UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO
EM CIÊNCIA DA COMPUTAÇÃO

Aracele Garcia de Oliveira Fassbinder

Uma Apresentação dos Principais Sistemas Relacionados à Lógica Clássica

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação.

Prof. Dr. Arthur Ronald de Vallauris Buchsbaum
Orientador

Florianópolis, julho de 2010

Catalogação na fonte pela Biblioteca Universitária
da
Universidade Federal de Santa Catarina

F249a Fassbinder, Aracele Garcia de Oliveira
Uma Apresentação dos principais sistemas relacionados à lógica clássica [dissertação] / Aracele Garcia de Oliveira Fassbinder; orientador, Arthur Ronald de Vallauris Buchsbaum. - Florianópolis, SC, 2010.
307 p.: il., tabs.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Ciência da Computação.

Inclui referências

1. Informática. 2. Ciência da computação. 3. Lógica formal. 4. Representação do conhecimento. I. Buchsbaum, Arthur Ronald de Vallauris. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Ciência da Computação. III. Título.

CDU 681

Uma Apresentação dos Principais Sistemas Relacionados à Lógica Clássica

Aracele Garcia de Oliveira Fassbinder

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação, área de concentração Sistemas de Conhecimento e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Prof. Dr. Mário Antônio Ribeiro Dantas
Coordenador do Programa de Pós-Graduação em Ciência da Computação

Banca Examinadora

Prof. Dr. Arthur Ronald de Vallauris Buchsbaum
Orientador

Prof. Dr. Carlos Becker Westphall
INE/UFSC

Prof. Dr. João Bosco Mangueira Sobral
INE/UFSC

Profª. Dra. Ana Teresa de Castro Martins
UFC

"It is reasonable to hope that the relationship between computation and mathematical logic will be as fruitful in the next century as that between analysis and physics in the last."[a]

[a]McCarthy 1963, p. 69.

Aos amantes da computação e da lógica.

Agradecimentos

Aos meus ex-alunos do IFET de Formiga/MG e da UNIPAC de Campo Belo/MG, que indiretamente fortaleceram em mim o desejo de buscar a qualificação profissional através do mestrado, a fim de continuar progredindo na carreira como docente e pesquisadora.

Ao meu orientador, Arthur Buchsbaum, pela ajuda na estruturação geral deste trabalho, por ter me incentivado a estudar Lógica e pela disposição em me atender sempre que dele precisei.

Aos demais professores e colaboradores administrativos do Programa de Pós-Graduação em Ciência da Computação da UFSC.

Aos colegas de curso, especialmente Marília Guterres, Alex Magalhães, Álvaro Altair e Iuri Sônego, pela convivência e pelo companheirismo nas aulas de Lógica Formal.

À CAPES, pelo auxílio financeiro concedido no último ano deste curso.

Aos professores Cléuzio Fonseca Filho e Ana Teresa de Castro Martins, pelas sugestões que muito contribuíram para este trabalho.

Aos meus pais, Isabel e João Batista. O simples fato de existirem seria suficiente para expressar a minha enorme gratidão. Gratidão pela vida, pelo amor, pela base familiar que constituíram, por incentivarem os meus sonhos, pelo apoio moral e financeiro. Obrigada por tudo.

Ao meu esposo, Marcelo Fassbinder, pelo amor e pelo companheirismo.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Símbolos

$\rightleftharpoons$	Sinal para abreviaturas
$=$	Igualdade
$\neq$	Diferença
$\in$	Pertinência
$\notin$	Negação da Pertinência
$\subseteq$	Inclusão
$\supseteq$	Relação inversa da inclusão
$\subset$	Inclusão própria
$\supset$	Relação inversa da inclusão própria
$\cup$	União simples
$\cap$	Interseção simples
$(A_i)i \in I$	Família indiciada por I
$\bigcup\limits_{i \in I} A_i$	União de uma família de conjuntos
$\bigcap\limits_{i \in I} A_i$	Interseção de uma família de conjuntos
$\langle x_1, \ldots, x_n \rangle$	n-tupla; conjunto ordenado de n elementos
$A \times B$	Produto cartesiano de A por B
$\wp(A)$	Coleção de subconjuntos de A
$\mathcal{D}(A)$	Domínio de uma relação
$\mathcal{I}(A)$	Imagem de uma relação
$R\langle A \rangle$	Imagem de A por R
$f : A \mapsto B$	Aplicação de A em B
$f : A \rightarrow B$	Função de A em B
$f : A \,)\!\!\!\rightarrow B$	Função parcial de A em B
$f : A \,(\!\!\!\rightarrow B$	Função transformação de A em B

$\rightarrow$	Implicação
$\wedge$	Conjunção
$\vee$	Disjunção
$\neg$	Negação
$\leftrightarrow$	Equivalência
$\top$	Veracidade ou Verum
$\bot$	Falsidade ou Falsum
$\forall$	Quantificação Universal
$\exists$	Quantificação Existencial
$\overline{\exists}$	Quantificador Numérico, significando "existe um máximo"
$\exists!$	Quantificador Numérico, significando "existe um único"
$\approx_c$	Congruência
$\approx$	Isomorfismo
$<$	Menor
$\leq$	Menor ou Igual
$>$	Maior
$\geq$	Maior ou Igual
$+$	Adição
$\times$	Multiplicação
$\{,\}$	Chavetas de conjunto
$\mathbb{N}$	Conjunto dos números naturais
$\mathbb{N}^*$	Conjunto dos números naturais sem o elemento zero
$\mathbb{Z}$	Conjunto dos números inteiros
$\mathbb{R}$	Conjunto dos números reais
$\mathbb{Q}$	Conjunto dos números racionais
$\varnothing$	Conjunto vazio
$\aleph_0$	O número de elementos do conjunto dos números naturais
$\min(A)$	Elemento mínimo de A
$\max(A)$	Elemento máximo de A
$D(x\|t)$	Instanciação de x por t em D
$D(E\|\|G)$	Substituição de E por G em D
$\vdash_{\mathcal{L}}$	Consequência Sintática na Lógica $\mathcal{L}$
$\models_{\mathcal{L}}$	Consequência Semântica na Lógica $\mathcal{L}$
$\#$	Refere-se a um dos conectivos de uma dada lista

$\overrightarrow{\Psi}$	Lista de Quantificadores $\Psi_1, \ldots, \Psi_n$
$\overrightarrow{D}$	Lista de Designadores $D_1, \ldots, D_n$
$\overrightarrow{x}$	Lista de variáveis $x_1, \ldots, x_n$, onde n é um número natural
$\smile$	Este sinal indica que um dado ramo está fechado
$\surd$	Sinal de checagem, significa que um nó já foi usado
$\square$	Fim da demonstração ou da prova
$\mathrm{gr}(P)$	Grau de uma certa fórmula P

Lista de Siglas

IFET	Instituto Federal de Educação Tecnológica
LDC	Lógica Descritiva Clássica
LDI	Lógica das Descrições Indefinidas
LEC	Lógica Equacional Clássica
LI	Logic for Inconsistency
LPC	Lógica Proposicional Clássica
LQC	Lógica Quantificacional Clássica
MG	Minas Gerais
UFSC	Universidade Federal de Santa Catarina
UNIPAC	Universidade Presidente Antônio Carlos

Lista de Notações

τ	Descritor
ϵ	Descritor Indefinido
ι	Descritor Definido
Ψ	Quantificador
Θ	Qualificador
Γ, Δ, φ	Coleções de fórmulas
$\mathcal{L}$	Uma dada Lógica
x, y, z, w	Variáveis
c	Constante
f	Sinal Funcional
p	Sinal Predicativo
D, E, F, G	Designadores
P, Q, R, S	Fórmulas
t, u, v	Termos

Lista de Abreviaturas

Abs	Absorção
AI	Antecedente da Implicação
AIF	Antecedente da Implicação Forte
ant	Conhecimento anterior
Ass	Associatividade
AUn	Lei da Absorção na Unicidade
CE	Comutatividade da Equivalência
Cgr	Congruência
CgT	Congruência Típica
CI	Consequente da Implicação
CMx	Corolário da Lei da Maximidade
CmQ	Comutatividade de Quantificadores
Cm	Comutatividade
CmQT	Comutatividade de Quantificadores Típica
CNENU	Cor. Lei de Neg. Fórmula Exist. e Neg. Fórmula Univ.
Ctp	Contraposição
DC	Dilema Construtivo
DCg	Descrições Congruentes
DE	Descrições Equivalentes
def	Definição
DFQ	Distributividade e Fatorabilidade de Quant.
DFDQ	Distributividade e Fatorabilidade Degeneradas de Quant.
DFQT	Distributividade e Fatorabilidade de Quant. Típicos

DM	Esquema de De Morgan
DN	Esquema da Dupla Negação
DP	Descrição Própria
dsg	Designação
Dt	Distributividade
el	Eliminação
ESE	Esquema da Substituição da Equivalência
EAb	Equivalência Absurda
ESIF	Esquema da Substituição da Igualdade para Fórmulas
FE	Fórmula Existencial
Fs	Falsidade
Fs(i)	Falsidade (i)
Fs(ii)	Falsidade (ii)
Gen	Regra da Generalização
GenT	Regra da Generalização Típica
HGb	Hipótese Global
HI	Hipótese de Indução
hip	Hipótese
IA	Inteligência Artificial
Idp	Lei da Idempotência
IE	Importação/Exportação
II	Esquema da Instanciação da Igualdade para Fórmulas
IM	Implicação Material
Ins	Esquema da Instanciação
int	Introdução
LSC	Lema da Substituição para Conectivos
LSCE	Lema da Substituição para Conectivos e Equivalência

LSCI	Lema da Substituição para Conectivos e Implicação
LSQ	Lema da Substituição para Quantificadores
ME	Lei dos Membros da Equivalência
Mon	Regra da Monotonicidade
MP	Modus Ponens
MPEq	Modus Ponens da Equivalência
MT	Modus Tollens
MX	Lei da Maximidade
NALL	Non Aletic Logic
NC	Lei da Não-Contradição
NCj	Negação da Conjunção
NE	Negação de Fórmula Existencial
NEq	Negação da Equivalência
NET	Negação de Fórmula Existencial Típica
NI	Negação de Implicação
NU	Negação de Fórmula Universal
NUT	Negação de Fórmula Universal Típica
PC	Lei da Prova por Casos
PCL	Paracomplete Logic
PIF	Princípio da Indução Forte
pr	premissa
RC	Regra da Compacidade
RD	Regra da Dedução
RDj	Regra da Redução da Disjunção
Ref	Esquema da Reflexividade
RE	Reflexividade da Equivalência
RI	Reflexividade da Implicação
RSI	Regra de Substituição da Implicação
RSIF	Regra de Substituição da Igualdade para Fórmulas
SD	Silogismo Disjuntivo
SE	Substituição para Equivalência
SFA	Substituição em Fórmulas Atômicas
SH	Silogismo Hipotético
SI	Simetria da Igualdade

spg	Sem perder a generalidade
sss	se, e somente se,
STF	Substituição em Termos Funcionais
STI	Simetria e Transitividade da Igualdade
STR	Sistema de Tablôs por Refutação
sup	suposição
TE	Transitividade da Equivalência
TEx	Terceiro Excluído
TG	Transitividade Geral
TNI	Teoria dos Números Inteiros
TNR	Teoria dos Números Reais
TNN	Teoria dos Números Naturais
TQ	Transporte de Quantificadores
Tran	Transitividade
Un	Esquema da Univocidade
Vc	Esquema da Vacuidade
Vr	Esquema da Veracidade
Vr(i)	Veracidade (i)
Vr(ii)	Veracidade (ii)
ZF	Teoria dos Conjuntos Zermelo-Fraenkel

Convenções Adotadas

Cada capítulo é estruturado em seções e subseções.

O sistema de referências cruzadas aqui utilizado é comum em textos lógicos e matemáticos. Na numeração das proclamações[1] só é dado o número da seção e o número da proclamação. A referência a uma proclamação, dentro do mesmo capítulo, se dá pelo número da seção em que ela está contida e o seu número na seção, e, em um capítulo externo, é incluído o número do capítulo. Por exemplo, o teorema 2.1.1 refere-se ao primeiro enunciado enumerado da primeira seção do capítulo 2 e, dentro do capítulo 2, o mesmo enunciado é referenciado por 1.1. A numeração da seção é precedida do símbolo § e só é dado o número desta. A referência a uma seção, dentro do mesmo capítulo, se dá pelo símbolo de seção "§" e pelo número da seção, e, em um capítulo externo, é incluído o número do capítulo. Por exemplo, a seção §2.1 refere-se à seção 1 do capítulo 2 e, dentro do capítulo 2, a mesma seção é referida por §1.

Indicamos o final de provas de teoremas, possuindo um ou mais parágrafos, pelo símbolo "□".

Toda argumentação presente nos capítulos seguintes faz uso de proclamações elaboradas e um tanto especializadas. Tais proclamações são necessárias para conduzir e fundamentar a teoria dos sistemas lógicos propostos.

Quando queremos falar de um dado objeto devemos utilizar o seu nome, e, quando queremos falar do seu nome, devemos escrevê-lo entre aspas simples. Por exemplo, na frase "Pernambuco é um estado brasileiro", estamos falando do Estado de Pernambuco, e, na frase " 'Pernambuco' possui 10 letras", estamos falando do nome do Estado de Pernambuco, o qual, neste caso, é escrito por 'Pernambuco'.

Os símbolos, as siglas, as notações e as abreviaturas utilizadas estão descritas em suas respectivas listas.

[1] Definições, lemas, teoremas e outros enunciados enumerados.

Resumo

Os principais sistemas relacionados à Lógica Clássica são apresentados.

Outrossim, faz-se uma elucidação da Lógica Proposicional Clássica, da Lógica Quantificacional Clássica, da Lógica Equacional Clássica, da Lógica Descritiva Clássica e da Lógica das Descrições Indefinidas, a fim de possibilitar uma visão inteligível e holística das mesmas. Para cada uma destas lógicas são fornecidas uma linguagem, uma semântica de valorações e um cálculo de sequentes.

Buscamos dar um tratamento geral aos diversos aspectos semânticos e sintáticos, seguindo a perspectiva universal para a construção de ferramentas úteis ao estudo e ao desenvolvimento de lógicas. Dentre estes aspectos está uma abordagem do método dos tablôs por confutação que abstrai as características essenciais desse método em uma generalização cujas instâncias podem ser aplicadas a uma grande variedade de lógicas. Neste trabalho, tal abordagem é aplicada na construção de um sistema de tablôs para a Lógica Proposicional Clássica e para a Lógica Quantificacional Clássica. Também é dado um conceito geral de semântica, o qual corresponde a uma ampla classe de lógicas, e através do mesmo é definido satisfabilidade e relação de consequência. Similarmente, expomos condições gerais de correção e completude dos cálculos de sequentes com respeito às semânticas das lógicas dadas.

Outras contribuições são um tratamento minucioso de algumas questões sintáticas relevantes das diversas linguagens formais, tais como um estudo acurado da instanciação e da substituição, sua aplicação para um estudo dos esquemas e das regras concernentes à equivalência e à implicação, apresentação das listas de teoremas, e algumas provas dos considerados mais importantes aos níveis da lógica clássica proposicional, quantificacional, equacional, descritiva clássica, e das descrições indefinidas.

Abstract

The main systems related to Classical Logic are presented.

Moreover, an elucidation of Classical Propositional Logic, Classical Quantificational Logic, Classical Equational Logic, Classical Descriptive Logic and Logic of Indefinite Descriptions is done, in order to enable the reader a holistic view of them. For each of these logics, are given a language, a semantics of valuations and a sequent calculus.

We aimed to give a general treatment of the various syntactic and semantic aspects, following the universal perspective for the construction of useful tools for the study and development of logics. Among these aspects there is an approach to the tableau method by refutation, which takes away the essential characteristics of this method in a generalization whose instances can be applied to a wide variety of logics. In this work, this approach is applied in the construction of a tableau system for Classical Propositional Logic and Classical Quantificational Logic. Is also given a general concept of semantics, which corresponds to a large class of logics, and through it the satisfiability and the consequence relation are defined. Similarly, we present general conditions of soundness and completeness which a sequent calculus should fulfill in order to be a correct and complete with respect to a semantics of a given logic.

Other contributions area a detailed treatment of some relevant syntactic issues of the various formal languages, such as an accurate study of instantiation and substitution, the application of concerning ideas of the replacement for a study of rules concerning to the implication, and lists of theorems , and some proofs of them, considered important to the levels of classical propositional logic, quantificational, equational, classical descriptive and indefinite descriptions.

