\mathcal{FL}_0$. É pertinente demonstrar esta característica através de um exemplo.

Exemplo 34. Considere a seguinte Tbox desdobrada de FL_{reg} :

$$A = \forall RR^* : \forall S^* : P \sqcap \forall S^* : (P \sqcap Q)$$

No primeiro passo, usa-se o fato de que são equivalentes os conceitos:

$$\forall \varepsilon : B \text{ e } B;$$

$$\forall L : (B \sqcap C) \text{ e } \forall L : B \sqcap \forall L : C;$$

$$\forall K \forall L : B \text{ e } \forall K L : B;$$

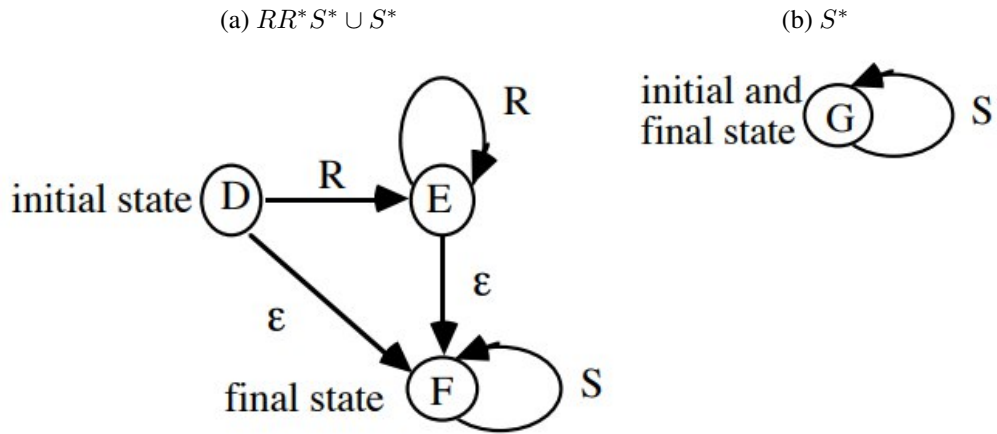
$$(\forall K : B) \sqcap (\forall L : B) \text{ e } \forall (K \cup L) : B.$$

Assim, é possível reorganizar o lado direito da igualdade para que somente ocorram conjunções finitas de termos na forma $\forall L : P$, tal que o conceito primitivo P ocorra somente uma vez na conjunção. A partir daí, obtém-se:

$$A = \forall (RR^* S^* \cup S^*) : P \sqcap \forall S^* : Q$$

Desta forma, o autômato finito equivalente é trivialmente concebido:

Figura 29 – Representação dos autômatos-resultado do Exemplo 34



Fonte: (BAADER, 1991)

E, munido disto, a definição de A envolvendo restrição regular de valores pode ser transformada nas seguintes definições de $\mathcal{FL}_0$:

$$\begin{aligned} A &= D \sqcap G \\ D &= \forall R : E \sqcap F \\ E &= \forall R : E \sqcap F \\ F &= \forall S : F \sqcap P \\ G &= \forall S : G \sqcap Q \end{aligned}$$

A correspondência 1–1 entre terminologias possivelmente cíclicas de $\mathcal{FL}_0$ e terminologia desdobrada de $\mathcal{FL}_{\text{reg}}$ foi demonstrada.

Enriquecimento da linguagem até $\mathcal{ALC}_{\text{reg}}$

Em sequência, segue a definição da extensão transitiva de $\mathcal{FL}$ conforme feito anteriormente:

Definição 44. (Uma "extensão transitiva" de $\mathcal{FL}$): Seja R um conjunto finito de nomes de papéis. O conjunto de papéis termos é definido indutivamente como a seguir: qualquer nome de papel é um termo (papel atômico), bem como o símbolo $\emptyset$ para papéis vazios. Para quaisquer dois papéis já definidos r e s , a união $r \sqcup s$, composição $r \circ s$ e o fecho transitivo de papéis $\text{trans}(R)$ são termos.

Em conclusão, para obter uma correspondência direta entre linguagens regulares em restrições de valor de FL_{reg} e papéis termos em restrições de valores de FL_{trans} é conveniente restringir as restrições regulares de valor de FL_{reg} para linguagens regulares não contendo a palavra vazia ε , ou, alternativamente, usar o fecho reflexivo-transitivo de papéis no lugar do fecho somente transitivo.

Agora apresenta-se a extensão $\mathcal{ALC}_{\text{reg}}$.

Definição 45. (Extensões transitiva e reflexiva de $\mathcal{ALC}$): Conforme as definições anteriores para $\mathcal{FL}_0$, assim é feito agora com $\mathcal{ALC}$:

1. Na extensão transitiva, é permitido o uso de papéis termos, como definido anteriormente em 10. A semântica segue da Definição 44 mais a extensão da função de interpretação I para papéis termos arbitrários:

$$\text{a) } \emptyset^I := \emptyset;$$

- b) $(r \sqcup s)^I := r^I \cup s^I$;
- c) $(r \circ s)^I := r^I \circ s^I$;
- d) $(trans(r))^I := \bigcup_{n \geq 1} (r^I)^n$;

2. Na extensão regular, é permitido o uso de restrição regular de valor e restrição regular existencial no lugar das restrições usuais. A semântica segue a Definição 43 mais a interpretação existencial:

- a) Para que a definição existencial siga a dualidade $\neg(\exists L.C) \equiv \forall r.(\neg C)$, $(\exists L.C)^I$ precisa ser definido como: $\{x \in dom(I); \exists W \in L \text{ e } \exists y \in dom(I), (x, y) \in W^I \text{ e } y \in CI\}$.

Proposição B.0.2. (*Poder de Expressividade*): Terminologias acíclicas de $\mathcal{ALC}_{trans}$ e $\mathcal{ALC}_{reg}$ ambas têm a mesma expressividade. Em particular, subsunção em $\mathcal{ALC}_{trans}$ pode ser reduzido em tempo linear a subsunção em $\mathcal{ALC}_{reg}$ e vice-versa, conforme (BAADER, 1991).

Expressividades em DL

Expressividade, ou poder expressivo, como escreve Divroodi (2014), é estudado nos campos de linguagens formais, banco de dados e lógica. A partir da hierarquia de Chomsky, obteve-se resultados fundamentais em expressividade de linguagens formais. Em Baader (1996), foi proposta uma definição formal das expressividades em DL. Tal definição permite comparar lógicas que possuem vocabulários diferentes. Em Borgida (1996), é mostrado a correspondência em poder expressivo entre certas DLs e fragmentos da LPO.

A metodologia usada para analisar a expressividade é chamada bissimulação. Ela o faz minimizando interpretações e aprendizagem de conceito nas DLs (DIVROODI, 2014). Os detalhes da técnica podem ser consultados em Divroodi (2014).

É válido citar as noções de invariância:

Definição 46. (Invariância na bissimulação): Um conceito C numa linguagem L em DL é invariante numa L -bissimulação, para quaisquer interpretações I e I' e qualquer bissimulação Z entre I e I' , se $Z(x, x')$ se verifica, então $x \in C^I$ sse $x' \in C^{I'}$. Uma Tbox T em L é invariante para L -bissimulação se, para quaisquer interpretações I e I' , se há pelo menos uma L -bissimulação entre as interpretações, então I é modelo de T sse I' é modelo de T . Similarmente, define-se uma Abox invariante.