Capítulo 1

Introdução

A Lógica é a ciência e a arte do raciocínio. O raciocínio é uma forma de processamento simbólico de informações que busca tornar explícitas formas de conhecimento que antes estavam implícitas. Enquanto ciência, esta possui uma metodologia própria, que, em geral, prioriza as manifestações do raciocínio que surgem no âmbito de contextos linguísticos organizados. Enquanto arte, busca a modelagem de sistemas formais que representem fielmente formas de raciocínio ainda não captadas em toda a sua plenitude. Este trabalho trata da natureza da Lógica e busca deslindar precisamente o seu significado, elucidando os elementos necessários para se formar uma visão inteligível e holística desta.

Este capítulo introduz algumas motivações para o estudo da Lógica e o escopo da mesma, a delimitação do tema, os objetivos específicos, as referências basilares, alguns trabalhos relacionados, e, por fim, a estrutura desta dissertação.

§1. Algumas Motivações para o Estudo da Lógica

O estudo, conhecimento e cultivo da Lógica revelam ferramentas bem importantes para construir argumentos corretos nas diversas áreas do conhecimento com as quais ela se relaciona intimamente, tais como a Matemática, a Filosofia, a Linguística e as Ciências Informáticas[1].

[1] Neste trabalho, o termo 'Ciências Informáticas' designa as seguintes áreas de conhecimento: Ciência da Computação, Engenharia da Computação, Processamento de Dados, Sistemas de Informação, Tecnologia da Informação e afins.

Toda a Matemática faz uso, em sua expressão linguística, de conceitos puramente lógicos; é fundamental entender a Lógica em detalhes para compreender as bases elementares da Matemática. A Lógica é um dos campos da Filosofia, e, por lidar com raciocínios e argumentos, é útil para a reflexão. Existe um ramo de pesquisa em Linguística que se interessa por formas de captar especificações em linguagem natural e representá-las na Lógica Formal, a fim de que possam ser processadas computacionalmente.

Similarmente, as Ciências Informáticas possuem suas raízes na Lógica e fazem amplo uso da mesma, no que concerne, por exemplo, à Arquitetura de Hardware (portas lógicas, design digital), Engenharia de Software (especificação e verificação de sistemas), Linguagens Formais e de Programação (semântica, lógica de programação), Banco de Dados (Álgebra Relacional), Algoritmos (complexidade computacional), Inteligência Artificial (provadores automáticos de teoremas), dentre outras; Além disso, vários dos fundadores da moderna Ciência da Computação foram lógicos, tais como Kurt Gödel, John von Neumann, Alonzo Church, dentre outros.

Essa relação ilustra o quão vasto é o campo de atuação da Lógica, mas não está no escopo deste trabalho tratar de todos esses assuntos. Em vez disso, trataremos da apresentação dos principais conceitos envolvendo os sistemas lógicos clássicos, a fim de que essa elucidação promova um maior esclarecimento dos leitores interessados na Lógica, independentemente do seu uso.

Logo, a conscientização sobre a importância da Lógica e a familiaridade com seus princípios norteadores são fundamentais para o conhecimento atento e aprofundado dos diversos ramos relacionados.

§2. O Escopo da Lógica

Como uma primeira aproximação, podemos dizer que a *Lógica* é a ciência e a arte do raciocínio correto. Enquanto ciência, ela estuda as manifestações da razão operativa em todos os contextos linguísticos possíveis, nas linguagens escrita e falada. O estudo de tais manifestações na vida quotidiana, sem nenhuma idealização, é feito por um ramo desta ciência dito *Lógica Informal*. Um estudo idealizado de tais manifestações, lançando mão de *linguagens formais* e métodos matemáticos, é feito pelo ramo dito *Lógica Formal* ou *Simbólica*. O objeto da *Automatização do Raciocínio* ou *Programação em Lógica* abrange a Lógica Formal, e dá uma ênfase especial ao estudo das rotinas racionais da Lógica Formal expressáveis em *algoritmos*.

Segundo nossa a acepção sobre o assunto, subdividimos o ensino de Lógica Formal em três níveis: elementar, intermediário e avançado.

(1) **Lógica Elementar**
Um conteúdo elementar sobre Lógica normalmente é utilizado em cursos básicos deste assunto para verificar se um argumento é correto ou não. Normalmente os resultados apresentados são proposições enunciando que uma dada coleção de fórmulas acarreta uma fórmula em algum sistema de lógica a nível proposicional ou quantificacional. Segundo [1], este nível é denominado de *lógica clássica de primeira ordem, cálculo clássico de predicados de primeira ordem com igualdade* ou *lógica elementar clássica*. As provas são feitas sem utilizar outros resultados de indução e operadas de maneira primária e simples com conectivos e quantificadores, dispensando requisitos.

(2) **Lógica Intermediária**
Exposição de resultados avançados sobre sistemas lógicos cujas provas envolvem indução matemática e conhecimentos adicionais sobre conjuntos.

(3) **Lógica Avançada**
Exposição de resultados mais avançados sobre sistemas lógicos e provas de resultados envolvendo uma série de outros recursos matemáticos, como, por exemplo, aspectos de teoria da computabilidade, teoria das categorias, dentre outros.

Convém, por oportuno, ressaltar que este trabalho faz uma apresentação *intermediária* dos principais sistemas relacionados à Lógica Clássica.

§3. Delimitação do Tema

Este trabalho é fundamentado na pesquisa bibliográfica que detectou algumas imperfeições, fraquezas e lacunas nos materiais didáticos que tratam da apresentação da Lógica. Alguns destes defeitos basilares para a formulação da situação-problema do trabalho são elucidados a seguir:

(i) Falta um consenso a respeito dos conceitos de instanciação e substituição, os quais são ideias distintas, com aplicações e resultados diferenciados.

(ii) Uma falta de cuidados no uso de teoremas envolvendo equivalência e igualdade pode provocar erros graves[2].

[2]Considerando como hipóteses $\Gamma \vdash t = u$ e $\Gamma \vdash P(x\|t)$, então, sob certas restrições, temos, como conclusão, $\Gamma \vdash P(x\|u)$. Isso não significa que t sempre pode ser trocado por u em P. No caso dessas hipóteses, a restrição é que x não pode estar no escopo em P de nenhuma variável livre em Γ e em t, u. Observamos então, que, as leis de substituição da igualdade e da equivalência, que são tratadas com maiores detalhes adiante, possuem restrições a serem seguidas.

(iii) Também existe uma depreciação dos assuntos considerados elementares, os quais são tratados de forma superficial e insuficientemente prática[3].
(iv) Estes livros carecem de listas expressivas de teoremas elementares no nível dos sistemas lógicos estudados.
(v) A apresentação de alguns conceitos na literatura relacionada utilizam um número excessivo de parênteses, como, por exemplo, em [4].
(vi) Alguns autores, tais como [5], [6] e [2], fixam uma linguagem quando definem conceitos úteis para a apresentação da semântica de uma certa Lógica. Tal abordagem obriga que diversas ideias semânticas sejam associadas a uma dada linguagem, o que é desnecessário.

Diante das motivações destacadas em §1 e das imperfeições citadas anteriormente, o tema norteador desta dissertação é a elucidação dos principais sistemas relacionados à Lógica Clássica. Tais sistemas são listados a seguir:

- Lógica Proposicional Clássica.
- Lógica Quantificacional Clássica.
- Lógica Equacional Clássica.
- Lógica Descritiva Clássica.
- Lógica das Descrições Indefinidas.

Para cada uma destas Lógicas são fornecidos uma classe de linguagens, uma semântica de valorações e um cálculo de sequentes.

§4. Objetivos do Trabalho Proposto

4.1 Objetivo Geral

O objetivo geral deste trabalho é realizar uma análise intermediária de alguns sistemas relevantes da Lógica Formal. Esta análise consiste de uma elucidação acurada e detalhada dos sistemas lógicos LPC, LQC, LEC, LDC e LDI, a fim de conhecer melhor a natureza, as funções e as relações entre os mesmos.

4.2 Objetivos Específicos

Em conformidade com a delimitação do tema, apresentado em §3, os objetivos específicos deste trabalho são:

- Realizar um estudo acurado e detalhado, com todos os resultados consequentes, da instanciação e da substituição, aplicando este conheci-

[3] Referências como [2] e [3] não possuem a exposição de resultados elementares de Lógica.

mento para uma análise de certos aspectos específicos da equivalência e da igualdade.

- Aplicar ideias concernentes à substituição para um estudo de regras concernentes à implicação, assim generalizando as leis lógicas conhecidas comumente como modus ponens e modus tollens.
- Listar leis consideradas importantes, aos níveis da lógica clássica proposicional, quantificacional, equacional, descritiva clássica, e das descrições indefinidas.
- Tratar, com certa minúcia, as questões sintáticas relevantes das diversas linguagens formais.
- Dentro do tópico de lógica quantificacional, realizar um estudo detalhado dos assim chamados quantificadores típicos, os quais refletem a maioria dos contextos em que tais quantificadores são utilizados.
- Dar um tratamento geral a diversos aspectos semânticos e sintáticos, seguindo a perspectiva universal para a construção de ferramentas úteis ao estudo e ao desenvolvimento de lógicas.

O referencial teórico norteador do trabalho é apresentado a seguir.

§5. Estudo Bibliográfico

Os fundamentos de Lógica foram estudados nas referências [2], [5], [7], [8] , [9], [10] , [11] e [12]. Elas são consideradas "portais" para o aprendizado dos conceitos básicos de Lógica.

Os fundamentos sobre Teoria dos Conjuntos foram estudados em [6], [13] , [14] e [15].

Para os sistemas lógicos apresentados na seção §3.2, as seguintes referências foram consultadas:

- Lógicas Paraconsistentes ([16], [17] e [18]).
- Lógicas Paracompletas ([18] e [19]).

As ideias utilizadas na definição de semântica de valorações também encontram-se descritas em [20], que considera a noção de veracidade de uma fórmula em uma interpretação de forma altamente intuitiva para vários propósitos, mas apresenta uma definição formal para o desenvolvimento do que ele denomina *theory of truth.* Em [2], uma semântica proposicional e uma semântica de primeira ordem são apresentadas.

Uma exposição mais detalhada sobre os conectivos *nou* e *nem*, apresentados na página 44, encontra-se em [21], [22] e [23].

O método dos tablôs por refutação foi estabelecido em [24]. Seus principais precursores foram Beth [25] e Hintikka [26]. Trata-se de um processo algorítmico capaz de provar uma tese específica através do fracasso de uma interpretação que satisfaça a negação dessa tese, após uma busca exaustiva. O trabalho apresentado por Smullyan [24] também inspirou muito do que foi realizado posteriormente sobre o método de provas por tablôs. Alguns melhoramentos e correções desta edição foram publicados na versão em português apresentada em [27]. De modo similar, o método direto é também um algoritmo que prova uma tese específica de forma direta, isto é, sem negá-la. O método dos tablôs por prova direta, sem refutação, é apresentado em [28], e é citado na seção §8.2 desta dissertação.

Verbetes, definições de termos lógicos relevantes, e elucidação de significados foram consultados em [8], [29] e [30]. Formulações de teoremas fundamentais da Lógica elementar também encontram-se em [8] e [20].

§6. Trabalhos Relacionados

Este trabalho apresenta três aspectos que podem ser comparados a outras dissertações:

- A análise de um assunto específico, ou seja, um estudo pormenorizado de cada parte de um todo, para conhecer melhor sua natureza, suas funções e relações.
- A preocupação em escrever um conteúdo teórico que possa beneficiar os próprios estudantes de Lógica.
- A preocupação em generalizar as ideias norteadoras do trabalho.

O trabalho crítico e abrangente apresentado por [10] analisa a *História da Computação*. Ele mostra o nascimento da Computação juntamente com as ideias e conceitos que possuem suas raízes na Grécia antiga. Segundo o autor desse trabalho, "um dos objetivos foi produzir um trabalho diferente das teses que se apresentavam, muito pragmáticas, na UnB[4], voltadas à produção de um software"[5].

Em [31], Russell faz uma série de considerações racionais a respeito de um assunto específico, ou seja, ele analisa a natureza da matéria.

§7. Considerações Relevantes

No dizer sempre expressivo de [32], uma possível classificação dos estilos de pesquisas realizadas em Ciência da Computação são: "Apresentação de um Produto","Apresentação de algo Diferente", "Apresentação de algo Presumidamente melhor", "Apresentação de algo relativamente melhor"

[4]Universidade de Brasília.

[5]Trecho de uma mensagem eletrônica enviada por Cléuzio para a autora desta dissertação.

e "Apresentação de uma Prova". Esse último estilo consiste na existência de provas matemáticas, de acordo com as regras da Lógica.

Ainda segundo [32], esses estilos podem ser classificados em três tipos básicos:

- Pesquisas formais, em que é exigida a elaboração de uma teoria e uma prova formal de que essa teoria é correta. A Lógica Formal será a grande ferramenta de trabalho do pesquisador que opta por essa linha;
- Pesquisas empíricas, em que uma nova abordagem apresentada é comparada com outras através de testes aceitos pela comunidade. Os métodos estatísticos serão a grande ferramenta de trabalho do pesquisador que opta por essa linha.
- Pesquisas exploratórias, em que não se consegue provar uma teoria nem apresentar resultados estatisticamente aceitos. Mas entram aqui os estudos de caso, as análises qualitativas e as pesquisas exploratórias em áreas emergentes. A argumentação e o convencimento são as principais ferramentas do pesquisador.

Similarmente, [33] apresenta três paradigmas básicos de pesquisa que os trabalhos em Ciência da Computação podem seguir: o teórico, o empírico e o estético.

No primeiro paradigma existe um argumento e ele é teórico. O espaço do problema é uma teoria, a dissertação é uma proposição, e as provas demonstram que a dissertação segue os axiomas descritos na sua teoria. Tais argumentos são típicos dos trabalhos mais matemáticos e formais da Ciência da Computação.

Considerando os aspectos acima, ressaltamos que este trabalho se trata de uma pesquisa formal. A parte pragmática do trabalho se refere principalmente à construção de provas.

É importante para a Ciência da Computação que existam trabalhos neste sentido formal, em oposição aos trabalhos pragmáticos relacionados à *construção de sistemas*. Desta forma eles podem servir de apoio e referência para estudantes e pesquisadores que possam se interessar por esta área.

A seguir apresentamos a forma como o trabalho é estruturado.

§8. Estrutura do Trabalho

Os aspectos a serem abordados neste trabalho foram discutidos de forma sucinta nas seções anteriores. A organização a seguir fornece uma visão geral e estruturada do modo como o tema será tratado.

A sequência da apresentação obedece à ordem cronológica de desenvolvimento do tema em questão. Optou-se aqui por uma ordem de complexidade do assunto, para uma melhor compreensão dos aspectos e das características dos sistemas lógicos apresentados.

De acordo com [6], a teoria dos conjuntos ocupa uma posição fundamental na construção da matemática, e seus conceitos são usados nos discursos da matemática pura e aplicada, e, portanto, em todas as ciências dedutivas relacionadas a ela. Em particular, a teoria dos conjuntos é usada consideravelmente nas discussões técnicas de Lógica. O propósito do capítulo 2 é apresentar um núcleo mínimo, a respeito da teoria dos conjuntos, essencial aos capítulos posteriores, embasados na Lógica Clássica.

O capítulo 3 é introdutório. Ele apresenta noções gerais sobre sistemas lógicos e em seguida os classifica quanto à profundidade e quanto à forma de raciocínio. Continuando, elucida de forma genérica as ideias de semântica e cálculo de sequentes, as quais serão aplicadas posteriormente para a Lógica Proposicional Clássica, a Lógica Quantificacional Clássica, a Lógica Equacional Clássica e as Lógicas Descritivas.

Conforme [34], quanto maior a expressividade de uma linguagem formal, maior também é a complexidade da manipulação desta linguagem. O capítulo 4 dá início ao estudo da Lógica Clássica, a partir do estudo das linguagens proposicionais. Apresenta, pois, a Lógica Proposicional Clássica. A expressividade desta lógica é limitada, mas a mesma é fundamental para este assunto.

No capítulo 5, sobre a Lógica Quantificacional Clássica, são apresentadas as propriedades clássicas de certas variações internas em fórmulas, expressas pelos *quantificadores*.

O capítulo 6 discorre sobre a Indução Matemática. Este conceito possui um papel muito importante neste trabalho, pois permite a realização de provas mais complexas. Os princípios de indução tratados neste capítulo são constantemente aplicados nos capítulos posteriores.

O capítulo 7 apresenta os resultados gerais de correção e completude de um cálculo de sequentes com respeito a uma semântica de valorações. Tais resultados serão aplicados nas lógicas específicas tratadas neste trabalho.

O capítulo 8 traz uma apresentação geral do método dos tablôs, de uma forma independente da aplicação do mesmo para uma lógica específica, juntamente com provas gerais de correção e completude dos sistemas de tablôs com respeito às semânticas de uma dada lógica.

Os capítulos 9 e 10 apresentam diversos resultados de Lógica Proposicional Clássica e Lógica Quantificacional Clássica, respectivamente, que são tratados via indução matemática.