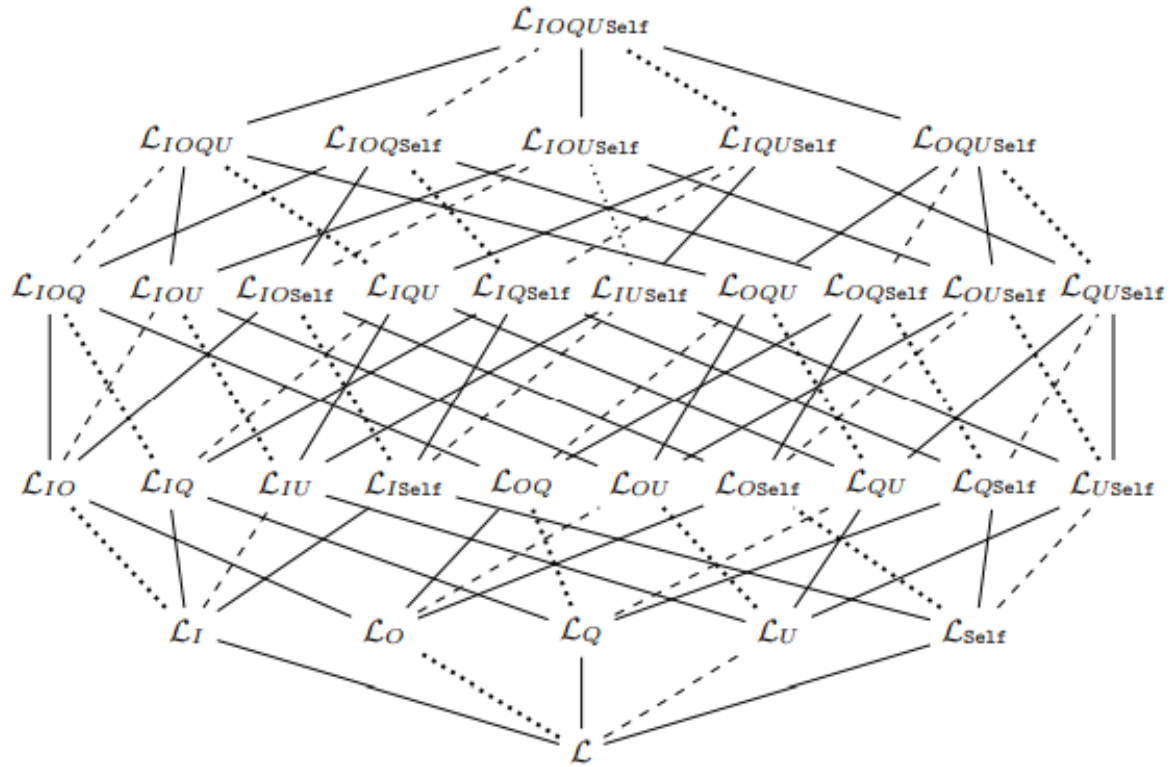
Ao comparar as expressividades, as noções de equivalência e linguagem mais ou menos expressiva segue como esperado:

Definição 47. (Equivalência entre linguagens): Estendendo a Definição 46, duas Tbox T_1 e T_2 são equivalentes se, para qualquer interpretação I , I é modelo de T_1 sse I é modelo de T_2 . Analogamente define-se Abox. Uma lógica L_1 é no máximo tão expressiva quanto uma lógica L_2 , denotado por $L_1 \leq L_2$, se todo conceito (respectivamente conceitos, Tbox e Abox) em L_1 tem um conceito equivalente em L_2 .

Definição 48. (Linguagem mais expressiva): Uma lógica L_2 é mais expressiva que uma lógica L_1 , denotado por $L_1 < L_2$, se $L_1 \leq L_2$ e $L_2 \not\leq L_1$.

Com base nestas definições, $\mathcal{ALC}_{\text{reg}}$ é mais expressiva que puramente $\mathcal{ALC}$. Existe uma família de linguagens entre $\mathcal{ALC}$ e $\mathcal{ALC}_{\text{reg}}$, como pode ser conferido no trabalho de Divroodi (2014, p. 11).

Figura 30 – Uma breve árvore genealógica DL: comparando a expressividade da Lógica de Descrições, onde $\mathcal{ALC} \leq L \leq \mathcal{ALC}_{reg}$. Se houver um caminho de uma lógica L_2 até uma lógica L_1 que contenha ou uma aresta normal ou pelo menos duas arestas, então L_2 é mais expressiva que L_1 com respeito a conceitos positivos, Tboxes e Aboxes. Se o caminho for uma aresta pontilhada então L_2 é mais expressiva que L_1 com respeito a conceitos, conceitos positivos e Tboxes. Se o caminho for uma aresta tracejada, então L_2 é mais expressiva que L_1 com respeito a conceitos, conceitos positivos e Aboxes.



Fonte: (DIVROODI, 2014)

PDL

Nesta seção, são comentadas algumas extensões da PDL e a possível equivalência com o μ -cálculo.

Extensões de PDL

A classe de programas *while* determinísticos (do inglês, *deterministic while programs*, DWP) é a classe dos programas em que os operadores $\cup$, $?$ e $*$ aparecem somente no contexto dos saltos, ou falhas e nos testes condicionais e laços, que são puramente proposicionais, sem os operadores modais. Os programas não determinísticos (do inglês, *while programs*, WP) são iguais, exceto que o operador $\cup$ é permitido ser não determinístico.

Restringindo a sintaxe ou semântica ou ambos, obtêm-se as lógicas:

- DPDL: PDL determinística sintaticamente idêntica a PDL;
- SPDL: PDL estrita, em que são permitidos os programas while;
- SDPDL: PDL estrita determinística, ambas as restrições anteriores.

Com a definição de um autômato, é possível obter APDL:

Definição 49. (Autômato para PDL): Seja um autômato $M = \{n, i, f, \delta\}$, em que n é o conjunto de estados, (i, f) são os estados destacados inicial e final respectivamente e δ a função de transição que atribui um subconjunto de $\pi_0 \cup \{\varphi? \mid \varphi \in \varphi\}$ para cada par de estados. A obtenção de APDL se dá por definir π como sendo: $\pi = \pi_0 \cup \{\varphi? \mid \varphi \in \varphi\} \cup F$ em que F é um conjunto dos estados de aceitação em M . Uma PDL que permite o fluxo ao contrário (do inglês, *converse*), é chamada CPDL. O operador *converse* é definido da seguinte forma:

Definição 50. (O operador *converse*): A semântica do operador *converse* é definida como:

$$m\mathfrak{K}(\alpha^-) \stackrel{def}{=} m\mathfrak{K}(\alpha)^- = (v, u) \mid (u, v) \in m\mathfrak{K}(\alpha).$$

Semanticamente, é como se a relação entrada/saída se transformasse em saída/entrada. Apesar de não ser possível sempre operar dessa forma na prática, em termos de expressividade, é um operador muito útil. Sendo o operador $^-$ simétrico, comutativo com os demais operadores, a prova do teorema 5 se torna trivial e será dispensada. O teorema que descreve a interação deste operador com os modais $\langle \rangle$ e $[]$ pode ser conferido em Harel, Kozen e Tiuryn (2000), cuja prova reside na página 179.