O capítulo 11 aborda sucintamente uma versão da Lógica Equacional Clássica. Esta Lógica estuda todas as propriedades da igualdade e as suas possíveis interações com os conectivos e com os quantificadores.

Nos capítulos 12 e 13 são apresentadas, respectivamente, as Lógicas das descrições definidas e indefinidas.

Finalmente, no capítulo 14, além das conclusões obtidas, discutimos brevemente algumas perspectivas de desenvolvimento para futuros trabalhos.

Capítulo 2

Teoria Ingênua dos Conjuntos

Apresentamos, de forma intuitiva[1] e concisa, um compêndio das ideias básicas e relevantes sobre Teoria dos Conjuntos consideradas fundamentais para o desenvolvimento deste trabalho: conjuntos, relações, funções, números finitos e infinitos, cardinais e ordinais.

Exposições mais detalhadas sobre *Teoria dos Conjuntos* podem ser encontradas em [15], [14], [6] e [13].

§1. Sinais Lógicos Básicos

Os sinais lógicos listados a seguir são úteis para a exposição dos resultados apresentados neste capítulo. Um estudo mais detalhado dos mesmos será feito nos capítulos seguintes.

- $\neg$
  "x não é par" pode ser reescrito, simbolicamente, por "$\neg$par(x)".
- $\rightarrow$
  "Se x é par, então $3x$ é par" pode ser reescrito por "par$(x) \rightarrow$ par$(3x)$".
- $\wedge$
  "$3 > 1$ e $8 > 3$" pode ser reescrito por "$3 > 1 \wedge 8 > 3$".
- $\vee$
  "x é par ou x é ímpar" pode ser reescrito por "par$(x) \vee$ ímpar(x)".

[1] *Intuitiva* porque faremos uma abordagem direta e imediata. Algo mais conciso e modesto que as exposições de Paul Halmos em [14] e de Machover em [6].

- $\leftrightarrow$
  "$3x$ é par se, e somente se, x é par" pode ser reescrito por
  "$\text{par}(3x) \leftrightarrow \text{par}(x)$".
- $\exists$
  "Existe pelo menos um número que é par" pode ser reescrito por
  "$\exists x\, \text{par}(x)$".
- $\forall$
  "O quadrado de um número par também é par" pode ser reescrito por "$\forall x\, (\text{par}(x) \rightarrow \text{par}(x^2))$".
- $\overline{\exists}$
  "Existe no máximo um número que é par e é primo" pode ser reescrito por "$\overline{\exists} x\, (\text{par}(x) \wedge \text{primo}(x))$".
- $\exists!$
  "Existe um único número que é par e é primo" pode ser reescrito por
  "$\exists! x\, (\text{par}(x) \wedge \text{primo}(x))$".

§2. Teoria Elementar

Intuitivamente falando, um *conjunto* é uma coleção definida e útil para reunir ou agrupar objetos que possuem uma propriedade em comum, e, além disso, é, por si próprio, interpretado como um único objeto.

2.1 Exemplos.
- O conjunto de móveis de uma casa.
- O conjunto dos números naturais.
- O conjunto dos mestrandos em Ciência da Computação na UFSC.

2.2 Notação. (**Conjuntos especiais**)
- $\mathbb{N}$= conjunto dos números naturais, $\{0, 1, 2, 3, \ldots\}$.
- $\mathbb{Z}$= conjunto dos números inteiros, $\{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}$.
- $\mathbb{Q}$= conjunto dos números racionais, $\{m/n : m \in \mathbb{Z} \wedge n \in \mathbb{Z} \wedge n \neq 0\}$.
- $\mathbb{R}$= conjunto dos números reais.

2.3 Notação. Neste capítulo, conjuntos são usualmente denotados pelas letras latinas maiúsculas A, B e C.

A definição de um conjunto pela enumeração de seus elementos consiste em separar os elementos por vírgulas e colocá-los entre chaves. Por exemplo, $A = \{a, b, c\}$.

Se um conjunto é definido através da enunciação da propriedade que seus elementos devem possuir, então se emprega uma letra, a qual representa um objeto qualquer, a propriedade, e, como sinal de pontuação, a barra vertical ($|$), a qual pode ser lida como "tal que". Por exemplo, se A é o conjunto dos alunos da UFSC que são mulheres, então $A = \{x \mid x \text{ é aluno da UFSC} \wedge x \text{ é mulher}\}$. Como um outro exemplo, se B é o conjunto dos elementos x que pertencem a A tal que x é aluno de graduação, então $B = \{x \in A \mid x \text{ é aluno de graduação}\}$.

2.4 Definição.
Os *objetos* de um conjunto são chamados de *membros* ou *elementos* deste conjunto. Assim, tais elementos *pertencem* ($\in$) a este conjunto e este conjunto *possui* ($\ni$) seus elementos[2].

2.5 Exemplo.
Seja $A = \{3, 5, 10\}$. Pela definição acima temos que:
- $3 \in A$, $5 \in A$, $10 \in A$.
- $6 \notin A$.
- $A \ni 3$, $A \ni 5$, $A \ni 10$.
- $A \not\ni 6$.

2.6 Definição.
- $x, y \in A \rightleftharpoons x \in A \wedge y \in A$.
- $x, y, z \in A \rightleftharpoons x \in A \wedge y \in A \wedge z \in A$.

2.7. (**Igualdade de Conjuntos**)
Dois conjuntos são *iguais*, $A = B$, se eles contém os mesmos elementos. Caso contrário, são *diferentes*, ou seja, $A \neq B$.

2.8 Exemplo. Seja $A = \{3, 5, 10\}$, $B = \{10, 3, 5\}$ e $C = \{10, 3, 1\}$. Então, $A = B$, $A \neq C$ e $B \neq C$.

2.9 Definição. (**Subconjunto e Superconjunto**)
A é dito um *subconjunto de* B (ou A está contido em B) se $\forall x(x \in A \to x \in B)$. Notamos isso por $A \subseteq B$. Dizemos também, neste caso, que B é *superconjunto de* A (ou B contém A), e notamos isso também por $B \supseteq A$.

2.10 Definição. (**Subconjunto Próprio e Superconjunto Próprio**)
A é dito um *subconjunto próprio de* B se $A \subseteq B \wedge A \neq B$. Notamos isso por $A \subset B$. Dizemos também, neste caso, que B é *superconjunto próprio de* A, e notamos isso também por $B \supset A$.

2.11 Notação. O conjunto que não possui elementos, ou seja, $\neg\exists x(x \in A)$, é dito *conjunto vazio*. Notamos isso por $\varnothing$.

[2] Em caso contrário, um ou mais elementos não pertencem a este conjunto ($\notin$) e este conjunto *não possui* certo(s) elemento(s) ($\not\ni$).

2.12 Proposição. $\varnothing$ é subconjunto de qualquer conjunto, ou seja, $\varnothing \subseteq A$[3].

2.13 Definição. (**Coleção de Conjuntos**)
A é dito ser uma coleção de conjuntos se as seguintes condições forem satisfeitas:
- A é coleção.
- $\forall B(B \in A \rightarrow B$ é conjunto$)$.

2.14 Definição. (**União e Interseção de uma Coleção de Conjuntos**)
Dada uma coleção A de conjuntos, definimos, nas cláusulas abaixo, a união dos elementos de A, notada por $\bigcup A$, e a interseção dos elementos de A, notada por $\bigcap A$:
- $\bigcup A = \{x | \exists B(B \in A \wedge x \in B)\}$.
- Se $A \neq \varnothing$, então $\bigcap A = \{x | \exists B(B \in A \wedge x \in B)\}$.

2.15 Exemplo.
Seja $A = \{\{1, 3, 7\}, \{3, 9, 7\}, \{7, 3, 10\}\}$. Pela definição acima temos que:
- $\bigcup A = \{1, 3, 7, 9, 10\}$.
- $\bigcap A = \{3, 7\}$.

2.16 Definição. **(Conjunto-Potência)**
O conjunto das partes de um conjunto A é o conjunto de todos os subconjuntos de A. Notamos isso por $\wp(A)$.

2.17 Exemplo.
Seja $A = \{1, 2, 3\}$.
$\wp(A) = \{\varnothing, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$.

2.18 Definição. **(Conjuntos Disjuntos)**
Dois conjuntos A e B são disjuntos se $A \cap B = \varnothing$.

2.1 Operações Fundamentais com Conjuntos

2.19 Definição. **(União)**
A união de A e B, $A \cup B$, é o conjunto dos elementos que estão em A ou em B, ou seja, $A \cup B = \{x | x \in A \vee x \in B\}$.

2.20 Definição. **(Interseção)**
A interseção de A e B, $A \cap B$, é o conjunto dos elementos que estão ao mesmo tempo em A e em B, ou seja, $A \cap B = \{x | x \in A \wedge x \in B\}$.

2.21 Definição. **(Diferença)**
A diferença entre A e B, $A - B$, é o conjunto dos elementos que estão em A e não estão em B, ou seja, $A - B = \{x | x \in A \wedge x \notin B\}$.

[3]Que pode ser provado através da exploração da veracidade vácua.

2.22 Proposições.
(i) $A \cap B \subseteq A$.
(ii) $A \cap B \subseteq B$.
(iii) $A \subseteq A \cup B$.
(iv) $B \subseteq A \cup B$.

As seguintes proposições são equivalentes:

2.23 Proposições.
(i) $A \subseteq B$.
(ii) $A \cap B = A$.
(iii) $A \cup B = B$.

Destacamos algumas proposições que envolvem a operação de união:

2.24 Proposições.
(i) Elemento Neutro da União: $A \cup \varnothing = \varnothing \cup A = A$.
(ii) Comutatividade: $A \cup B = B \cup A$.
(iii) Associatividade: $A \cup (B \cup C) = (A \cup B) \cup C$.
(iv) Idempotência: $A \cup A = A$.

E que, similarmente, envolvem a operação de interseção:

2.25 Proposições.
(i) Elemento de Absorção da Interseção: $A \cap \varnothing = \varnothing \cap A = \varnothing$.
(ii) Comutatividade: $A \cap B = B \cap A$.
(iii) Associatividade: $A \cap (B \cap C) = (A \cap B) \cap C$.
(iv) Idempotência: $A \cap A = A$.

2.26 Proposição.
(i) $A \cap (A \cup B) = A$.
(ii) $A \cup (A \cap B) = A$.

2.27 Proposição.
(i) Se $C \subseteq A$ e $C \subseteq B$, então $C \subseteq A \cap B$.
(ii) Se $A \subseteq C$ e $B \subseteq C$, então $A \cup B \subseteq C$.

§3. Relações e Funções

Relações entre um ou mais objetos são importantes para a especificação de diversas entidades matemáticas. As mesmas estão implicitamente presentes em frases incompletas a serem preenchidas por nomes de um ou mais objetos. Um exemplo de tal frase é "... é pai de ...".

Temos, por exemplo, segundo algumas tradições religiosas, que José é pai de Jesus e Adão é pai de Abel, ou seja, José relaciona-se com Jesus segundo a relação de *ser pai*, e Adão também relaciona-se com Abel segundo a mesma relação.

Podemos considerar também relações entre n objetos para qualquer número natural $n \geq 0$. Por exemplo, um ponto P pode estar ou não entre dois pontos Q e R. Assim, temos a relação de *estar entre*, que é uma possível relação envolvendo três pontos.

Para modelar matematicamente relações em geral, faz-se daí necessário o conceito de *tupla (ordenada)*, que é um agrupamento de um ou mais objetos no qual a ordem é relevante. Consideraremos também, na definição abaixo, o caso da lista ordenada vazia.

3.1 Definição. (**n-tupla, tupla**)
Seja n um número natural. Uma n-tupla é uma lista ordenada de n objetos $x_1, \ldots, x_n$. Notamos a mesma por $\langle x_1, \ldots, x_n \rangle$. Para cada $i \in \{1, \ldots, n\}$, dizemos que x_i é o i-ésimo termo da tupla $x_1, \ldots, x_n$, ou a sua i-ésima coordenada. Uma *tupla* é uma n-tupla, para algum número natural n.

3.2 Proposição.
Se n é um número natural, então
$\langle x_1, \ldots, x_n \rangle = y_1, \ldots, y_n \leftrightarrow x_1 = y_1 \wedge \ldots \wedge x_n = y_n$.

3.3 Definição. (**Par Ordenado, Tripla, Quádrupla, Quíntupla** e **Sêxtupla**)
Um *par ordenado* é uma 2-tupla. Da mesma forma, definimos *tripla (ordenada)*, *quádrupla (ordenada)*, *quíntupla (ordenada)* e *sêxtupla (ordenada)*, como sendo, respectivamente, uma 3-tupla, 4-tupla, 5-tupla e 6-tupla. Dado um par ordenado $\langle a, b \rangle$, dizemos que a é a sua *primeira coordenada* ou *abscissa*, e b é dito a sua *segunda coordenada* ou sua *ordenada*. Dada uma tripla ordenada $\langle a, b, c \rangle$, dizemos, da mesma forma, que a é a sua *primeira coordenada* ou *abscissa* e b é a sua *segunda coordenada* ou sua *ordenada*. Dizemos também, com respeito a esta tripla, que c é a sua *terceira coordenada* ou sua *cota*.

O par ordenado $\langle a, b \rangle$ não deve ser confundido com o conjunto $\{a, b\}$, também denominado *par não-ordenado*, cujos elementos são a e b. Por exemplo, os conjuntos $\{a, b\}$ e $\{b, a\}$ são sempre iguais, mas os pares ordenados $\langle a, b \rangle$ e $\langle b, a \rangle$ são iguais se, e somente se, $a = b$.

A relação de *ser pai* pode, pois, ser matematicamente modelada pelo conjunto $\{\langle x, y \rangle | x \text{ é pai de } y\}$. Neste exemplo, segundo algumas tradições religiosas, temos que os pares $\langle$ José, Jesus$\rangle$ e $\langle$ Adão, Abel$\rangle$ são elementos desta relação.

3.4 Definição. (Relação)
Um conjunto de pares ordenados é dito uma *relação (binária)*. Um conjunto de n-tuplas, onde n é um número natural, é dito *relação n-ária* ou *n-ádica*. Um conjunto de tuplas é dito uma *relação poliádica*.

3.5 Definição. (Aridade de uma Relação)
Se R é uma relação n-ária, dizemos também que n é a *aridade de R*.

3.6 Definição.
Seja R uma relação poliádica. Se $\langle x, y\rangle \in R$, notamos isto, abreviadamente, por xRy. Se $\langle x_1, \ldots, x_n\rangle \in R$, podemos notar isto também por $R(x_1, \ldots, x_n)$.

3.7 Definição.
Seja R uma relação poliádica. Dizemos que x *não está relacionado com y em R*, e notamos isto por $x \not R y$, se $\neg(xRy)$.

3.8 Definição. (Produto Cartesiano)
Dados dois conjuntos A e B, o *produto cartesiano de A e B*, notado por $A \times B$, é o conjunto dos pares ordenados tal que o primeiro componente pertence ao conjunto A e o segundo componente pertence ao conjunto B, ou seja, $A \times B \rightleftharpoons \{\langle x, y\rangle | x \in A \wedge y \in B\}$.

3.9 Definição. R é dito ser uma *relação em A* se $R \subseteq A \times A$.

3.10 Definição. R é dito ser uma *relação de A em B* se $R \subseteq A \times B$.

Uma *relação binária* R de um conjunto A em um conjunto B, é um subconjunto do produto cartesiano $A \times B$.

3.11 Definição. (Tipos de Relações Notáveis com respeito a um certo conjunto)
Sejam R uma relação e A um conjunto. Uma relação R num conjunto A é
- reflexiva em $A \equiv \forall x(x \in A \to xRx)$.
- irreflexiva em $A \equiv \forall x(x \in A \to \neg(xRx))$.
- simétrica em $A \equiv \forall x \forall y(x, y \in A \wedge xRy \to yRx)$.
- assimétrica em $A \equiv \forall x \forall y(x, y \in A \wedge xRy \to y \not R x)$.
- antissimétrica em $A \equiv \forall x \forall y(x, y \in A \wedge xRy \wedge yRx \to x = y)$.
- transitiva em $A \equiv \forall x \forall y \forall z(x, y, z \in A \wedge xRy \wedge yRz \to xRz)$.
- linear[4] em $A \equiv \forall x \forall y(x, y \in A \wedge x \neq y \to xRy \vee yRx)$.

3.12 Exemplos.
- Relação Simétrica: a relação de ‘igualdade’ em um certo conjunto; relação de *ser irmão* no conjunto dos seres humanos.

[4] Uma relação linear também pode ser dita *conexa* ou *tricotômica*.