Teorema 5. (Operações com *converse*): Para quaisquer programas α e β ,

$$m\mathfrak{K}((\alpha \cup \beta)^-) = m\mathfrak{K}(\alpha^- \cup \beta^-)$$

$$m\mathfrak{K}((\alpha; \beta)^-) = m\mathfrak{K}(\beta^-; \alpha^-)$$

$$m\mathfrak{K}(\varphi?^-) = m\mathfrak{K}(\varphi?)$$

$$m\mathfrak{K}(\alpha^*-) = m\mathfrak{K}(\alpha)^-*$$

$$m\mathfrak{K}(\alpha^-) = m\mathfrak{K}(\alpha).$$

Ainda há as versões repeat e loop (RPDL e LPDL) que tentam tratar o conceito de bem-fundado, mas de maneira mais fraca, sendo mais expressivas que PDL, mas menos que o μ -cálculo estudado a seguir e, por isso, foram somente mencionadas. Elas foram trazidas por Streett e Sherman individualmente (STREETT, 1985).

μ -cálculo

Sempre que se fala em características infinitárias, é necessário para o raciocínio a concepção de modelos finitos das entidades com estas características. Por se tratar de um modelo, preserva as todas as propriedades do sistema original. Estes modelos têm a propriedade de possuir pontos fixos. Em PDL, o operador $*$ pode ser considerado um operador de ponto fixo inerentemente a sua definição. De fato, ele é o operador que representa trivialmente uma expressão no μ -cálculo proposicional, como estudado mais adiante. Antes disso, observe o exemplo:

Exemplo 35. Suponha infinito o tempo a partir do presente ao futuro e um indivíduo metódico que deseja compartimentalizar para melhor aproveitá-lo. Ele organiza tudo em conjuntos e define o menor deles como sendo $S = \{1s\}$. Apesar de somarem *ad infinitum*, como os naturais, é certo que, sendo uma organização monotônica, ele consiga estabelecer um conjunto que contenha S , mas que tenha um significado prático, uma interpretação no mundo dele. A conjuntos maiores foram dados nomes: $M = \{60S\}$, $H = \{60M\}$, $D = \{24H\}$, $Sem = \{7D\}$, $Mês = \{30D\}$, $Ano = \{365D\}$. No mundo em que ele vive, D parecia ser o menor conjunto que continha a base e representava significado e a partir de onde tudo voltaria a se repetir. Exceto que era também a partir de onde a ordenação dos conjuntos importava: Sem não poderia ser definido como a união de 7 D apenas, mas que cada D teria um nome, portanto definindo uma ordem. Assim, o menor conjunto que respeita a ordem e contém o conjunto atômico base S é Sem , o menor ponto fixo da organização: a partir dele, nada cresce, tudo se replica e se repete.

No μ -cálculo, o operador μ amarra variáveis relacionais. Se $\varphi(X)$ for uma expressão com uma variável livre X , então a expressão $\mu X.\varphi(X)$ representa o menor X tal que seja igual a expressão original, ou $\varphi(X) = X$.

Exemplo 36. O fecho reflexivo transitivo R^* de uma relação R binária é a menor relação que contém R e é fechada sob reflexividade e transitividade. No μ -cálculo, a forma de representação seria:

$$R_* = \mu X(x, y).(x = y \vee \exists z(R(x, z) \wedge X(z, y))).$$

E deveria ser lido como "a menor relação binária $X(x, y)$ tal que ou $x = y$, ou é relacionada por R a algum z , tal que z e y já são relacionados por X ".

Como em Pratt (1981a), a noção de minimização é vista tanto na teoria de funções recursivas, quando usada para especificar a raiz da função, ou seja, o último argumento que faz a função parar, quanto na semântica de programas, quando especifica o menor ponto fixo de uma função definida num reticulado, formalizando a definição de função recursivamente definida, conforme o exemplo anterior. Esta preocupação provê a ambas uma alternativa algébrica atrativa para iteração e recursão.

D. Kozen (1982) descreve o μ -cálculo proposicional, ou $P\mu$, também chamado de modal μ -cálculo, ou μML , como sendo "uma classe de lógicas de programa que consistem em lógicas modais proposicionais com um operador de menor ponto fixo μ ". Originado em Scott e J. (1969), foi desenvolvido por outros autores em sequência (KOZEN, D., 1982) (HAREL; KOZEN; TIURYN, 2000). Na época em que o trabalho foi publicado, ele não sabia de $P\mu$ continha estritamente PDL e não havia sistema dedutivo. Já em Harel, Kozen e Tiuryn (2000, p. 278), é afirmado que $P\mu$ é estritamente mais expressivo que PDL pelo trabalho de Streett (1985).

No trabalho de Pratt (1981b), ele define uma semântica focada na minimização da última raiz, com intuito de capturar a noção de *converse* de PDL. Apesar de seguir a axiomatização de Segerberg (1982), esta versão de Pratt se abstém dos construtores usuais de programa (união, sequência, teste, estrela e *converse*), substituindo tudo por minimização apenas. Outros aspectos desta versão do $P\mu$ é que o teorema da filtragem se aplica e ainda pode simplificar ao ponto de eliminar o fecho de Fischer-Ladner.

A existência de um menor ponto fixo não é garantida exceto por sob certas restrições. Uma fórmula $\neg X$ não tem ponto fixo, portanto $\mu X. \neg X$ não existe (HAREL; KOZEN; TIURYN, 2000). O que é praticado consiste em restringir a aplicação do operador μ a fórmulas que são positivas ou sintaticamente monotônicas em X , ou seja, fórmulas em que todas as ocorrências livres de X estejam no escopo de um número equivalente de negações. Assim, é garantido que o operador da relação $X \rightarrow \varphi(X)$ seja semanticamente monotônico, que, pelo teorema de Knaster-Tarski, é garantia de existência de menor ponto fixo.

Sintaxe e Semântica

A sintaxe e semântica do $P\mu$ apresentado aqui seguem de Streett (1985) e de Harel, Kozen e Tiuryn (2000).

Definição 51. (Sintaxe de $P\mu$): A sintaxe segue idêntica à de PDL com a adição do item

1. Se $\varphi(X)$ é uma fórmula sintaticamente monotônica na proposição X , $\mu X.\varphi(X)$ é uma fórmula.

Sendo assim, a dualidade modal de preserva:

$$\langle \alpha \rangle \varphi \equiv \neg[\alpha]\neg\varphi$$

$$\nu X.\varphi(X) \equiv \neg\mu X.\neg\varphi(\neg X)$$

A letra ν (ni) é usada para representar o maior ponto fixo e, portanto, a fórmula $\nu X.\varphi(X)$ denota o maior ponto fixo de φ .