- Relação Assimétrica: a relação de 'ser menor' no conjunto dos números reais; a relação de 'inclusão própria'.
- Relação Antissimétrica: A relação de 'inclusão' na coleção de todos os conjuntos é antissimétrica na coleção de todos os conjuntos pois o primeiro conjunto está contido no segundo e o segundo está contido no primeiro, então eles são iguais.
- Relação Transitiva: a relação de 'descendência' no conjunto dos seres humanos.
- Relação Linear: a relação 'menor', $<$, no conjunto dos números naturais.

3.13 Escólio. *Toda relação assimétrica é antissimétrica por vacuidade, mas o contrário nunca é verdadeiro.*

3.14 Definição. (**Relação de Ordem (parcial)**)
R é uma *relação de ordem (parcial) em um conjunto* A ou uma *ordem (parcial) em* A se R é uma relação, R é antissimétrica em A e R é transitiva em A.

3.15 Definição. (**Relação de Ordem (parcial) Estrita**[5])
R é *ordem (parcial) estrita* em A se R é ordem em A e R é irreflexiva em A.

3.16 Definição. (**Relação de Ordem (parcial) Reflexiva**)
R é *ordem (parcial) reflexiva* em A se R é ordem em A e R é reflexiva e A.

3.17 Notação.
- Se R é uma ordem estrita em algum conjunto A, notamos R também por '$\prec$'.
- Se R é uma ordem reflexiva em algum conjunto A, notamos R também por '$\preceq$'.

3.18 Definição. (**R-mínimo de A**)
a é R-mínimo de A se $\forall x(x \in A \wedge x \neq a \rightarrow aRx)$.

3.19 Exemplo. Se $<$ é a ordem estrita padrão no conjunto $\mathbb{N}$, então 0 é o $<$-mínimo de $\mathbb{N}$.

3.20 Definição. (**R-máximo de A**)
a é R-máximo de A se $\forall x(x \in A \wedge x \neq a \rightarrow xRa)$.

3.21 Definição. (**Elemento R-Minimal de A**)
a é R-minimal de A se $\forall x(x \in A \wedge x \neq a \rightarrow \neg(xRa))$.

3.22 Definição. (**Elemento R-maximal de A**)
a é R-maximal de A se $\forall x(x \in A \wedge x \neq a \rightarrow \neg(aRx))$.

[5] Ou, irreflexiva.

3.23 Definição.
Se A possui um R-elemento mínimo e este for único, notamos o mesmo por $\min\limits_{R}(A)$, e, se R está implícito, podemos notar isto também por $\min(A)$.
Se A possui um R-elemento máximo e este for único, notamos o mesmo por $\max\limits_{R}(A)$, e, se R está implícito, podemos notar isto também por $\max(A)$.

3.24 Fato. *Se R é uma relação antissimétrica em A, então A possui no máximo um R-elemento mínimo e no máximo um R-elemento máximo.*

3.25 Definição. (**Boa Ordem**)
R é boa ordem em A se R é ordem em A e
$\forall B \subseteq A(B \neq \varnothing \to B$ possui um R-elemento mínimo em $A)$.

3.26 Fato. *Se R é boa ordem em A, então R é linear*[6] *em A.*

3.27 Definição. (**Boa Ordem Estrita**)
R é boa ordem estrita em A se R é boa ordem em A e R é irreflexiva em A.

3.28 Definição. (**Coleção Bem Ordenada**)
$\langle \mathcal{O}, R \rangle$ é uma *coleção bem ordenada* se $\mathcal{O}$ é uma coleção e R é uma boa ordem em $\mathcal{O}$.

3.29 Definição. (**Coleção Estritamente Bem Ordenada**)
$\langle \mathcal{O}, \prec \rangle$ é uma *coleção estritamente bem ordenada* se $\mathcal{O}$ é uma coleção e $\prec$ é uma boa ordem estrita em $\mathcal{O}$.

3.30 Definição. (**R-sucessor de um dado elemento b em A**)
b é *R-sucessor de a em A* se
$a, b \in A \wedge aRb \wedge \forall x(x \in A \wedge x \neq a \wedge x \neq b \to \neg(aRx \wedge xRb))$.

3.31 Definição. (**R-predecessor de um dado elemento b em A**)
a é *R-predecessor de b em A* se
$a, b \in A \wedge aRb \wedge \forall x(x \in A \wedge x \neq a \wedge x \neq b \to \neg(aRx \wedge xRb))$.

3.32 Proposição. b é R-sucessor de a em $A \leftrightarrow a$ é R-predecessor de b em A.

3.33 Definição.
Seja $\langle \mathcal{O}, \prec \rangle$ uma coleção estritamente bem ordenada. n é um *elemento $\prec$-limite em $\mathcal{O}$* se n não possui $\prec$-predecessor em $\mathcal{O}$.

3.34 Definição.
Seja $\langle \mathcal{O}, \prec \rangle$ uma coleção estritamente bem ordenada.
Então $\lim\limits_{\prec} \mathcal{O} = \{n \in \mathcal{O} \mid n$ é *elemento $\prec$-limite em $\mathcal{O}$*$\}$.

---

[6] Conexo.

3.35 Notação. Se $\langle \mathcal{O}, \prec \rangle$ é coleção estritamente bem ordenada e não houver motivo para confusão, então notaremos $\lim_{\prec} \mathcal{O}$ por $\lim \mathcal{O}$.

3.36 Definição. (**Minorante**)
a é *R-minorante*[7] de b se aRb.

3.37 Definição. (**Majorante**)
b é *R-majorante*[8] de a se aRb.

3.38 Proposição. a é R-minorante de $b \leftrightarrow b$ é R-majorante de a.

3.39 Leituras. (**Leituras alternativas para** aRb)

- *a minora b em R*,
- *a R-minora b*,
- a é um *minorante de b em R*,
- a é um *R-minorante de b*,
- *b majora a em R*,
- *b R-majora a*,
- b é um *majorante de a em R*,
- b é um *R-majorante de a*.

3.40 Definição. (**Domínio de uma Relação**)
O *domínio de R* é a coleção de todos os elementos que minoram algum objeto em R, isto é, $\mathcal{D}(R) \equiv \{x | \exists y(xRy)\}$.

3.41 Definição. (**Imagem de uma Relação**)
A *imagem de R* é a coleção de todos os elementos que majoram algum objeto em R, isto é, $\mathcal{I}(R) \equiv \{y | \exists x(xRy)\}$.

3.42 Definição. (**Campo de uma Relação**)[9]
O *campo de R* é a coleção de todos os elementos que minoram ou que majoram algum objeto em R, isto é, $\mathcal{C}(R) \equiv \mathcal{D}(A) \cup \mathcal{I}(A)$.

3.43 Definição.
Seja R uma relação. x é dito um *argumento de R* se $x \in \mathcal{D}(R)$. E x é dito um *valor* de R se $x \in \mathcal{I}(R)$.

---

[7] Ou seja, a R-minora b.
[8] Ou seja, b R-majora a.
[9] O conceito de *campo* é exposto com mais detalhes em [13].

3.44 Definição. (Cota Inferior e Cota Superior)

Sejam $\begin{cases} B \subseteq A, \\ R \text{ uma relação em } A. \end{cases}$

- a é *R-cota inferior de B em A* se $\forall x(x \in B \wedge x \neq a \to aRx)$[10].
- b é *R-cota superior de B em A* se $\forall x(x \in B \wedge x \neq b \to xRb)$[11].

3.45 Exemplo.
Seja $\mathcal{A} = \wp(\mathbb{N})$.
Seja $\mathcal{B} = \{\{0,2,4\},\{0,2,9\},\{0,2,4,9\}\}$.
O conjunto $\{0,2\}$ é $\subseteq$-cota inferior de $\mathcal{B}$ em $\mathcal{A}$.

3.46 Exemplo.
Seja $\mathcal{A} = \wp(\mathbb{N})$.
Seja $\mathcal{B} = \{\{0,2,4\},\{0,2,9\},\{0,2,4,9\}\}$.
O conjunto $\{0,2,4,9,10,11\}$ é $\subseteq$-cota superior de $\mathcal{B}$ em $\mathcal{A}$.

3.47 Definição.

Sejam $\begin{cases} B \subseteq A, \\ R \text{ ordem em } A. \end{cases}$

- B é *R-limitado inferiormente em A* se B possui uma R-cota inferior em A
- B é *R-limitado superiormente em A* se B possui uma R-cota superior em A.
- B é *R-limitado em A* se B é R-limitado inferiormente em A e B é R-limitado superiormente em A.

3.48 Exemplo.
- A coleção $(2,+\infty)$ é limitada inferiormente em $\mathbb{R}$.
- A coleção $(-\infty,-3)$ é limitada superiormente em $\mathbb{R}$.
- A coleção $(2,3)$ é limitada em $\mathbb{R}$.

3.49 Definição. (Ínfimo e Supremo)

Sejam $\begin{cases} B \subseteq A, \\ R \text{ uma relação}. \end{cases}$

Definimos abaixo R-ínfimo e R-supremo de um subconjunto de um conjunto.
- a é *R-ínfimo de B em A* se a é R-cota inferior de B em A e
  $\forall x(x$ é R-cota inferior de B em $A \wedge x \neq a \to xRa)$[12].
- b é *R-supremo de B em A* se b é R-cota superior de B em A e
  $\forall x(x$ é R-cota superior de B em $A \wedge x \neq b \to bRx)$[13].

---

[10] a R-minora qualquer elemento de B distinto de a.
[11] b R-majora qualquer elemento de B distinto de b.
[12] a é uma das maiores R-cotas inferiores de B em A.
[13] b é uma das menores R-cotas superiores de B em A.

3.50 Notação.
Se $\begin{cases} B \subseteq A, \\ R \text{ é uma relação,} \end{cases}$ então adotamos as seguintes notações:

- Se existe um único a tal que a é R-ínfimo de B em A, então podemos notar a por $\underset{R}{inf}(B, A)$, e, se R e A estão implícitos, podemos notar a também por $inf(B)$[14].
- Da mesma forma, se existe um único b tal que b é R-supremo de B em A, então podemos notar b por $\underset{R}{sup}(B, A)$, e, se R e A estão implícitos, podemos notar b também por $sup(B)$.

3.51 Exemplo.
Considere o intervalo de reais $(0, 2)$, isto é, o conjunto de números reais entre 0 e 2, excluindo os números 0 e 2.
Temos, neste caso, que o superconjunto implícito de $(0, 2)$ é a coleção $\mathbb{R}$ de todos os reais, e a relação implícita é a relação de ordem padrão em $\mathbb{R}$, daí notamos o $<$-ínfimo de $(0, 2)$ em $\mathbb{R}$ simplesmente por $inf((0, 2))$ ou $inf(0, 2)$, e também notamos o $<$-supremo de $(0, 2)$ em $\mathbb{R}$ por $sup(0, 2)$. Neste caso, temos que:

- $inf(0, 2) = 0$;
- $sup(0, 2) = 2$.

3.52 Definição. (**Fecho Transitivo**)
Sendo R uma relação, R^t é a menor relação transitiva que contém R[15].

3.53 Exemplo. Seja $R = \{\langle 0, 1\rangle, \langle 1, 3\rangle, \langle 3, 4\rangle\}$.
Temos que $R^t = \{\langle 0, 1\rangle, \langle 0, 3\rangle, \langle 0, 4\rangle, \langle 1, 3\rangle, \langle 1, 4\rangle, \langle 3, 4\rangle\}$.

3.54 Definição. (Relação de Equivalência)
Uma relação R em um conjunto A é dita uma *relação de equivalência* em A se R é uma relação reflexiva, simétrica e transitiva em A.

3.55 Definição. (**Classe de Equivalência**)
Seja R uma relação de equivalência, então $[x]_R = \{y \mid xRy\}$.

3.56 Teorema. $[x]_R = [y]_R$ *sss* xRy.

3.57 Definição. Π é partição se:
(i) Π é coleção de conjuntos não vazios,
(ii) $\forall A \forall B(A, B \in \Pi \to A = B \vee A \wedge B = \varnothing)$.

3.58 Definição. Π é partição de A se Π é partição $\wedge \bigcup \Pi = A$.

---

[14]Neste caso, temos que $inf(B)$ é a maior das R-cotas inferiores de B em A.

[15]Neste trabalho, definida a função de sucessão em sistemas de Peano, a relação de ordem correspondente é o fecho transitivo da sucessão.

3.59 Notação. $\mathcal{A}$ e $\mathcal{B}$ denotam coleções de conjuntos.

3.60 Definição.
$\mathcal{A}$ é *cadeia*[16] se $\forall B \forall C(B, C \in \mathcal{A} \rightarrow B \subseteq C \vee C \subseteq B)$ e $\mathcal{A} \subseteq \mathcal{B}$.

3.61 Definição. $\mathcal{A}$ é *cadeia em* $\mathcal{B}$ se $\mathcal{A}$ é cadeia e $\mathcal{A} \subseteq B$.

3.62 Definição. (R**-cadeia**)
A é R*-cadeia* se $\forall a \forall b(a, b \in A \wedge a \neq b \rightarrow aRb \vee bRa)$[17].

3.63 Definição.
$\mathcal{A}$ é $\preceq$*-cadeia* sss $\preceq$ é ordem reflexiva em A e
$\forall a \forall b(a, b \in \mathcal{A} \rightarrow a \preceq b \vee b \preceq a)$.

3.64 Definição. A é $\preceq$*-cadeia em* B se A é $\leq$-cadeia e $A \subseteq B$.

O Lema de Zorn será utilizado na prova da completude para uma lógica $\mathcal{L}$.

3.65 Lema. *(Zorn) Primeira versão*
Se $\mathcal{A}$ é uma coleção de conjuntos na qual cada cadeia é limitada superiormente, então $\mathcal{A}$ possui um elemento maximal.

3.66 Lema. *(Zorn) Segunda versão*
Seja $\preceq$ uma ordem reflexiva em A.
Se cada $\preceq$-cadeia em A é limitada superiormente em A, então A possui um $\preceq$-elemento maximal.

Informalmente podemos dizer que uma função é uma relação em que cada argumento corresponde a um único valor, ou seja, para cada argumento ela associa um único valor. Por exemplo, a relação entre pais e filhos não é função, pois um pai pode ter mais de um filho.

3.67 Definição. (Função)
f é dito ser uma *função* se as seguintes condições forem satisfeitas:
- f é relação,
- $\forall x \forall y_1 \forall y_2(xfy_1 \wedge xfy_2 \rightarrow y_1 = y_2)$.

3.68 Definição. (Aplicação de uma função a um argumento)
Se f é função e x é um argumento de f, notamos por $f(x)$ o valor de f correspondente a x.

[16]Ou seja, uma cadeia é uma coleção de conjuntos, em que dado dois conjuntos, um está contido no outro. Essa definição geral para cadeia é com respeito à relação de inclusão, mas ela pode ser generalizada para a definição de R-cadeia que veremos posteriormente.

[17]Se Cadeia for definida como uma relação irreflexiva, então é necessário exigir que, para quaisquer dois elementos diferentes, ou o primeiro minora o segundo ou o segundo minora o primeiro.

3.69 Proposição. f é função, $x \in \mathcal{D}(f) \vdash y = f(x)$ sss xfy.

3.70 Definição. (Função de A em B)
f é dito ser uma *função de A em B*, e notamos isso por $f : A \to B$, se as seguintes propriedades forem satisfeitas:
- f é uma função;
- $\mathcal{D}(f) = A$;
- $\mathcal{I}(f) \subseteq B$.

Dizemos também, neste contexto, que B é o *contradomínio de f*.

Existem mais três variantes desta definição, dadas abaixo:

3.71 Definição. (**Função Parcial de** A **em** B)
f é dito ser uma *função parcial de A em B*, e notamos isso por $f : A \rightarrowtail B$, se f é uma função $\wedge\ A \supseteq \mathcal{D}(f)\ \wedge\ \mathcal{I}(f) \subseteq B$.

3.72 Definição. (**Transformação de** A **em** B)
f é dito ser *transformação de A em B*, e notamos isso por $f : A \hookrightarrow B$, se f é função $\wedge\ A \subseteq \mathcal{D}(f)\ \wedge\ f\langle A\rangle \subseteq B$.

3.73 Definição. (Aplicação de A em B)
f é dito ser *aplicação de A em B*, e notamos isso por $f : A \circ\!\!\rightarrow B$, se f é função $\wedge\ f\langle A\rangle \subseteq B$.

Existem certas propriedades notáveis sobre funções, a saber.

3.74 Definição. (**Função Injetiva**)
f é uma *função injetiva* se
$\forall x_1 \forall x_2 (x_1, x_2 \in \mathcal{D}(f) \wedge x_1 \neq x_2 \to f(x_1) \neq f(x_2))$.

3.75 Proposição.
f é injetiva sss $\forall x_1 \forall x_2 (x_1, x_2 \in \mathcal{D}(f) \wedge f(x_1) = f(x_2) \to x_1 = x_2)$.