Exemplo 37. A fórmula $\nu X.\langle \alpha \rangle(X)$ é verdade num estado x se houver uma cadeia infinita de transições α a partir de x .

Definição 52. (Semântica de $P\mu$): As fórmulas do $P\mu$ são interpretadas em modelos de Kripke, em que fórmulas são interpretadas como conjuntos de estados e programas como relações binárias entre os estados. Assim, x é satisfatível quando:

$$x \models \mu X.\varphi(X) \text{ sse } x \in \cap \{S \subseteq \Phi \mid S = \{y \mid y \models \varphi(X) \text{ com } X \text{ interpretado como } S\}\}$$

Exemplo 38. As fórmulas a seguir são equivalentes:

$$\mu X.\varphi \vee \alpha(X) \equiv \langle \alpha^* \rangle \varphi \equiv \exists r * \alpha.\Phi.$$

Interpretando, estas fórmulas são verdade em x se existe uma cadeia, ou sequência (possivelmente vazia) de α transições partindo de um estado x até um estado que satisfaça φ . A primeira notação diz respeito ao μ -cálculo, a segunda, à PDL, e a terceira, ao $\mathcal{ALC}_{\text{reg}}$.

A fórmula $\mu X.\langle \alpha \rangle[\alpha]X$ expressa num jogo de 2 jogadores a existência de uma vitória forçada para o primeiro jogador.

Definição 53. (Axiomatização de $P\mu$): A axiomatização consiste nos axiomas e regras da PDL mais o axioma

$$\varphi[X/\mu X.\varphi] \rightarrow \mu X.\varphi$$

e a regra para μ

$$\varphi[X/\psi] \rightarrow \psi$$

$$\mu X.\varphi \rightarrow \psi$$

Como escreve Baader et al. (2007), as Lógicas de Descrições e Modais têm um relacionamento muito próximo, primeiro descrito em Schild (1991b) apud (BAADER et al., 2007, p. 121). A princípio, Schild aponta que $\mathcal{ALC}$ pode ser enxergada como uma variante notacional da lógica multimodal $\mathbf{K}_m$. Posteriormente, foi observado em Schild (1991b) e De Giacomo e Lenzerini apud (BAADER et al., 2007, p. 160) um relacionamento similar em vertentes ainda mais expressivas, as abordadas neste trabalho. A extensão $\mathcal{ALC}_{\text{trans}}$ seria correspondente à PDL, e a $\mathcal{ALC}_{\text{reg}}$ seria correspondente ao μ -cálculo. Ainda mais, a extensão de $\mathcal{ALC}_{\text{trans}}$ pelo construtor de inverso, $\mathcal{ALC}\mathcal{I}_{\text{trans}}$, corresponderia inteiramente à PDL com *converse*, como escreve Donini em Baader et al. (2007, p. 122).

Embora, no mesmo texto, Salter *et al.* afirmem que " $\mathcal{ALC}_{\text{reg}}$ pode ser vista como variante notacional de PDL" (pág 160), $\mathcal{ALC}_{\text{reg}}$ trata de pontos fixos, tanto quanto o μ -cálculo, enquanto que PDL não expressamente, a despeito do axioma 3 da Definição 37. Em sequência, ele cita que o μ -cálculo é estritamente mais expressivo que todas estas linguagens, quando "o μ -cálculo pode expressar um programa bem fundado, [...] uma relação sem cadeias infinitas", como de fato já mencionado. Mais adiante será demonstrada uma correspondência, oriunda deste texto em referência, entre $\mathcal{ALC}\mathcal{I}_{\text{reg}}$ ($\mathcal{ALC}_{\text{reg}}$ com inverso) e *converse*-PDL.

Alguns dos algoritmos usados em PDL para decisão de satisfatibilidade são muito semelhantes aos algoritmos baseados em *tableau* desenvolvidos para DL (BAADER et al., 2007). Por isso, para decidir nessas lógicas é possível, no lugar de usar algoritmos baseados em *tableau*, usar o método de autômato em árvores (VARDI; WOLPER, 1986). Um catálogo de complexidades pode ser conferido em Baader et al. (2007, p. 139).

MAPEAMENTO ENTRE AS EXTENSÕES

A seguir, é mostrada a correspondência entre $\mathcal{ALCI}_{\text{reg}}$ e *converse*-PDL. Como escrevem Calvanese e De Giacomo (BAADER et al., 2007, p. 191), num nível extensional, indivíduos membros de δI na Lógica de Descrições correspondem a estados em PDL, enquanto as transições entre estes estados são ligações entre dois indivíduos em DL. No nível intensional, os conceitos correspondem às proposições e os papéis correspondem aos programas. Formalmente, é feito um mapeamento π de conceitos $\mathcal{ALCI}_{\text{reg}}$ a fórmulas *converse*-PDL e de papéis $\mathcal{ALCI}_{\text{reg}}$ a programas *converse*-PDL, definido indutivamente como:

$$\begin{array}{ll}
 \pi(A) = \varphi & \pi(P) = \alpha \\
 \pi(\neg(C)) = \neg\pi(C) & \pi(R^-) = \pi(R)^- \\
 \pi(C \sqcap D) = \pi(C) \wedge \pi(D) & \pi(R \sqcup S) = \pi(R) \cup \pi(S) \\
 \pi(C \sqcup D) = \pi(C) \vee \pi(D) & \pi(R \circ S) = \pi(R); \pi(S) \\
 \pi(\forall r.C) = [\pi(r)]\pi(C) & \pi(R^*) = \pi(R)^* \\
 \pi(\exists r.C) = \langle \pi(r) \rangle \pi(C) & \pi(id(C)) = \pi(C)?
 \end{array}$$

No trabalho de Schild (1991a), ele demonstra a correspondência de *converse*-PDL para um enriquecimento de uma linguagem terminológica (a ACC) chamada TSL, que adiciona identidade e composição, união, fecho transitivo-reflexivo, restrição de imagem e o inverso de papéis.

APÊNDICE C – TEOREMAS E PROVAS

FILTRAGEM

O método de filtragem exerce um papel importante no contexto da lógica modal: é usado para provar decidibilidade e na obtenção de teoremas de completude (GABBAY, 1972). Com um conjunto infinito de variáveis proposicionais, existem também conjuntos maximalmente consistentes de fórmulas do mesmo tamanho.