3.76 Definição. (**Função Injetiva de** A **em** B)
Dizemos que f é uma *função injetiva de A em B*, e notamos isso por $f : A \xrightarrow{inj} B$, se as seguintes condições forem satisfeitas:
(i) $f : A \to B$.
(ii) $\forall x_1 \forall x_2 (x_1, x_2 \in A \wedge x_1 \neq x_2 \to f(x_1) \neq f(x_2))$.

3.77 Proposição.
$\vdash f : A \xrightarrow{inj} B \leftrightarrow$
$f : A \to B \wedge \forall x_1 \forall x_2 (x_1, x_2 \in A \wedge f(x_1) = f(x_2) \to x_1 = x_2)$.

3.78 Definição. (**Função Sobrejetiva de A em B**)
Dizemos que f é uma *função sobrejetiva de A em B*, e notamos isso por $f : A \overset{sobre}{\longrightarrow} B$, se as seguintes condições forem satisfeitas:
(i) $f : A \to B$.
(ii) $\forall y \in B\, \exists x \in A, y = f(x)$.

3.79 Proposição. As seguintes proposições são equivalentes:
(i) $f : A \overset{sobre}{\longrightarrow} B$.
(ii) $f : A \to B \wedge \mathcal{I}(f) = B$.

A *Teoria das Categorias* generaliza a ideia de função através do conceito de morfismo, bem como as ideias de injetividade e sobrejetividade. Ela possui fortes associações com a Ciência da Computação. Mais detalhes podem ser obtidos em [35], [36] e [37].

3.80 Definição. (**Função Bijetiva de A em B**)
Dizemos que f é uma *função bijetiva de A em B*, e notamos isso por $f : A \overset{bij}{\longrightarrow} B$, se $f : A \overset{inj}{\longrightarrow} B$ e $f : A \overset{sobre}{\longrightarrow} B$. Dizemos também, neste caso, que f é uma *correspondência biunívoca entre A e B*.

3.81 Definição. (**Equipolência entre dois conjuntos**)
Dizemos que dois conjuntos A e B são equipolentes, e notamos isso por $A \approx B$, se existe uma bijeção de A em B.

3.82 Definição. (**Aridade de uma função**)
A *aridade* de uma função é o número de argumentos ou operandos tomados. Uma função unária mapeia um argumento, uma função binária mapeia dois elementos, uma função ternária mapeia três elementos e uma função n-ária mapeia n argumentos.

3.83 Definição. (**Função Identidade**)
Seja A uma coleção.
A coleção $I_A = \{\langle x, x\rangle \mid x \in A\}$ é chamada de *identidade em A* e é notada por I_A.

3.84 Definição. (**Função Escolha**)
f é uma *função-escolha*[18] de A, se, para cada $B \neq \varnothing$ tal que $B \subseteq A$, $f(B) \in B$.

[18] A função-escolha depende do domínio em questão.

3.85 Definição. (Função Característica[19])
Considere as coleções A e B tais que $A \subseteq B$. A *função característica de A com respeito a B* é a função f onde

$$\begin{aligned} f\ &: A \to \{0,1\} \\ x\ &\mapsto 1, \text{se } x \in A \\ x\ &\mapsto 0, \text{se } x \notin A \end{aligned}$$

3.86 Definição. (Família de Conjuntos)
A é dito ser uma *família de conjuntos* indexada por I se A é uma função de I para uma coleção de conjuntos. Neste caso, notamos A também por $(A_i)_{i\in I}$.

3.87 Exemplo.
Seja $I = \{1,2,3\}$.

$$\begin{aligned} A\ &: I \to \{\{1\},\{1,2\},\{1,2,3\}\} \\ 1 &\mapsto \{1\} \\ 2 &\mapsto \{1,2\} \\ 3 &\mapsto \{1,2,3\} \end{aligned}$$

Podemos notar A por $(\{1,\ldots,i\})_{i\in\{1,2,3\}}$.

3.88 Definição. (União e Interseção de uma Família de Conjuntos)
Seja $(A_i)_{i\in I}$ uma família de conjuntos.
$\bigcup\limits_{i\in I} A_i = \bigcup\{A_i | i \in I\}$.
Se $I \neq \varnothing$, então $\bigcap\limits_{i\in I} A_i = \bigcap\{A_i | i \in I\}$.

3.89 Proposição.
Se $(A_i)_{i\in I}$ é uma família de conjuntos, então:
- $\bigcup\limits_{i\in I} A_i = \{x | \exists i (i \in I \wedge x \in A_i)\}$.
- Se $I \neq \varnothing$, então $\bigcap\limits_{i\in I} A_i = \{x | \forall i (i \in I \to x \in A_i)\}$.

3.90 Exemplo. Conforme o exemplo 3.87, temos que:
- $\bigcup\limits_{i\in I} A_i = \{1,2,3\}$.
- $\bigcap\limits_{i\in I} A_i = \{1\}$.

[19] Também conhecida por Função Indicadora.

3.91 Definição. (**Sucessão Induzida**)
Seja R uma boa ordem estrita em A.
s é a *sucessão induzida* por R em A se as seguintes condições forem satisfeitas:
(i) $s : A \rightarrowtail A$.
(ii) $\forall x(x \in \mathcal{D}(s) \leftrightarrow \exists y(x \prec y))$.
(iii) $\forall x \forall y(x \in \mathcal{D}(s) \to (x \prec y \leftrightarrow s(x) \preceq y))$.

3.92 Proposição.
Se $\begin{cases} \bullet\ \prec \text{ é boa ordem estrita em } \mathcal{O}, \\ \bullet\ s \text{ é a sucessão induzida por } \prec \text{ em } \mathcal{O}, \\ \bullet\ a \in \mathcal{D}(s), \end{cases}$
então $s(a)$ é o único $\prec$-sucessor de a em $\mathcal{O}$ e a é o único $\prec$-predecessor de $s(a)$ em $\mathcal{O}$.

§4. Cardinais e Ordinais

Existem duas generalizações dos números naturais, que foram descobertas pelo matemático alemão Georg Cantor em [38], que são os números cardinais e os números ordinais. Os primeiros são utilizados para medir o número de elementos de um dado conjunto. Para conjuntos finitos, esta medida é dada pelos números naturais, os quais são um caso particular dos números cardinais. Para conjuntos infinitos esta medida é dada por números cardinais que não são números naturais.

Os segundos são utilizados para rotular cada um dos elementos de uma coleção bem ordenada, sendo que, quando tal coleção é finita, os rótulos utilizados são os próprios números naturais, e, quando tal coleção é infinita, os rótulos utilizados são números ordinais que não são números naturais.

4.1 Definição. Se A é um conjunto, notamos o *número de elementos de* A, também chamado de *cardinalidade de* A, por '$\#(A)$'.

4.2. **Lei Básica da Cardinalidade**
Se A e B são conjuntos, então $\#(A) = \#(B)$ sss $A \approx B$.

Os números cardinais que não são números naturais, também chamados de *números cardinais infinitos*, são notados, sucessivamente, por $\aleph_0, \aleph_1, \aleph_2, \ldots$ onde $\aleph_0$ é o número de elementos do conjunto dos números naturais, $\aleph_1$ é o menor cardinal que majora $\aleph_0$, $\aleph_2$ é o sucessor de $\aleph_1$[20].

[20] Aleph, $\aleph$, é a primeira letra do alfabeto grego.

4.3 Exemplos.

- $\#(\{8, 5, 6\}) = 3$.
- $\#(\{16, 7, 9, 4, 8\}) = 5$.
- $\#(\mathbb{N}) = \aleph_0$.
- $\#(\varnothing) = 0$.
- $\#$ (coleção de problemas computáveis)$= \aleph_0$.
- $\#$ (coleção de problemas não computáveis)$= \#(\mathbb{R})$.

Os conceitos de problemas computáveis e não computáveis podem ser vistos com maiores detalhes em [39, 40].

Os *números ordinais* ditos *finitos* são os próprios números naturais. O primeiro *ordinal infinito*, isto é, não finito, é notado por ω. Os ordinais infinitos são notados, em ordem crescente, segundo a sequência abaixo

$\omega, \omega + 1, \omega + 2, \ldots, \omega + \omega (= \omega.2)$
$\omega.2, \omega.2 + 1, \omega.2 + 2, \ldots, \omega.2 + \omega (= \omega.3)$
$\omega, \ldots, \omega.2, \ldots, \omega.3, \ldots, \omega.\omega (= \omega^2)$
$\omega, \ldots, \omega^2, \ldots, \omega^3, \ldots, \omega^4, \ldots, \omega^\omega$, e assim por diante.

4.4 Exemplo.
Dado o conjunto $\{8, 5, 9, 4\}$ com a boa ordem $8 \prec 5 \prec 4 \prec 9$, então, conforme esta ordem, podemos rotular os elementos deste conjunto por:

- $8 \mapsto 0$
- $5 \mapsto 1$
- $9 \mapsto 2$
- $4 \mapsto 3$

4.5 Exemplo.
Dado o conjunto dos números naturais $\mathbb{N}$, considerando a ordem padrão, cada número natural é rótulo de si próprio, isto é:

- $0 \mapsto 0$
- $1 \mapsto 1$
- $2 \mapsto 2$
- $\ldots$

4.6 Exemplo.
Considerando novamente o conjunto dos números naturais $\mathbb{N}$, com a ordem alternativa $0 \prec 2 \prec 4 \prec \ldots \prec 1 \prec 3 \prec 5 \prec \ldots$, cada número natural é associado a um número ordinal de acordo com a seguinte sequência:
- $0 \mapsto 0$
- $2 \mapsto 1$
- $4 \mapsto 2$
- $\ldots$
- $1 \mapsto \omega$
- $3 \mapsto \omega + 1$
- $5 \mapsto \omega + 2$
- $\ldots$

Em Teoria dos Conjuntos existem diferentes maneiras de construir números cardinais e números ordinais. Adotaremos, neste trabalho, a construção devida a John von Neumann, apresentada, por exemplo, em [6, 13, 15], da qual expomos, a seguir, os fatos essenciais.

4.7 Definição. Um dado *conjunto* é dito ser *transitivo* se as seguintes condições forem satisfeitas:
(i) A é uma coleção de conjuntos.
(ii) $\forall B(B \in A \to B \subseteq A)$.

4.8 Exemplo. Se $A = \{\{\varnothing, \{\varnothing\}\}, \varnothing, \{\varnothing\}\}$, temos que A é um conjunto transitivo, pois todos os seus elementos são também seus subconjuntos.

4.9 Notação. α, β e γ representam ordinais.

4.10 Definição. (**Número Ordinal**)
Dizemos que α é um *número ordinal* se as duas seguintes cláusulas forem atendidas:
(i) α é um conjunto transitivo.
(ii) '$\in$' é uma boa ordem em A.

4.11 Exemplo.
Nem todo conjunto transitivo é bem ordenado pela relação de pertinência. Se $A = \{\{\varnothing, \{\varnothing\}\}, \{\{\varnothing\}\}, \varnothing, \{\varnothing\}\}$, então temos que A é um conjunto transitivo. Temos também que $\{\varnothing, \{\varnothing\}\}$ e $\{\{\varnothing\}\}$ são elementos de A, mas nenhum deles pertence ao outro, daí A não é bem ordenado pela relação de pertinência, logo A não é um ordinal.

4.12 Definição.
Dados dois ordinais α e β, dizemos que $\alpha < \beta$ (pela ordem padrão), se $\alpha \in \beta$.

4.13 Proposição. Se α é um número ordinal, então $\alpha = \{\beta | \beta < \alpha\}$.

4.14 Definição. Dizemos que α é um *ordinal limite* se α não possui um $\in$-predecessor.

4.15 Definição.
Os *ordinais finitos* são definidos sucessivamente pelas seguintes cláusulas:
- $0 \rightleftharpoons \varnothing$
- $1 \rightleftharpoons \{0\}$
- $2 \rightleftharpoons \{0, 1\}$
- $\vdots$
- $n + 1 \rightleftharpoons \{0, \ldots, n\}$

4.16 Definição. (**Conjunto Finito e Conjunto Infinito**)
Um dado *conjunto* é dito ser *finito* se existe um ordinal finito α tal que $A \approx \alpha$; em caso contrário, tal *conjunto* é dito ser *infinito*.

4.17 Definição. (**Número Cardinal**)
λ é dito ser um *número cardinal* se λ é um ordinal finito ou λ é um ordinal limite.

4.18 Proposição. Todo conjunto está em correspondência biunívoca com um único número cardinal.

4.19 Definição.
Sejam $\begin{cases} \lambda \text{ e } \mu \text{ cardinais,} \\ A \text{ e } B \text{ conjuntos disjuntos tais que } \#(A) = \lambda \text{ e } \#(B) = \mu. \end{cases}$
Então $\lambda + \mu = \#(A \cup B)$.

4.20 Definição.
Sejam $\begin{cases} \lambda \text{ e } \mu \text{ cardinais,} \\ A \text{ e } B \text{ conjuntos tais que } \#(A) = \lambda \text{ e } \#(B) = \mu. \end{cases}$
Então $\lambda \times \mu = \#(A \times B)$.

4.21 Proposição. Sejam λ e μ cardinais.
- Se $\lambda \prec \mu$ e μ é cardinal infinito, então $\lambda + \mu = \mu$.
- Se $\lambda \prec \mu$, $\lambda \neq 0$ e μ é cardinal infinito, então $\lambda . \mu = \mu$.

Capítulo 3

Introdução à Lógica Geral

Este capítulo apresenta uma visão holística das definições e convenções acerca de lógicas, não se atendo a uma lógica específica.

Ele fundamenta a argumentação das demais lógicas apresentadas no restante deste trabalho. Trata-se, portanto, de um conteúdo que fornece os conceitos gerais que serão aplicados nos capítulos posteriores, de acordo com as características de cada lógica considerada.

§1. Linguagens para a Lógica

Uma *Lógica* ou *Sistema Lógico* é definida em duas etapas:

- A primeira etapa estabelece uma ou mais linguagens formais para essa lógica através da exposição de um alfabeto e de uma gramática, com o propósito de alcançar as suas expressões significativas.
- A segunda etapa estabelece uma teoria associada a estas linguagens, a fim de verificar quais expressões significativas são verdadeiras ou não.

O funcionamento da razão operativa está essencialmente ligado à obtenção de frases condicionalmente verdadeiras a partir de frases hipoteticamente verdadeiras, deste modo, a compreensão e uso de pelo menos uma *linguagem* parece ser imprescindível.

Podemos classificar as linguagens em dois tipos básicos:

- *Linguagens naturais* ou *informais* – correspondem às línguas usadas habitualmente pelos seres humanos para a sua comunicação quotidiana, tais como o português, o inglês, dentre outras. A sua descrição necessita, em geral, de livros com centenas de páginas, pois as regras de suas gramáticas são complexas e numerosas.

- *Linguagens artificiais* ou *formais* – as suas regras gramaticais são simples e, em geral, em pequeno número; algumas, especialmente aquelas utilizadas pela lógica, detém um poder expressivo comparável ao das linguagens naturais.

  Uma linguagem artificial é definida em duas etapas:
- escolha de um conjunto não vazio de sinais (em geral, gráficos), chamado de *alfabeto*, a serem usados na construção de suas *expressões significativas*;
- enunciação de uma (em geral pequena) coleção de regras (isto é, uma *gramática*) destacando, entre as expressões da linguagem, quais são significativas.

Em uma linguagem natural tais etapas ocorrem de um modo bem mais complexo. Todas as linguagens naturais modernas, em sua versão escrita, também possuem um alfabeto. Os sinais deste alfabeto formam expressões significativas básicas, ditas *palavras* desta linguagem. Algumas destas palavras são nomes para objetos reais ou imaginários, possuindo funções análogas aos *termos* de uma linguagem formal para a lógica. A coleção de todas as palavras utilizadas por uma linguagem natural é dita *vocabulário*, o qual pode existir implícita ou explicitamente. Neste último caso pode haver uma publicação oficial, editada por alguma instituição mantenedora das normas desta linguagem, contendo uma relação completa de todas as palavras constantes do seu vocabulário. Finalmente, certas sequências de palavras de uma linguagem natural formam outro tipo de expressões significativas, ditas *frases*, as quais correspondem, grosso modo, às *fórmulas* de uma linguagem formal utilizada por alguma lógica.

Para um estudo das regularidades presentes nas diversas formas de raciocínio, bem como dos possíveis algoritmos correspondentes, as linguagens formais são as mais adequadas, devido à sua simplicidade inerente.

1.1 Notação. L é dita uma *linguagem formal*.

1.2 Definição. Um *alfabeto* é uma coleção não vazia de sinais.

1.3 Definição.
Uma *samblagem*[1] é uma sequência finita de sinais. Uma *samblagem em um dado alfabeto* é uma sequência finita de sinais deste alfabeto. A lista vazia de sinais é dita a *samblagem vazia*; notamos isto por 'Λ'.