No texto de Marcus (1981, p. 280), ele menciona os sistemas modais normais **S4** e **S5**. Veja o Exemplo 39:

Exemplo 39. Para cada um dos sistemas abaixo, o aspecto infinitário das sentenças pode ser tratado da seguinte forma:

Observe que, no sistema **S4**, há um número arbitrariamente grande de operadores modais em sua estrutura com preordem¹³:

(**S4**): $\Box_1 \Box_2 \dots \Box_n p \rightarrow \Box p$ e $\Diamond_1 \Diamond_2 \dots \Diamond_n p \rightarrow \Diamond p$ (Estruturas com preordem);

Observe agora que, mesmo com um número infinito de fórmulas antes do operador, no sistema **S5** (com relações de equivalência), a relação é soberana e prevalece somente um operador, que se mantém no final:

(**S5**): $\phi_1 \phi_2 \dots \Box p \rightarrow \Box p$ e $\phi_1 \phi_2 \dots \Diamond p \rightarrow \Diamond p$ (Estruturas com relação de equivalência);

Gabbay afirma que o método foi introduzido por Lemmon et al. (1966 apud GABBAY, 1972, p. 1), mas modificado por outro autor e ele próprio posteriormente. Ele consiste em conceber a partir de um dado modelo de Kripke $\langle \mathcal{W}, \mathfrak{R}, 0 \rangle$, que é modelo para uma sentença ψ , um modelo $\mathfrak{M} = \langle \mathcal{W}^*, \mathfrak{R}^*, 0 \rangle$ tal que respeite a condição de ser modelo para ψ também. O último termo da tripla representa o mundo atual e não a constante **0**, como o item 8 da Definição 20. A sentença ψ está associada a um conjunto finito de sentenças (Θ). O segundo modelo pode ser gerado assumindo Θ como sendo o conjunto de subfórmulas de ψ , identificando 2 mundos possíveis que deem os mesmos valores verdade a todos os membros de Θ e definir xR^*y como $\forall \alpha \in \Theta$ (se $\Box \alpha$ se verifica em x , então também se verifica em y). Aqui, $[]$ é uma função binária que $\forall x \in \mathcal{W}$ e todas as variáveis proposicionais, $[y]_x$ é **0**, ou **1**. Portanto, $[]$ é considerada a função de avaliação que, para uma fórmula φ , seu valor verdade num mundo x é denotado por $[\varphi]_x$.

¹³ Mais sobre este conceito, consultar (HRBACEK; JECH, 1999)

Quando é o caso de provar decidibilidade, é preciso fazer com que $\mathcal{W}^*$ seja finito e modelo do sistema a ser considerado. Pode haver a necessidade de provar a completude de um conjunto de mundos possíveis, por exemplo o conjunto dos inteiros, fazendo-se $\mathcal{W}^*$ tal que seja os inteiros. Portanto, seja X uma modalidade, X admite filtragem sse para qualquer modelo $\langle \mathcal{W}, \mathfrak{R}, 0 \rangle$ de ψ que é modelo para X , um modelo finito $\langle \mathcal{W}^*, \mathfrak{R}^*, 0 \rangle$ para ψ pode ser derivado. Assim, é necessário sempre que sejam gerados $\langle \mathcal{W}^*, \mathfrak{R}^* \rangle$ tais que sejam finitos, ou admitam a propriedade do modelo pequeno, como em Harel, Kozen e Tiuryn (2000, p. 191).

Para tanto, sendo $\mathcal{W}$ o conjunto de mundos possíveis ou estados, $\mathfrak{R}$ a relação de acessibilidade entre os mundos e 0 o estado atual, é suficiente que $\mathcal{W}$ seja um conjunto binário tal que preserve todos os valores verdade para um conjunto Σ sendo ele um conjunto finito de sentenças que contém ψ e é fechado sob negação, conjunção e subfórmulas.

Teorema 6. *Seja K o sistema modal com o axioma $\Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q)$ e a regra da necessitação $\models p \rightarrow \models \Box p$, então os sistemas modais normais admitem filtragem.*¹⁴

Prova: Defina:

$\mathfrak{w} \equiv v$ sse $\forall \varphi \in \Sigma ([\varphi]\mathfrak{w} = [\varphi]v)$.

Sejam:

$[\mathfrak{w}]_\Sigma$ a classe de equivalência sobre $\equiv$ de $\mathfrak{w}$ (ou $\mathfrak{w}^*$, por alívio notacional) e

$L(y^*)$ a fórmula $(\exists y \in y^*) \forall z (yRz \rightarrow y = z)$,

Defina:

$$\mathfrak{w}^* R^+ v^* \text{ sse } (\exists \mathfrak{w}_1 \in \mathfrak{w}^*) (\exists v_1 \in v^*) (\mathfrak{w}_1 R^+ v_1) \quad (\text{C.1})$$

e seja $\mathfrak{w}^* R_1^* v^*$ definido por $\mathfrak{w}^* R^+ v^* \wedge \neg L(x^*)$. A variável proposicional γ é portanto definida:

$$[\gamma]_{x^*}^+ = [\gamma]_x \quad (\text{C.2})$$

□

Lema 5. *Para qualquer subfórmula φ de ψ tem-se:*

$$[\varphi]_{\mathfrak{w}} = [\varphi]_{\mathfrak{w}^*}^+$$

Prova: caso atômico: trivial.

¹⁴ Atentar para os sistemas *quasi* e *multi* de Bull e Segerberg (2001)[p. 16]

caso $\neg$: se existe um modelo, então $[\neg\varphi]_{\mathfrak{w}} = 0$ e $[\varphi]_{\mathfrak{w}} = 1$.

caso $\wedge$: indução simples dos casos anteriores.

caso $\Box$: seja $[\Box\varphi]_{\mathfrak{w}} = 1$;

se $\Box\varphi \in \Sigma$, então $[\Box\varphi]_{\mathfrak{w}}^1 = 1$ para todo $\mathfrak{w}^1 \in \mathfrak{w}^*$;

se C.1, então $[\Box\varphi]_{v^1} = 1$ e $[\Box\varphi]_{v^*} = 1$.

Seja $[\Box\varphi]_{\mathfrak{w}} = 0$ e $\neg L(\mathfrak{w}^*)$ para algum v tal que $\mathfrak{w}Rv$, $[\Box\varphi]_v = 0$ e

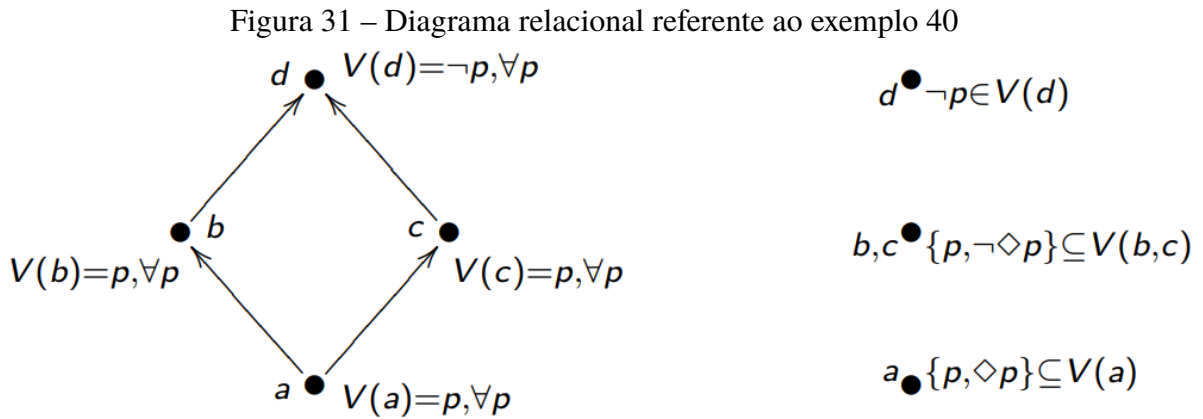
por C.1: $\mathfrak{w}^*R^+v^*$ e $[\Box\varphi]_{v^*} = 0$. Se $L(x^*)$ se verifica, e $[\Box\varphi]_{\mathfrak{w}} = 0$ $v = \mathfrak{w}$.

□

Interpretando o teorema:

Tendo entendido que $\mathfrak{w} \equiv v$ é uma relação de equivalência e $[\mathfrak{w}]\Sigma$ como a classe de equivalência, de maneira geral, o conjunto quociente de todas estas classes de equivalência é denotado por W_Σ .

Exemplo 40. os conjuntos $\{a\}$, $\{b, c\}$ e $\{d\}$ são as classes de equivalência dos modelos abaixo pelo conjunto de fórmulas $\{p, \Diamond p\}$:



Fonte: Elaborada pelo autor (2022)

Agora, sobre o $\mathfrak{R}^*$, em sendo uma relação binária em W_Σ , ela é chamada Σ -apropriada quando para toda dupla (w, v) pertencente a W :

- Se $(w, v) \in \mathfrak{R}$, então $([w]\Sigma, [v]\Sigma) \in \mathfrak{R}^*$.

- Se:

- $\Diamond\varphi \in \Sigma$,
- $([w]\Sigma, [v]\Sigma) \in \mathfrak{R}^*$,
- $\mathfrak{M}, v \models \varphi$,

Então $\mathfrak{M}, w \models \Diamond \varphi$.

O entendimento da característica Σ -apropriada é direto a partir do entendimento dos operadores modais canônicos.

Assim, de uma maneira geral, a filtragem de um modelo $\mathfrak{M} = \langle \mathcal{W}, \mathfrak{R}, \nu \rangle$ através de Σ , o conjunto de fórmulas fechado sob subfórmulas, conjunção e negação, é o modelo

$$\mathfrak{M}_\Sigma = \langle \mathcal{W}_\Sigma, \mathfrak{R}_\Sigma, \nu_\Sigma \rangle,$$

tal que:

- $\mathcal{W}_\Sigma$ é o conjunto quociente,
- $\mathfrak{R}_\Sigma$ é a relação binária Σ -apropriada e
- ν_Σ é a função de valoração definida por: $\nu_\Sigma(p, [w]_\Sigma) = 1$ sse $\nu(p, w) = 1$.

DEMAIS PROVAS

Prova: Eis a prova do item i) do teorema 1, pois tendo em vista que a prova desses teoremas se trata de álgebra básica relacional, por motivo de sucintez e celeridade, é provado explicitamente o i). O restante segue a partir dele:

Para qualquer modelo de Kripke $\mathfrak{M} = \langle \mathcal{W}, \mathfrak{R}, \nu \rangle$,

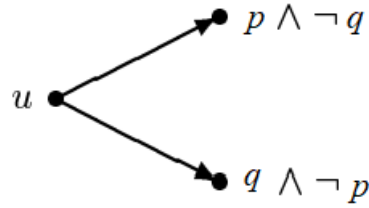
$$\begin{aligned}
 \nu_{\mathfrak{M}}(\Diamond(p \vee q)) &= \nu_{\mathfrak{M}}(\Diamond p \vee \Diamond q): \\
 \nu_{\mathfrak{M}}(\Diamond(p \vee q)) &= \mathbf{R} \circ \nu_{\mathfrak{M}}(p \vee q) && \text{semântica de } \Diamond \\
 &= \mathbf{R} \circ (\nu_{\mathfrak{M}}(p) \cup \nu_{\mathfrak{M}}(q)) && \text{semântica da lógica proposicional} \\
 &= (\mathbf{R} \circ \nu_{\mathfrak{M}}(p)) \cup (\mathbf{R} \circ \nu_{\mathfrak{M}}(q)) && \text{álgebra relacional} \\
 &= \nu_{\mathfrak{M}}(\Diamond p) \cup \nu_{\mathfrak{M}}(\Diamond q) && \text{semântica de } \Diamond \\
 &= \nu_{\mathfrak{M}}(\Diamond p \vee \Diamond q) && \text{semântica da lógica proposicional}
 \end{aligned}$$

□

Para o iii), a equivalência não se verifica, quer dizer, somente é válida a ida, a implicação, não a volta. O contraexemplo seria como mostra o grafo:

O seguinte teorema está presente em Harel, Kozen e Tiuryn (2000). Logo após ele é definido o enriquecimento da lógica clássica para uma multimodal, introduzindo os operadores como na PDL.

Figura 32 – Grafo que exibe o ponto de insatisfatibilidade do modelo.



Fonte: Elaborada pelo autor (2022)

Teorema 7. *São regras corretas na PML:*

1. *Generalização Modal:*

$$\frac{p}{\Box p}$$

2. *Monotonicidade do $\Box$:*

$$\frac{p \rightarrow q}{\Box p \rightarrow \Box q}$$

3. *Monotonicidade do $\Diamond$:*

$$\frac{p \rightarrow q}{\Diamond p \rightarrow \Diamond q}$$

Prova: Seja $\mathfrak{M} = \langle \mathcal{W}, \mathfrak{R}, v \rangle$ um modelo de Kripke.

i) Se

$v_{\mathfrak{M}}(p) = \mathcal{W}$, então

$\mathfrak{R}_{\mathfrak{M}} \circ (\mathcal{W} \setminus v_{\mathfrak{M}}(p)) = \emptyset$ e assim

$\mathcal{W} - (\mathfrak{R}_{\mathfrak{M}} \circ (\mathcal{W} - v_{\mathfrak{M}}(p))) = \mathcal{W}$.

ii) Pela propriedade monotônica de $\circ$, se

$v_{\mathfrak{M}}(p) \subseteq v_{\mathfrak{M}}(q)$, então

$\mathfrak{R}_{\mathfrak{M}} \circ v_{\mathfrak{M}}(p) \subseteq \mathfrak{R}_{\mathfrak{M}} \circ v_{\mathfrak{M}}(q)$.

iii) Se

$$\begin{aligned} \nu_{\mathfrak{M}}(p) &\subseteq \nu_{\mathfrak{M}}(q), \text{ então} \\ (\mathcal{W} - \nu_{\mathfrak{M}}(p)) &\subseteq (\mathcal{W} - \nu_{\mathfrak{M}}(q)). \end{aligned}$$

Por ii), tem-se:

$$\begin{aligned} \mathfrak{R}_{\mathfrak{M}} \circ (\mathcal{W} - \nu_{\mathfrak{M}}(p)) &\subseteq \mathfrak{R}_{\mathfrak{M}} \circ (\mathcal{W} - \nu_{\mathfrak{M}}(q)) \text{ e assim} \\ \mathcal{W} - (\mathfrak{R}_{\mathfrak{M}} \circ (\mathcal{W} - \nu_{\mathfrak{M}}(p))) &\subseteq \mathcal{W} - (\mathfrak{R}_{\mathfrak{M}} \circ (\mathcal{W} - \nu_{\mathfrak{M}}(q))). \end{aligned}$$

□

REGRAS DE ANÁLISE PARA ALGUMAS LÓGICAS MODAIS NORMAIS

Eis aqui as regras de análise para algumas Lógicas Modais Normais:

Em BALBIANI e BOUDOU (2015), a regra do $\langle \rangle$ leva em consideração o tamanho de α , a quantidade de execuções do programa, em termos da PDL. Tal detalhamento será omitido com o propósito de sucintez e celeridade. As regras para o operador de execução sequencial têm preocupação semelhante. Aqui será apresentada somente a regra do operador com o $[]$ da necessidade. Da mesma forma, será apresentada somente a regra com o operador $[]$ para a iteração.