1.4 Definição.
Dadas duas samblagens u e v, a concatenação de u e v, notada por $u + v$, é a samblagem obtida justapondo-se, da esquerda para a direita, preservando a ordem original, os sinais de u com os sinais de v.

[1]Termo também conhecido, na língua inglesa, por "string".

1.5 Definição.
Se u e v são samblagens, especificamos a seguir o que entendemos por *iniciar uma samblagem* e por *terminar uma samblagem*.

- u inicia $v \rightleftharpoons$ existe uma samblagem u' tal que $u + u' = v$.
- u termina $v \rightleftharpoons$ existe uma samblagem u' tal que $u' + u = v$.
- u inicia propriamente $v \rightleftharpoons u \neq \Lambda$ e u inicia v e $u \neq v$.
- u termina propriamente $v \rightleftharpoons u \neq \Lambda$ e u termina v e $u \neq v$.

1.6 Exemplo.
Sejam $p_1, \ldots, p_n$ letras sentenciais em LPC, a qual será apresentada em 4.
Seja ‘$)\, p_1 \rightarrow \wedge\, p_2$’ a samblagem v e seja ‘$)p_1$’ a samblagem u.
Podemos dizer, pela definição acima, que ‘$)p_1$’ inicia ‘$)\, p_1 \rightarrow \wedge\, p_2$’, ou seja, u inicia v.
Seja ‘$\rightarrow \wedge\, p_2$’ a samblagem w. Dizemos então, neste caso, que w termina v, porque existe uma samblagem u concatenada com w que resulta na samblagem v.
Ainda pela definição anterior, podemos dizer que a samblagem vazia, Λ, inicia e termina a samblagem v, pois:

- $v + \Lambda = v =)\, p_1 \rightarrow \wedge\, p_2$.
- $\Lambda + v = v =)\, p_1 \rightarrow \wedge\, p_2$.

1.7 Definição. (Samblagens Significativas)
As *samblagens significativas de uma linguagem formal* em uma dada lógica são samblagens de um dos dois tipos básicos:

- *termos* – são nomes de objetos do universo de discurso, na linguagem formal considerada;
- *fórmulas* – são afirmações ou asserções, na linguagem formal considerada, acerca de objetos do universo de discurso.

Todos os *sistemas lógicos*, os quais serão definidos em §2, com exceção das *lógicas proposicionais*, possuem linguagens suficientemente ricas cujas samblagens significativas são termos e fórmulas.

1.8 Notação. Adotaremos como convenção, em todo este trabalho, as seguintes referências para as listas de letras abaixo, seguidas ou não de plicas ou subíndices, com respeito a uma linguagem formal para uma lógica:

- t, u, v – referem-se a termos em L;
- P, Q, R, S – referem-se a fórmulas em L;
- Γ, Δ, φ – referem-se a coleções de fórmulas em L.

§2. Sistemas Lógicos

Definimos um *sistema lógico* ou *lógica* em duas fases:

- Fornecemos uma *gramática*, a qual especifica as *linguagens formais deste sistema*. Cada uma destas linguagens formais possui a sua coleção própria de fórmulas.
- Para cada uma destas linguagens formais, fornecemos uma *teoria*, a qual especifica, dentre as fórmulas do sistema, quais devem ser consideradas absolutamente verdadeiras[2], e, para cada contexto de fórmulas supostas hipoteticamente verdadeiras, quais são verdadeiras neste contexto.

Quanto à profundidade do raciocínio envolvido, existem sistemas lógicos concebidos para diferentes níveis, a saber:

- *Lógica Proposicional* – estuda relações simples entre fórmulas, as quais correspondem a expressões tais como ‘se–então’, ‘e’, ‘ou’, ‘não é verdade que’, dentre outras.
- *Lógica Quantificacional* – além das relações citadas acima, estuda o comportamento de certas variações de referências feitas no interior das fórmulas. Tais variações correspondem às expressões do tipo ‘para todo’, ‘para cada’, ‘para algum’, ‘existe pelo menos um’, ‘nenhum’, dentre outras.
- *Lógica Equacional* – além do que é feito nos dois níveis acima, lida com a equivalência de referências. Corresponde às expressões do tipo ‘é igual a’, ‘é idêntico a’, dentre outras.
- *Lógica Descritiva* – além do que é feito nos níveis anteriores, lida também com especificações, ambíguas ou não, de objetos do universo de discurso. Trata expressões do tipo ‘um objeto x possuindo a propriedade P’, ‘o objeto x possuindo a propriedade P’.
- *Teoria dos Conjuntos* – é o substrato comum em que se baseia toda a matemática tradicional, e geralmente estuda as propriedades comuns em coleções.

Quanto à forma do raciocínio envolvido, existem, entre outros, os seguintes sistemas lógicos:

- *Lógica Clássica* – é o sistema lógico que serve de base implícita para quase toda a matemática; até meados do século XX era praticamente o único sistema existente, e ainda hoje, no século XXI, é considerado um padrão de raciocínio correto. A Lógica Clássica possui, entre outros, os seguintes princípios, formulados originalmente por Aristóteles:
  - ∗ Toda fórmula ou a sua negação é verdadeira, ou seja, não há uma terceira condição possível. Este é dito o *princípio do terceiro excluído*.
  - ∗ Uma fórmula e a sua negação não podem ser ambos verdadeiros. Este é dito o *princípio da não contradição*.

[2]Veracidade absoluta e veracidade relativa são definidas em 2.6.

$*$ Uma fórmula verdadeira é sempre verdadeira, e uma fórmula falsa é sempre falsa; ou seja, o valor veritativo de uma fórmula com respeito a uma dada interpretação é estável e permanente. Este é dito o *princípio da identidade*.

- *Lógicas Paracompletas* – são lógicas que não respeitam o princípio do terceiro excluído e onde '$P \vee \neg P$' não é sempre verdadeiro. Uma Lógica Paracompleta especial, a qual tem sido extensivamente estudada, e que talvez seja a única que rivaliza seriamente, em importância, com a Lógica Clássica, é a *Lógica Intuicionista*.
- *Lógicas Paraconsistentes* – não respeitam o princípio da não contradição, e, por isso, são ideais para lidar com *contradições*. Nas mesmas, uma dada contradição não acarreta qualquer fórmula, isto é, '$P \to \neg P \to Q$' não é sempre verdadeiro.
- *Lógicas Não Reflexivas* – são aquelas que não respeitam o princípio da identidade. Nelas pode acontecer de uma fórmula não implicar a si própria, ou seja, '$P \to P$' não é necessariamente uma tese.
- *Lógicas Relevantes* – são aquelas que interpretam a implicação de uma forma mais estrita, na qual é suposto haver algum tipo de relacionamento estrutural ou causal entre o antecedente e o consequente de uma implicação considerada verdadeira.
- *Lógicas Não Monotônicas* – são aquelas nas quais o acréscimo de novas premissas pode invalidar conclusões já obtidas. Até o ano de 1970, praticamente só eram conhecidas as Lógicas Monotônicas. As *Lógicas Indutivas* são um dos gêneros mais importantes destas lógicas.
- *Lógicas Modais* – são aquelas que estudam as possíveis variações da veracidade ou falsidade ao longo de certas entidades denominadas *mundos*. O *valor veritativo de uma fórmula* pode variar ao longo dos mundos. Duas das principais ideias estudadas por tais lógicas são a *necessidade* e a *possibilidade*.

Os sistemas que divergem da Lógica Clássica são ditos *deviantes*, como, por exemplo, as Lógicas Paraconsistentes, Paracompletas, Não Reflexivas, Relevantes e Não Monotônicas. As Lógicas Modais podem erigir-se tanto sobre a Lógica Clássica (quando neste caso a enriquecem em expressividade), como sobre algum sistema deviante.[3]

Após definir as linguagens de um dado sistema lógico, a segunda etapa para defini-lo completamente consiste em fornecer uma relação entre coleções de fórmulas e fórmulas do mesmo, que preserve a ideia de veraci-

[3]Não há uma distinção completamente nítida entre o que é *deviante* e o que é *complementar* à Lógica Clássica, pois é possível, em muitos casos, fazer conviver no mesmo sistema, formas deviantes e clássicas de negação, de implicação, e assim por diante. Assim sendo, certos sistemas, que poderiam ser classificados, a princípio, como deviantes, podem possuir extensões conservativas que são complementares à Lógica Clássica.

dade.

Existem três vias básicas para isto:

- A *via sintática* parte de certos *objetos formais iniciais* e *regras* que fornecem, de um modo em geral mecânico, novos objetos a partir de uma lista finita de objetos formais já obtidos. Para as lógicas usuais, existem três caminhos para isto:
  * por *cálculos axiomáticos* ou *sistemas de Hilbert* – os objetos formais são *fórmulas*, as *fórmulas iniciais* são ditas *axiomas* e as *regras* associam listas finitas de fórmulas a fórmulas;
  * por *cálculos de sequentes* – os objetos formais são pares ordenados, ditos *sequentes*, cujo primeiro componente é uma coleção de fórmulas e cujo segundo componente é uma fórmula; os *sequentes iniciais* representam inferências consideradas verdadeiras a priori e as regras associam listas finitas de sequentes a sequentes;
  * por *dedução natural* – os objetos formais são *árvores de fórmulas*, onde algumas folhas podem estar marcadas; as folhas não marcadas representam premissas de uma inferência e a raiz representa a conclusão da mesma inferência; as *árvores iniciais* representam inferências consideradas verdadeiras a priori, e as *regras* associam listas finitas de tais árvores a novas árvores de fórmulas.
- A *via semântica* baseia-se em um *domínio veritativo*, isto é, em uma coleção de rótulos representando diferentes *gradações de verdade e falsidade*, e em uma coleção de funções, denominadas *valorações*, apresentando certas propriedades básicas, que podem variar de lógica a lógica, associando *fórmulas* a *valores veritativos* (elementos do domínio veritativo).
- A *via da automatização* baseia-se em *algoritmos* cujas *entradas* são sequentes e cujas *saídas* são possíveis respostas que avaliam cada sequente considerado. Para as lógicas usuais, existem três métodos básicos:
  * o método da *resolução* – a partir de um processamento simbólico inicial, obtém-se uma lista de *fórmulas normalizadas* ditas *cláusulas*. Tal lista é sucessivamente ampliada através de uma regra especial, dita *regra da resolução*, visando à derivação da *cláusula vazia*. Se a mesma for encontrada, o sequente inicial é considerado correto.
  * o método dos *tablôs* – a partir de uma árvore de fórmulas inicial, obtida em geral pela negação da conclusão do sequente inicial, é realizada uma expansão sucessiva da mesma, visando ao fechamento de todos os seus ramos. Se isto for conseguido, então o sequente inicial é considerado correto.
  * o método dos *sequentes de Gentzen* – a partir do sequente inicial é construída uma árvore de sequentes especiais, ditos *sequentes de Gentzen*. O objetivo é obter o fechamento de todos os seus ramos e,

então, se isto for conseguido, o sequente inicial é considerado correto.

Por qualquer uma das vias acima citadas, uma lógica define uma *relação de consequência* entre coleções de fórmulas e fórmulas nesta lógica. Tal relação de consequência exprime tanto a *verdade absoluta* como a *verdade relativa* da lógica considerada, definidas a seguir.

2.1 Notação. No restante deste trabalho, a não ser que seja dito explicitamente o contrário, $\mathcal{L}$ é um dado sistema lógico ou lógica.

2.2 Definição.
Seja L a coleção de todas as fórmulas de $\mathcal{L}$.
Uma coleção Γ de fórmulas em L é um subconjunto de L, ou seja, $\Gamma \in \wp(L)$.
Uma fórmula P de L é um elemento de L, ou seja, $P \in L$.
A relação de consequência de $\mathcal{L}$, notada por '$\vdash_{\mathcal{L}}$', relaciona coleções de fórmulas em $\mathcal{L}$ com fórmulas em $\mathcal{L}$, daí temos que '$\vdash_{\mathcal{L}}$' é um subconjunto de $\wp(L) \times L$, ou seja, $\vdash_{\mathcal{L}} \subseteq \wp(L) \times L$.

2.3 Notação.
Notamos por '$\vdash_{\mathcal{L}}$' a relação de consequência definida em $\mathcal{L}$. Esta expressão é usada para denotar a relação de consequência em $\mathcal{L}$ de um modo geral, quando não são dados maiores detalhes a respeito de como tal relação é especificada, e também quando tal relação de consequência é obtida por uma via sintática. Usamos também a expressão alternativa $\models_{\mathcal{L}}$ para denotar a relação de consequência em $\mathcal{L}$, quando esta é especificada através de uma via semântica.

2.4 Definição. (**Sequente**)
Se L é a coleção de todas as fórmulas de $\mathcal{L}$, então um *sequente em* $\mathcal{L}$ é dito um elemento de $\wp(L) \times L$, ou seja, é um par ordenado $\langle \Gamma, P \rangle$, onde Γ é uma coleção de fórmulas em $\mathcal{L}$ e P é uma fórmula em $\mathcal{L}$. Dizemos, neste caso, que as fórmulas de Γ são as *premissas* deste sequente e P é dito a sua *conclusão*.

2.5 Definição. (**Sequente Correto**)
Um *sequente* $\langle \Gamma, P \rangle$ em $\mathcal{L}$ é dito ser *correto em* $\mathcal{L}$ se $\langle \Gamma, P \rangle \in \vdash_{\mathcal{L}}$. Notamos isto também por $\Gamma \vdash_{\mathcal{L}} P$. Usamos também a expressão '$\Gamma \vdash_{\mathcal{L}} P$' para denotar o sequente $\langle \Gamma, P \rangle$, se este for correto.

2.6 Definição. (Veracidade de uma fórmula em uma lógica)
'$\Gamma \models_{\mathcal{L}} P$' significa que P é *verdadeiro em* $\mathcal{L}$ se todas as fórmulas de Γ também o forem. Se $\Gamma \neq \varnothing$, '$\Gamma \models_{\mathcal{L}} P$' exprime a *veracidade relativa de* P *em* $\mathcal{L}$ *dependendo de* Γ. Por outro lado, '$\varnothing \models_{\mathcal{L}} P$' exprime a *veracidade absoluta de* P, ou seja, que P *é verdadeiro em* $\mathcal{L}$. Neste caso, dizemos também que P é uma *tese de* $\mathcal{L}$.

2.7 Leituras. '$\Gamma \models_{\mathcal{L}} P$' pode ser lido de uma das seguintes formas:
- "Γ *acarreta* P em $\mathcal{L}$".
- "*De* Γ, *em* $\mathcal{L}$, *afirma-se* P".
- "P *é consequência de* Γ *em* $\mathcal{L}$".
- "P *é teorema de* Γ *em* $\mathcal{L}$".

2.8 Notação. Se $\mathcal{L}$ está implícita, podemos prescindir da indicação da lógica envolvida, notando '$\Gamma \models_{\mathcal{L}} P$' simplesmente por '$\Gamma \models P$'. Mais algumas notações:
- $\models_{\mathcal{L}} P \rightleftharpoons \varnothing \models_{\mathcal{L}} P$;
- $P_1, \ldots, P_n \models_{\mathcal{L}} P \rightleftharpoons \{P_1, \ldots, P_n\} \models_{\mathcal{L}} P$;
- $\Gamma, P \models_{\mathcal{L}} Q \rightleftharpoons \Gamma \cup \{P\} \models_{\mathcal{L}} Q$;
- $\Gamma_1, \ldots, \Gamma_n, P_1, \ldots P_r \models_{\mathcal{L}} Q \rightleftharpoons \Gamma_1 \cup \ldots \cup \Gamma_n \cup \{P_1, \ldots, P_r\} \models_{\mathcal{L}} Q$.

2.9 Definição.
Γ é *trivial em* $\mathcal{L}$ se $\Gamma \models_{\mathcal{L}} P$, para toda fórmula P em $\mathcal{L}$.[4] Uma fórmula P é dita *trivial em* $\mathcal{L}$ se $\{P\}$ é trivial em $\mathcal{L}$.

2.10 Definição. (Esquema)
Um *esquema em* $\mathcal{L}$ é uma coleção de sequentes em $\mathcal{L}$. Os elementos de um esquema em $\mathcal{L}$ são ditos serem os seus *exemplares* ou *instâncias*.

2.11 Exemplo.
Exemplo geral de esquema em uma lógica $\mathcal{L}$:
$\{P \models_{\mathcal{L}} P \vee Q \mid P, Q$ são fórmulas em $\mathcal{L}\}$.
Alguns exemplares deste esquema:
- $P \models_{\mathcal{L}} P \vee Q$,
- $P \to Q \models_{\mathcal{L}} (P \to Q) \vee R$.