Regra do $?$: Dependendo do operador modal, a análise para o operador de teste mantém o *tableau* sequencial ou cria uma bifurcação.

$$\begin{aligned} ([?]): & \frac{[\varphi?] \psi}{\neg \varphi \mid \psi} \\ (\langle ? \rangle): & \frac{\varphi?(x, x)}{\varphi(x)} \end{aligned}$$

Regra do $;$: A ramificação continua sequencialmente.

$$([;]): \frac{[\alpha; \beta] \varphi}{[\alpha][\beta] \varphi}$$

Regra do $*$: A ramificação continua sequencialmente.

Figura 33 – Regras de análise para Lógicas Modais Normais

$$(K) \frac{\Box X; \neg \Box P}{X; \neg P} \quad (T) \frac{X; \Box P}{X; \Box P; P} \quad (D) \frac{X; \Box P}{X; \Box P; \neg \Box \neg P}$$

$$(KD) \frac{\Box X; \neg \Box P}{X; \neg P} \text{ where } \{\neg \Box P, \neg P\} \text{ may be empty}$$

$$(K4) \frac{\Box X; \neg \Box P}{X; \Box X; \neg P} \quad (S4) \frac{\Box X; \neg \Box P}{\Box X; \neg P}$$

$$(45) \frac{\Box X; \neg \Box Y; \neg \Box P}{X; \Box X; \neg \Box Y; \neg \Box P; \neg P}$$

$$(45D) \frac{\Box X; \neg \Box Y; \neg \Box P}{X; \Box X; \neg \Box Y; \neg \Box P; \neg P} \text{ where } Y \cup \{P\} \cup \{\neg P\} \text{ may be empty}$$

$$(B) \frac{X; \neg \Box P}{X; \neg \Box P; P \mid X; \neg \Box P; \neg P; \Box \neg \Box P} \quad (T_{\Diamond}) \frac{X; \Box P; \neg \Box Q}{X; \Box P; P; \neg \Box Q}$$

$$(5) \frac{X; \neg \Box P}{X; \neg \Box P; \Box \neg \Box P} \quad (S5) \frac{\Box X; \neg \Box Y; \neg \Box P}{\Box X; \neg \Box Y; \neg \Box P; \neg P}$$

Fonte: (GORÉ, 1999)

$$([*]): \frac{\frac{[\alpha*]\varphi}{\varphi}}{[\alpha][\alpha*]\varphi}$$

APÊNDICE D – APORTE

HIERARQUIA DE RAMIFICAÇÕES DA DL

Observe a tabela de hierarquia de linguagens, cujo modelo foi inspirado em Baader et al. (2007)

Quadro 2 – Hierarquia de linguagens

Linguagem	Nome do Construtor	Sintaxe abstrata	Sintaxe concreta	Semântica
Todas	Indivíduos	x	nome de indivíduo	x^I
	Papéis	r	letras minúsculas	r^I
$\mathcal{FL}_0$	Conceito	C	letras maiúsculas	C^I
	Topo	$\top$	TOP	Δ^I
	Fundo	$\perp$	BOTTOM	$\emptyset$
	Interseção	$C_1 \sqcap \dots \sqcap C_n$	(e $C_1 \dots C_n$)	$C^I \cap D^I$
	Restrição de valor	$\forall r.C$	(todo rC)	$\{a \in \Delta^I \mid \forall b. (a, b) \in r^I \rightarrow b \in C^I\}$
$\mathcal{EL} : \mathcal{FL}_0 + \exists - \forall$	Quantificação existencial	$\exists r.C$	(algum rC)	$\{a \in \Delta^I \mid \exists b. (a, b) \in r^I, b \in C^I\}$
$\mathcal{FL}^- : \mathcal{FL}_0 + \exists. \top - \exists$	Quantificação existencial limitada	$\exists r. \top$	(algum r)	$\{a \in \Delta^I \mid \forall b. (a, b) \in r^I, b \in C^I\}$
$\mathcal{AL} : \mathcal{FL}_0 + \mathcal{EL} + \neg$	Negação	$\neg C$	(não C)	$\Delta^I \setminus C^I$
$\mathcal{ALL} : \mathcal{AL} + \cup$	União	$C_1 \sqcup \dots \sqcup C_n$	(ou $C_1 \dots C_n$)	$C^I \cup D^I$

Fonte: Elaborada pelo autor (2022)

NOMINAIS EM DL

Em se tratando de DL, os nominais permitem o uso de instâncias de uma ABox para criar classes, que dizer, transformam instâncias em fórmulas próprias da TBox. Como são definidos como conjuntos unitários, os nominais se aplicam essencialmente a todas as DLs e têm

muita utilidade para propósitos teóricos, visto que raciocinar com eles é em geral muito mais complicado que com a ABox simplesmente.

Definição 54. (Nominal e codificação para TBox): Um nominal é definido como sendo um conceito que representa um conjunto unitário. Se n é uma instância, $\{n\}$ é um nominal. Assim, as formas de instanciação de conceitos e papéis devem ser escritas como fórmulas da TBox como:

- $C(a) \stackrel{def}{=} \{a\} \sqsubseteq C$
- $r(a, b) \stackrel{def}{=} \{a\} \sqsubseteq \exists r. \{b\}$

Exemplo 41. Repare nas sentenças a seguir reescritas em DL com a utilização de nominais:

"Existem exatamente 195 países"

1. País $\equiv \{\text{afeganistão}, \text{albânia}, \dots, \text{zimbábue}\}$
2. $\{\text{afeganistão}\} \sqsubseteq \neg\{\text{albânia}\} \sqcap \dots \sqcap \neg\{\text{zimbábue}\} \sqcap \{\text{afeganistão}\}$
196. $\{\text{zimbábue}\} \sqsubseteq \neg\{\text{afeganistão}\} \sqcap \neg\{\text{albânia}\} \sqcap \dots \sqcap \{\text{zimbábue}\}$

"Ana ama Bob ou Cal"

1. $\{\text{ana}\} \sqsubseteq \exists \text{ama}. \{\text{bob}, \text{cal}\}$

AUTÔMATOS E LINGUAGENS REGULARES

Todas as definições aqui apresentadas podem ser consultadas em Sipser (1997), quanto o teorema, cuja prova também pode, bem como um aprofundamento maior na teoria.

Definição 55. (Autômato finito determinístico (AFD)) Um autômato finito é uma 5-upla $(Q, \Sigma, \delta, q_0, F)$ tal que:

1. Q é um conjunto finito de estados;
2. σ é um conjunto finito chamado alfabeto;
3. $\delta : Q \times \Sigma \rightarrow Q$ é a função de transição;
4. q_0 é o estado inicial (de símbolo arbitrário);
5. $F \subseteq Q$ é o conjunto de estados de aceitação (comumente chamado de finais).