2.12 Definição. (Esquema correto)
Um *esquema* é dito *correto em* $\mathcal{L}$ se todos os seus exemplares forem corretos em $\mathcal{L}$.

[4] Se $\mathcal{L}$ for a Lógica Clássica e Γ for trivial em $\mathcal{L}$, Γ também é dito *inconsistente em* $\mathcal{L}$, pois, neste caso, $\Gamma \models_{\mathcal{L}} P$ e $\Gamma \models_{\mathcal{L}} \neg P$.

2.13 Definição. (Aplicação)
Uma *aplicação em* $\mathcal{L}$ é um par $\langle \Sigma, \zeta \rangle$, onde Σ é uma lista finita de sequentes em $\mathcal{L}$ e ζ é um sequente em $\mathcal{L}$; os elementos de Σ são ditos serem *hipóteses da aplicação*, e ζ é dito ser *a conclusão da aplicação*.

2.14 Exemplo.
Exemplo geral de aplicação em uma lógica $\mathcal{L}$:
$\langle\langle \Gamma_1 \vdash_{\mathcal{L}} P_1, \ldots, \Gamma_n \vdash_{\mathcal{L}} P_n \rangle, \varphi \vdash_{\mathcal{L}} Q \rangle$.
Exemplo de uma aplicação específica:
$P, Q \vdash_{\mathcal{L}} R,\ \ P, Q \vdash_{\mathcal{L}} S,\ \ P, Q \vdash_{\mathcal{L}} R \wedge S.$

Que de modo prático pode ser notado assim $\frac{P,Q \vdash_{\mathcal{L}} R,\ P,Q \vdash_{\mathcal{L}} S}{P,Q \vdash_{\mathcal{L}} R,S}$.

2.15 Definição. (Aplicação correta)
Uma aplicação $\frac{`\Gamma_1 \vdash P_1, \ldots, \Gamma_n \vdash P_n\text{'}}{\vartheta \vdash Q}$ em $\mathcal{L}$ é dita *correta em* L se a correção de todas as suas hipóteses em $\mathcal{L}$ implicar na correção de sua conclusão em L.

2.16 Definição. (Regra)
Uma *regra em* $\mathcal{L}$ é uma coleção de aplicações em $\mathcal{L}$.

2.17 Exemplo.
$$\left\{ \frac{\Gamma \vdash_{\mathcal{L}} P,\ \Gamma \vdash_{\mathcal{L}} Q}{\Gamma \vdash_{\mathcal{L}} P \wedge Q} \mid \Gamma \text{ é coleção de fórmulas e } P, Q \text{ são fórmulas em LPC} \right\}$$

Todo esquema pode ser visto também, grosso modo, como uma regra, que associa uma lista vazia de sequentes a cada sequente deste esquema.

2.18 Definição. (Regra correta)
Uma regra em $\mathcal{L}$ é dita ser *correta em* L se todas as suas aplicações forem corretas em L.

2.19 Definição. (Lei)
Uma *lei em* $\mathcal{L}$ é um esquema em $\mathcal{L}$ ou uma regra em $\mathcal{L}$.

2.20 Definição. (Lei correta)
Uma *lei* é dita *correta em* $\mathcal{L}$ se esta for um esquema correto em $\mathcal{L}$ ou uma regra correta em $\mathcal{L}$.

O ideal, para uma dada lógica, está na obtenção de descrições sintáticas, semânticas e computacionais da mesma, e na demonstração da equivalência de todas estas descrições. Assim, na pesquisa das propriedades desta lógica, pode-se usar as ferramentas mais adequadas em cada caso.

Segundo [2], no tratamento de sistemas lógicos nós devemos fazer uma distinção entre *sintaxe* e *semântica*. Questões sintáticas são puramente formais e se preocupam com expressões da lógica $\mathcal{L}$ enquanto sequências de símbolos, independentemente de qualquer interpretação. A semântica, por outro lado, está preocupada com o significado e o sentido que expressões de $\mathcal{L}$ recebem quando os símbolos que ocorrem nas mesmas são interpretados de alguma forma.

Segundo [30], as inferências dos métodos sintáticos dependem de regras que consideram apenas a estrutura das fórmulas, e não a sua interpretação.

Ao longo de todo este trabalho, a via sintática preferencial será o método dos cálculos de sequentes, a via da automatização será o método dos tablôs, e a via semântica será detalhada em §3.

§3. Semântica de Valorações

Nesta seção expomos uma abordagem geral sobre semântica de valorações para sistemas lógicos, de acordo com a qual, através de alguns componentes previamente dados, uma semântica completa para uma lógica é fornecida.

Esta abordagem é aplicada para lógicas específicas nas seções §4.2, §11.2, §12.4 e §13.2, a fim de definir uma semântica de valorações para LPC, LQC, LEC, LDC e LDI, respectivamente.

A seguinte pré-definição apresenta os cinco componentes essenciais que uma valoração semântica deve conter a fim de especificar a satisfabilidade, a validade e a relação de consequência da semântica de uma lógica.

3.1 Pré-Definição.
Para diversas lógicas $\mathcal{L}$, consideram-se conhecidos os seguintes tipos de objetos, os quais especificam *uma semântica de valorações para* $\mathcal{L}$:

- uma coleção de valores veritativos de $\mathcal{L}$, os quais são valores de veracidade ou valores de falsidade, atribuídos às fórmulas em $\mathcal{L}$.
- uma subcoleção da anterior, cujos elementos atribuem às fórmulas em $\mathcal{L}$ valores de veracidade; estes elementos são chamados *valores distinguidos de* $\mathcal{L}$; os outros elementos da coleção de valores veritativos são chamados *valores não distinguidos de* $\mathcal{L}$, os quais atribuem às fórmulas em $\mathcal{L}$ valores de falsidade.
- $\mathcal{L}$*-interpretações*, que destinam-se a dar informações suficientes para atribuir valores de veracidade para algumas fórmulas em $\mathcal{L}$.
- uma relação entre $\mathcal{L}$-interpretações e fórmulas em $\mathcal{L}$, de acordo com a qual, uma dada $\mathcal{L}$-interpretação é ou não uma $\mathcal{L}$*-interpretação para uma dada*

fórmula em $\mathcal{L}$. Se I é uma $\mathcal{L}$-interpretação para uma fórmula P em $\mathcal{L}$, então P é dito uma *I-fórmula em $\mathcal{L}$*.

- Uma função que associa cada $\mathcal{L}$-interpretação I a outra função I_V, chamada de *$\mathcal{L}$-valoração especificada por I*, onde I_V é uma função da coleção de I-fórmulas para a coleção de valores veritativos de $\mathcal{L}$.

3.2 Definição.
Seja $\mathcal{L}$ uma lógica dotada de semântica de valorações e seja I uma $\mathcal{L}$-interpretação:

- *I satisfaz P em $\mathcal{L}$* se P é uma I-fórmula e $I_V(P)$ é um valor distinguido de $\mathcal{L}$.
- *I satisfaz Γ em $\mathcal{L}$* se I satisfaz cada fórmula de Γ.
- P é *$\mathcal{L}$-válido* se todas $\mathcal{L}$-interpretações para P satisfizerem P, em caso contrário, P é dito ser *$\mathcal{L}$-inválido*.
- Γ é *$\mathcal{L}$-satisfatível* se existe pelo menos uma $\mathcal{L}$-interpretação que satisfaz Γ, em caso contrário, Γ é dito ser *$\mathcal{L}$-insatisfatível*.
- *Γ acarreta P em $\mathcal{L}$* se toda $\mathcal{L}$-interpretação para $\Gamma \cup \{P\}$ que satisfaz Γ satisfaz P. Notamos isto por $\Gamma \models_{\mathcal{L}} P$.

3.3 Definição.
P é uma $\mathcal{L}$-consequência semântica de Γ se toda $\mathcal{L}$-interpretação para $\Gamma \cup \{P\}$ que satisfaz Γ também satisfaz P. Notamos isto por $\Gamma \models_{\mathcal{L}} P$.

Quando Γ é uma coleção finita de fórmulas em $\mathcal{L}$ e P é uma fórmula em $\mathcal{L}$, o problema de decidir se $\Gamma \models_{\mathcal{L}} P$ pode ser reduzido, na maioria das lógicas conhecidas, incluindo a Lógica Clássica, ao problema de decidir se uma dada fórmula é $\mathcal{L}$-satisfatível.

§4. Cálculo de Sequentes

A nossa via sintática preferencial será o *método dos cálculos de sequentes*[5].

Um cálculo de sequentes é um tipo de sistema dedutivo, ou seja, um sistema que possibilita inferir, derivar ou deduzir as consequências de uma coleção de fórmulas em uma dada lógica.

4.1 Definição.
Um cálculo de sequentes para $\mathcal{L}$ é definido por *leis*, as quais são ditas serem as suas *leis primitivas*.
As leis primitivas do cálculo são ditas serem os seus *postulados*.

[5] Os Cálculos de Sequentes foram definidos por Gerhard Gentzen em [41].

Uma *demonstração* neste cálculo é uma lista de sequentes em que cada sequente ou é um exemplar de um esquema primitivo ou é uma consequência de sequentes anteriores nesta lista, por aplicação de uma regra primitiva.

4.2 Escólio. *Uma demonstração em um cálculo de sequentes é uma sequência construtiva na geração da coleção de sequentes corretos deste cálculo.*

4.3 Definição.
Os *esquemas primitivos* de um cálculo de sequentes para $\mathcal{L}$ são coleções de sequentes considerados corretos, a priori, em $\mathcal{L}$.
E as *regras primitivas* deste cálculo são relações fornecidas, a priori, entre listas finitas de sequentes e sequentes que preservam a correção em $\mathcal{L}$.

4.4 Definição.
Um esquema correto em $\mathcal{L}$, que não for primitivo em $\mathcal{L}$, é dito *derivado em* $\mathcal{L}$.
Uma regra correta em $\mathcal{L}$, que não for primitiva em $\mathcal{L}$, é dita *derivada em* $\mathcal{L}$.
Uma lei correta em $\mathcal{L}$, que não for primitiva em $\mathcal{L}$, é dita *derivada em* $\mathcal{L}$.

Um cálculo de sequentes trabalha com sequentes corretos em $\mathcal{L}$ e utiliza regras que possam garantir a correção dos mesmos.

As condições gerais de correção de um cálculo de sequentes com respeito a uma semântica de valorações de uma dada lógica, neste trabalho, são apresentadas em §7.3.

4.5 Exemplo.
Seja L uma linguagem formal cujos sinais são de um dos seguintes tipos:
- $p_1, p_2, p_3, \ldots$[6]
- $\neg, \to$[7].
- '(', ')'[8].

Uma fórmula de L é definida pelas seguintes cláusulas:
- Uma letra sentencial é fórmula de L.
- Se P é fórmula de L, então $\neg P$ é fórmula de L.
- Se P e Q são fórmulas de L, então $(P \to Q)$ é fórmula de L.

Quando $(P \to Q)$ não estiver no interior de outra fórmula de L, podemos prescindir do par exterior de parênteses.
Considere L a linguagem da lógica $\mathcal{L}_0$, daí as fórmulas em $\mathcal{L}_0$ são as fórmulas de L.
Damos a seguir um cálculo de sequentes para $\mathcal{L}_0$:

[6]Veremos depois, que tais sinais são chamados, usualmente, de *letras sentenciais*, e que são usados para representar frases elementares de uma linguagem natural.

[7]Estes sinais são chamados, usualmente, de *conectivos*, os quais formam frases a partir de frases.

[8]Estes sinais são chamados, usualmente, de *sinais de pontuação*, e sao usados para evitar ambiguidade na escrita de fórmulas.

- Esquema da Reflexividade: Se $P \in \Gamma$, então $\Gamma \vdash_{\mathcal{L}_0} P$.
- Regra Modus Ponens: $\dfrac{\Gamma \vdash_{\mathcal{L}_0} P \quad \Gamma \vdash_{\mathcal{L}_0} P \to Q}{\Gamma \vdash_{\mathcal{L}_0} Q}$.
- Regra do $\neg$-introdução: $\dfrac{\Gamma, P \vdash_{\mathcal{L}_0} Q \quad \Gamma, P \vdash_{\mathcal{L}_0} \neg Q}{\Gamma \vdash_{\mathcal{L}_0} \neg P}$.

Uma demonstração em $\mathcal{L}_0$.

.

1	$\neg Q, P \to Q, P \vdash_{\mathcal{L}_0} \neg Q$	Ref
2	$\neg Q, P \to Q, P \vdash_{\mathcal{L}_0} P \to Q$	Ref
3	$\neg Q, P \to Q, P \vdash_{\mathcal{L}_0} P$	Ref
4	$\neg Q, P \to Q, P \vdash_{\mathcal{L}_0} Q$	2,3,MP
5	$\neg Q, P \to Q \vdash_{\mathcal{L}_0} \neg P$	4,1,$\neg$-int

$\square$

Capítulo 4

Lógica Proposicional Clássica Elementar

Este capítulo apresenta a *Lógica Proposicional Clássica* ou simplesmente LPC. Essa lógica estuda certas relações externas entre fórmulas, expressas pelos *conectivos*.

Adotaremos aqui a notação especificada em 3.1.8, e as letras p, q, r serão utilizadas conforme a notação 1.2.

§1. Uma Linguagem para LPC

Definir uma *linguagem proposicional* é o primeiro passo para podermos estabelecer um sistema formal para LPC.

1.1 Definição.
Um *alfabeto proposicional* é uma coleção de símbolos que contém os seguintes tipos de sinais, mutuamente disjuntos dois a dois:

- *letras sentenciais* – são em si próprias, *fórmulas em* LPC; podemos considerá-las como referências a frases de qualquer gênero proferíveis em alguma linguagem. Todo alfabeto proposicional deve possuir pelo menos uma letra sentencial.
- *conectivos* – formam fórmulas a partir de fórmulas; se um conectivo usa n fórmulas para formar uma fórmula, dizemos que n é a *aridade* deste sinal; os conectivos são em geral de aridades 1 ou 2. Todo alfabeto proposicional deve possuir pelo menos um conectivo.
- *sinais de pontuação* – são '(', o parêntese de abertura, e ')', o parêntese de fechamento. Se tal alfabeto possuir pelo menos um conectivo de aridade

maior ou igual a 3, então ',', a vírgula, também é um sinal de pontuação.

1.2 Notação. Doravante, a menos que seja dito o contrário, as letras p, q, r, seguidas ou não de plicas ou subíndices, referem-se a letras sentenciais em LPC.

1.3 Convenção. Uma letra sentencial é uma letra sentencial em qualquer linguagem L de LPC, ou seja, letras sentenciais possuem a mesma funcionalidade em todas as linguagens de LPC.

1.4 Convenção. Uma linguagem para LPC pode ser desenvolvida utilizando apenas os conectivos de negação ($\neg$) e implicação ($\rightarrow$), assim como Moché Machover apresenta em seu "Set theory, logic and their limitations" no capítulo sobre Lógica Proposicional. Por razões didáticas os seguintes conectivos são adotados para o desenvolvimento da LPC:

1.5 Definição. Um alfabeto para LPC é um alfabeto proposicional possuindo os conectivos:
- '$\rightarrow$', lê-se *implicação*,
- '$\wedge$' , lê-se *conjunção*,
- '$\vee$' , lê-se *disjunção*,
- '$\neg$' , lê-se *negação*,
- '$\top$' , lê-se *verum*,
- '$\bot$' , lê-se *falsum*.

1.6 Notação. A partir de agora, a menos que o contrário seja dito, usaremos o sinal '#' para nos referir a um dos conectivos '$\rightarrow$','$\wedge$', '$\vee$'.

Tendo como fundamento os conectivos apresentados em 1.5, novos conectivos diádicos podem ser definidos:
- $(P \leftrightarrow Q) \rightleftharpoons ((P \rightarrow Q) \wedge (Q \rightarrow P))$[1].

A fórmula '$(P \leftrightarrow Q)$' é também dita a *equivalência de* P *e* Q, cujos *membros* são P e Q.
- $P \,\bar{\vee}\, Q \rightleftharpoons \neg(P \leftrightarrow Q) \rightleftharpoons (P \vee Q) \wedge \neg(P \wedge Q)$.

A fórmula '$P \,\bar{\vee}\, Q$' é chamada de *disjunção exclusiva* (de P *e* Q), e lê-se 'ou P ou Q'. Declarar 'ou P ou Q' significa dizer que uma, e somente uma das duas fórmulas P e Q é verdadeira.
- $P \downarrow Q \rightleftharpoons \neg(P \vee Q)$.

A fórmula '$P \downarrow Q$' é chamada de *negação conjunta (de* P e Q), e lê-se 'nem P nem Q' ou simplesmente 'P nem Q'[2]. Declarar 'nem P nem Q' significa dizer que ambas as fórmulas P e Q são falsas.