Definição 56. (Autômato finito não-determinístico (AFN)) Um autômato finito não-determinístico é uma 5-upla $(Q, \Sigma, \delta, q_0, F)$ tal que:

1. Q é um conjunto finito de estados;
2. σ é um conjunto finito chamado alfabeto;
3. $\delta : Q \times \Sigma_\epsilon \rightarrow Q$ é a função de transição (ϵ representa a palavra vazia);
4. q_0 é o estado inicial (de símbolo arbitrário);
5. $F \subseteq Q$ é o conjunto de estados de aceitação.

Teorema 8. *Todo autômato finito não-determinístico tem um autômato determinístico equivalente.*

Definição 57. (Computação) Seja $A = (Q, \Sigma, \delta, q_0, F)$ um autômato finito e $w = w_1w_2...w_n$ uma palavra membro do alfabeto Σ . Diz-se que A aceita w se uma sequência de estados $r_0 \rightarrow r_n$ em Q existe com as seguintes condições:

1. $r_0 = q_0$ (o autômato começa no estado inicial);
2. $\delta(r_i, w_i + 1) = r_{i+1}$, para $0 \leq i < n$ (o autômato segue de estado a estado pela função de transição);
3. $r_n \in F$ (se o autômato termina num estado final, ele aceita a cadeia de entrada).

É dito que A reconhece a linguagem L se $L = \{w | A \text{ aceita } w\}$

Definição 58. (Linguagem Regular): Uma linguagem é dita regular se algum autômato finito a reconhece.

RETICULADOS E PONTOS-FIXOS

Como um conhecimento prévio necessário, é válido mencionar como uma brevíssima introdução o conceito formal de ponto fixo. Muito da teoria foi obtido em DAVEY e PRIESTLEY (2002), onde está definido, por exemplo, como construir reticulados (do inglês, *lattice*) (pág. 11). Para uma introdução aos conceitos de teoria dos conjuntos, consultar (HRBACEK; JECH, 1999).

Definição 59. (Reticulado): Uma estrutura ou diagrama $L = (L, R)$ é um reticulado se L é parcialmente ordenado por R e, para cada par (a, b) de elementos em L , existe supremo e ínfimo de $\{a, b\}$.

Definição 60. (Mapeamentos entre conjuntos ordenados): Sejam P e Q conjuntos ordenados. Um mapeamento $\pi: P \rightarrow Q$ é dito ser:

1. Ordem-preservante (ou monotônico): se $x \leq y$ em $P \rightarrow \pi(x) \leq \pi(y)$ em Q .
2. Ordem-embarcante (escrito $\pi: P \hookrightarrow Q$): se $x \leq y$ em P sse $\pi(x) \leq \pi(y)$ em Q .
3. Ordem-isomorfismo: se é um ordem-embarcante e mapeia por sobrejeção P em Q .

Teorema 9. (Teorema de Knaster-Tarski): Seja L um reticulado completo e $F: L \rightarrow L$ um mapeamento ordem-preservante. Então:

$$\alpha := \text{Sup} \{x \in L \mid x \leq F(x)\}$$

é um ponto fixo de F . Tal definição diz respeito ao maior ponto fixo de F . Como definição dual,

$$\alpha := \text{Inf} \{x \in L \mid F(x) \leq x\}$$

diz respeito ao menor ponto fixo.

Definição 61. (Mapeamento parcial): Um mapeamento $\sigma: S \rightarrow Y$ de X para Y é dito parcial quando S é um subconjunto do domínio X .

Exemplo 42. A função fatorial.

Considere o conjunto $\mathbb{N}^0$. A função fatorial, **fat**, é o mapeamento em $\mathbb{N}^0$ dado por **fat**(k) = $k!$ que satisfaz

$$\mathbf{fat}(k) = \begin{cases} 1 & \text{se } k = 0, \\ k \mathbf{fat}(k-1) & \text{se } k \geq 1. \end{cases}$$

Para cada mapeamento $f: \mathbb{N}^0 \rightarrow \mathbb{N}^0$, pode-se associar um novo mapeamento $\bar{f}$ dado por

$$\bar{f}(k) = \begin{cases} 1 & \text{se } k = 0, \\ k \mathbf{fat}(k-1) & \text{se } k \geq 1. \end{cases}$$

Defina-se um mapeamento F em $(\mathbb{N}^0 \rightarrow \mathbb{N}^0)$ por $F(f) = \bar{f}$. Tem-se, portanto que $F(\mathbf{fat}) = \mathbf{fat}$.

Eis um exemplo de uma equação recursiva. A função fatorial inteira não pode ser destrinchada de sua especificação recursiva num número finito de passos. Entretanto, um mapeamento

parcial f_n pode ser definido como a restrição de **fat** a $\mathbb{N}^0$. Para acomodar aproximações a **fat**, é natural trabalhar com todos os mapeamentos parciais em $\mathbb{N}^0$. Neste caso, $\bar{f}$ tem como domínio $\{0\} \cup \{k \mid k-1 \in \text{dom } f\}$ para cada f que pertence ao mapeamento parcial. Em resumo, F pode ser estendido a um mapeamento parcial para ele mesmo e considerar a **fat** como a solução da equação recursiva $F(f) = f$.

De modo similar, podem-se considerar soluções de mapeamento parcial f em $\mathbb{Z}$ da forma $G(f) = f$ tal que

$$G(f)(0) = 1,$$

$$G(f)(k) = kf(k-1) \text{ se } k \geq 1 \text{ e } f(k-1) \neq \perp \text{ e}$$

$$G(f)(k) = f(k) \text{ se } k < 0.$$

Um ponto fixo f de G é determinado unicamente pelos inteiros não negativos. Contudo, a $f(-1)$ pode ser atribuído qualquer valor em $\mathbb{Z}$. Assim, G tem pontos fixos diferentes e demais! Não obstante, ele tem o menor ponto fixo: o mapa parcial que coincide com **fat** em $\mathbb{N}^0$ e indefinido caso contrário. As aplicações de menor ponto fixo têm geralmente maior relevância.

ÁLGEBRA DE KLEENE

É pertinente lembrar operações importantes com o operador estrela de Kleene (KOZEN, D. C., 1997, p. 55). A álgebra de Kleene é uma estrutura algébrica $(K, +, \cdot, *, 0, 1)$ que satisfaz os axiomas como em Harel, Kozen e Tiuryn (2000, p. 419).

- $0^* = 1$ e $1^* = 1$
- $a \leq b \rightarrow a^* \leq b^*$ (monotonicidade),
- $a_n \leq a^*$ para todo número natural n , tal que a_n é definido como a n -ésima multiplicação de a ,
- $(a^*)(a^*) = a^*$,
- $(a^*)^* = a^*$,
- $1 + a(a^*) = a^* = 1 + (a^*)a$,
- $ax = xb \rightarrow (a^*)x = x(b^*)$,

- $((ab)^*)a = a((ba)^*),$
- $(a+b)^* = a^*(b(a^*))^*,$
- $pq = 1 = qp \rightarrow q(a^*)p = (qap)^*.$