[1] Isto é, a expressão '$(P \leftrightarrow Q)$' é uma abreviatura para a expressão '$((P \rightarrow Q) \wedge (Q \rightarrow P))$'.

[2] Na língua inglesa, a fórmula '$P \downarrow Q$' pode ser lida como 'neither P nor Q', ou, simplesmente, 'P nor Q'.

- $P \uparrow Q \rightleftharpoons \neg(P \wedge Q)$.
  A fórmula '$P \uparrow Q$' é chamada de *negação alternativa ou negação disjunta (de* P e Q), e lê-se 'P nou Q'[3]. Declarar 'P nou Q' significa dizer que pelo menos uma das fórmulas P e Q é falsa.

Os símbolos $\downarrow$ e $\uparrow$ também são chamados de *conectivos de Scheffer*.

1.7 Definição.
Seja s um conectivo. Se s possui aridade n, dizemos também que s é n*-ário* ou n*-ádico*. Se s possui uma das aridades 1, 2 ou 3, então s é chamado respectivamente de *monádico*, *diádico* ou *triádico*.

Os conectivos *verum* e *falsum* são de aridade zero. Como os mesmos são *zerários*, ambos são, por si próprios, fórmulas em LPC. Temos que '$\top$' denota a veracidade e '$\bot$' denota a falsidade em LPC[4].

1.8 Teorema. *As cláusulas abaixo especificam os conectivos '$\top$' e '$\bot$':*
- $\top \rightleftharpoons p_i \rightarrow p_i$;
- $\bot \rightleftharpoons \neg(p_i \rightarrow p_i)$, *onde* p_i *é uma letra sentencial em* LPC *escolhida arbitrariamente.*

O conectivo da negação é monádico ou unário. Os demais são considerados conectivos diádicos ou binários.

1.9 Definição. (Fórmulas Contraditórias)
Seja $\mathcal{L}$ uma lógica possuindo '$\neg$' como um conectivo monádico. Duas fórmulas em $\mathcal{L}$ são ditas *contraditórias* se uma delas for a negação da outra.

1.10 Definição. (Fórmulas Equivalente)
Seja $\mathcal{L}$ uma lógica na qual qualquer linguagem para a mesma possui '$\rightarrow$' e '$\wedge$' como conectivos. Duas fórmulas P e Q são ditas serem *equivalentes em* $\mathcal{L}$ se $P \leftrightarrow Q$ for uma tese em $\mathcal{L}$.

1.11 Definição.
Uma *fórmula em* LPC é uma samblagem em um alfabeto para LPC definida pelas regras de formação de fórmulas da linguagem proposicional[5]:
- Toda letra sentencial é uma fórmula em LPC, dita *fórmula atômica*.
- Se P é uma fórmula em LPC, então $\neg P$ é uma fórmula em LPC, dita a *negação de* P.

[3] Para a língua inglesa existe o neologismo 'nand'. Assim '$P \uparrow Q$' pode ser lido 'P nand Q'.

[4] Eles serão úteis quando da apresentação do método dos tablôs e os conceitos de representação do conhecimento para LPC.

[5] Parece acertado que nem toda samblagem é uma fórmula, mas toda fórmula é uma samblagem.

- Se P e Q são fórmulas em LPC, então:
  - $*$ $(P \to Q)$ é uma fórmula em LPC, dita a *implicação* de P e Q, cujo *antecedente* é a fórmula P, e cujo *consequente* é a fórmula Q.
  - $*$ $(P \wedge Q)$ é uma fórmula em LPC, dita *a conjunção* de P e Q, cujos *conjuntores* são P e Q.
  - $*$ $(P \vee Q)$ é uma fórmula em LPC, dita *a disjunção* de P e Q, cujos *disjuntores* são P e Q.
  - $*$ $\top$ é fórmula em LPC e representa qualquer fórmula que é sempre verdadeira em LPC.
  - $*$ $\bot$ é fórmula em LPC e representa qualquer fórmula que é sempre falsa em LPC.

1.12 Comentário. De acordo com [29], uma proposição ou fórmula atômica (nos sistemas de Lógica elementar) é (1) uma letra sentencial isolada; ou (2) uma letra predicado n-ádica, seguida de n termos. Intuitivamente, é uma sentença não "decomponível". Por exemplo, em "O gato dorme na lareira" e "Caim é irmão de Abel" não existem conectivos que permitem *desdobrar* essas sentenças para tê-las como combinações de outras mais *curtas*.

1.13 Exemplo. São exemplos de fórmulas em LPC:
- q_7,
- $(p_3 \vee q_7)$,
- $\neg(p_6 \vee (p_5 \to p_7))$,
- $\bot$,
- $\top$,
- $\bot \wedge p_3$.

1.14 Definição. (**Condição Necessária e Condição Suficiente segundo [29] e [30]**)
Considere a implicação condicional $P \to Q$.

Temos que $\begin{cases} \bullet\ Q \text{ é condição necessária de (ou para) } P, \\ \bullet\ \text{Uma condição necessária aparece como consequente de uma} \\ \quad\ \text{implicação condicional.} \end{cases}$

Um exemplo de condição necessária: 'Todo mineiro é brasileiro'. Ser brasileiro é condição necessária para ser mineiro, mas não é suficiente pois não basta ser brasileiro para ser mineiro, é necessário nascer no estado de Minas Gerais.

Temos que $\begin{cases} \bullet\ P \text{ é condição suficiente de (ou para) } Q, \\ \bullet\ \text{Uma condição suficiente aparece como antecedente de uma} \\ \quad\ \text{implicação condicional.} \end{cases}$

Um exemplo de condição suficiente: 'Todo mineiro é brasileiro'. Ser mineiro é condição suficiente para ser brasileiro, mas não é necessário pois é possível ser brasileiro sem necessariamente ser mineiro.

Chama-se *condição necessária e suficiente* à conjunção de uma condição necessária com uma condição suficiente, o que garante que tudo o que é P é Q, e vice-versa.

1.15. **Leituras de Fórmulas em** LPC

- '$\neg P$' pode ser lido como:
  - *'Não P'*,
  - *'Não é verdade que P'*,
  - *'Não é o caso que P'*.
- '$P \rightarrow Q$' pode ser lido de uma das seguintes formas:
  - 'Se P, então Q',
  - 'P implica Q',
  - 'Q se P',
  - 'P só se Q',
  - 'P somente se Q',
  - 'P é suficiente para Q',
  - 'Q é necessário para P',
  - 'P é condição suficiente para Q',
  - 'Q é condição necessária para P',
  - 'A ocorrência de P implica em Q'.
- '$P \wedge Q$' é lido como '*P e Q*'.
- '$P \vee Q$' é lido como '*P ou Q*' no sentido inclusivo, isto é, declarar '*P ou Q*' neste sentido significa admitir também a veracidade de ambas as fórmulas.
- '$P \leftrightarrow Q$' é lido de uma das seguintes formas:
  - 'P equivale a Q',
  - 'P é equivalente a Q',
  - 'P se, e só se, Q',
  - 'P se, e somente se, Q',
  - 'P é necessário e suficiente para Q',
  - 'P é uma condição necessária e suficiente para Q'.

1.16. **Escrita informal de fórmulas em** LPC

Sempre que não houver margem para confusão, podemos escrever termos e fórmulas em LPC de um modo mais informal, evitando um uso excessivo de parênteses, segundo as convenções:

- Quando $(P \rightarrow Q)$, $(P \wedge Q)$ e $(P \vee Q)$ não estão escritas como subfórmulas de outra fórmula, podemos prescindir o par exterior de parênteses.
- A lista '$\{\wedge, \vee, \bar{\vee}, \downarrow, \uparrow\}, \rightarrow, \leftrightarrow$' fornece a ordem de prioridade para parentetização de fórmulas.

  Exemplo: $P \vee Q \rightarrow R \wedge S \leftrightarrow P_1 \downarrow P_2$ representa $((P \vee Q) \rightarrow (R \wedge S)) \leftrightarrow (P_1 \downarrow P_2)$. E $P \wedge Q \vee R \rightarrow S_1 \rightarrow S_2 \wedge S_3 \uparrow S_4$ representa $((P \wedge Q) \vee R) \rightarrow (S_1 \rightarrow ((S_2 \wedge S_3) \uparrow S_4))$.

- Inversamente, a lista '$\leftrightarrow, \rightarrow, \{\wedge, \vee, \bar{\vee}, \downarrow, \uparrow\}$' fornece a ordem de prioridade para a separação em subfórmulas, como, por exemplo, em '$P{\leftrightarrow}Q{\vee}R{\rightarrow}S$', que representa '$P \leftrightarrow ((Q \vee R) \rightarrow S)$'.
- Quando o conectivo da implicação ocorrer em uma fórmula, a parentetização implícita dá-se da direita para a esquerda, como, por exemplo, em '$P \rightarrow Q \rightarrow R \rightarrow S$', que representa '$P \rightarrow (Q \rightarrow (R \rightarrow S))$'.
- Quando os conectivos diádicos de mesmo nível hierárquico, distintos do conectivo da implicação, ocorrerem em uma fórmula, a parentetização implícita dá-se da esquerda para a direita, por exemplo:
  - '$P \leftrightarrow Q \leftrightarrow R \leftrightarrow S$' representa '$((P \leftrightarrow Q) \leftrightarrow R) \leftrightarrow S$',
  - '$P \wedge Q \wedge R \wedge S$' representa '$((P \wedge Q) \wedge R) \wedge S$',
  - '$P \vee Q \vee R \vee S$' representa '$((P \vee Q) \vee R) \vee S$',
  - '$P \wedge Q \vee R \wedge S_1 \vee S_2$' representa '$(((P \wedge Q) \vee R) \wedge S_1) \vee S_2$'.
- Consideramos todos os conectivos diádicos definidos distintos dos conectivos '$\leftrightarrow$' e '$\rightarrow$' como sendo do mesmo nível hierárquico dos conectivos '$\wedge$' e '$\vee$'.

Não é possível haver ambiguidade na leitura de uma samblagem significativa em uma lógica bem formulada, tais como todas as consideradas neste trabalho. Por exemplo, a frase "um velho bateu em uma velha com um pau" permite pelo menos duas interpretações distintas, dependendo de quem estava segurando o dito pau. A seguir é formulada esta não ambiguidade concernente a uma linguagem para LPC.

1.17. **Teorema da Legibilidade Única**
Cada fórmula em LPC só pode ser lida de uma única forma, isto é:

- Uma dada fórmula em LPC não possui mais de uma classificação entre as opções letra sentencial, negação, implicação, conjunção e disjunção.
- Se $\neg P$ e $\neg Q$ são negações idênticas, então $P = Q$,
- Se $(P \rightarrow Q)$ e $(R \rightarrow S)$ são implicações idênticas, então $P = R$ e $Q = S$,
- Se $(P \wedge Q)$ e $(R \wedge S)$ são conjunções idênticas, então $P = R$ e $Q = S$,
- Se $(P \vee Q)$ e $(R \vee S)$ são disjunções idênticas, então $P = R$ e $Q = S$.

§2. Uma Semântica para LPC

Esta seção define uma *semântica para* LPC.

2.1 Semântica para LPC

2.1 Definição. Os *valores veritativos em* LPC são v e f, onde o único *valor distinguido* é o v. Uma ordem é considerada na coleção de valores veritativos em LPC, em que f $<$ v.

2.2 Definição. (LPC**-interpretação**)
Uma LPC-interpretação é uma função I tal que, para toda letra sentencial $p \in \mathcal{D}(I)$, $I(p) \in \{\mathrm{v}, \mathrm{f}\}$.

2.3 Exemplo.
Considere I uma LPC-interpretação cujo domínio é $\{p_1, p_2, q\}$ definido por $I(p_1) = \mathrm{v}$, $I(p_2) = \mathrm{f}$ e $I(q) = \mathrm{v}$.

2.4 Exemplo.
Seja $I_1 : \{p_1, q_2\} \rightarrow \{\mathrm{v}, \mathrm{f}\}$, onde $p_1 \mapsto$ v e $q_2 \mapsto$ f.
Assim I_1 é uma LPC-interpretação onde $I_1(p_1) = \mathrm{v}$ e $I_1(q_2) = \mathrm{f}$. O domínio desse exemplo é $\mathcal{D}(I_1) = \{p_1, q_2\}$ e a imagem é $\mathcal{I}(I_1) = \{\mathrm{v}, \mathrm{f}\}$.

2.5 Exemplo.
Considere I_1 definido acima. Será que I_1 é interpretação para $(p_1 \wedge p_3)$?
I_1 não é LPC-interpretação para $p_1 \wedge p_3$ pois p_3 não está em I_1 no seu domínio $\mathcal{D}(I_1)$.

2.6 Exemplo.
Seja $I_2 : \{p_1, q_2, p_3\} \rightarrow \{\mathrm{v}, \mathrm{f}\}$, onde $p_1 \mapsto f$, $q_2 \mapsto v$ e $p_3 \mapsto v$.
Assim I_2 é uma LPC-interpretação onde $I_2(p_1) = \mathrm{f}$, $I_2(q_2) = \mathrm{v}$ e $I_2(p_3) = \mathrm{v}$.
O domínio desse exemplo é $\mathcal{D}(I_2) = \{p_1, q_2, p_3\}$ e a imagem é $\mathcal{I}(I_2) = \{\mathrm{v}, \mathrm{f}\}$.

2.7 Definição. (LPC**-interpretação para uma fórmula** P)
I é LPC-interpretação para P se toda letra sentencial em P pertence a $\mathcal{D}(I)$.

No exemplo 2.5, I_1 não é LPC-interpretação pois p_3 não pertence a $\mathcal{D}(I)$.

2.8 Definição. (LPC**-interpretação para uma coleção de fórmulas** Γ)
I é LPC-interpretação para Γ se, para todo $P \in \Gamma$, I é LPC-interpretação para P.

2.9 Lema. ***(Coincidência entre*** LPC***-interpretações)***
Sejam $\begin{cases} P \textit{ uma fórmula,} \\ I \textit{ e } I' \textit{ interpretações para } P. \end{cases}$
Se I e I' atribuem os mesmos valores para as letras sentenciais que estão na fórmula P, então essas interpretações atribuem o mesmo valor para a fórmula P. Dessa forma temos uma coincidência entre LPC*-interpretações.*

O lema da coincidência entre interpretações é comumente utilizado quando é necessário fazer provas que exigem uma mudança de linguagem. Um caso que necessita da mudança de linguagem, durante uma prova, é aquele em que na hipótese existem fórmulas que não estão na conclusão.

2.10 Definição. (**Extensão e Restrição de** LPC**-interpretações**)
Sejam I uma LPC-interpretação e I' uma extensão de I.
Diminuindo a quantidade de letras sentenciais que são interpretadas por I', obtemos I, que é chamada de *restrição de* I'. Também dizemos que I' é uma *extensão* de I.
Se I' é uma extensão de I, então I e I' interpretam as mesmas letras sentenciais. Verifica-se que $I(P) = I'(P)$ para toda fórmula fechada P de uma linguagem L de LPC.

2.11 Definição. (I**-fórmula**)
P é *I-fórmula em* LPC $\rightleftharpoons$ I é LPC-interpretação para P.

2.12 Definição. (I**-coleção de fórmulas**)
Γ é *I-coleção de fórmulas em* LPC $\rightleftharpoons$ I é LPC-interpretação para Γ.

2.13 Definição. (LPC**-valoração**)
Seja I uma LPC-interpretação. A partir de I especifica-se uma função I_V, da coleção de I-fórmulas para $\{\mathrm{v}, \mathrm{f}\}$, dita LPC-valoração, pelas seguintes cláusulas:

- se p é letra sentencial e $p \in \mathcal{D}(I)$, então $I_V(p) = I(p)$.
- $I_V(\neg P) \neq I_V(P)$.
- $I_V(P \to Q)=\begin{cases} I_V(Q), \text{ se } I_V(P) = \mathrm{v}, \\ \mathrm{v}, \text{ se } I_V(P) = \mathrm{f}. \end{cases}$
- $I_V(P \wedge Q)= \min\{I_V(P),\ I_V(Q)\}$.
- $I_V(P \vee Q)= \max\{I_V(P),\ I_V(Q)\}$.
- $I_V(\top)= \mathrm{v}$.
- $I_V(\bot)= \mathrm{f}$.

Ou seja, I_V é a função semântica que associa a coleção de I-fórmulas à coleção de valores veritativos em LPC.

2.14 Escólio.
Seja I uma LPC-*interpretação para P e Q. Então as seguintes proposições são válidas:*

(i) $I_V(\neg P) = \mathrm{v}$ *sss* não é o caso que $I_V(P) = \mathrm{v}$,
(ii) $I_V(P \to Q) = \mathrm{v}$ se $I_V(P) = \mathrm{v}$ implica que $I_V(Q) = \mathrm{v}$,
(iii) $I_V(P \wedge Q) = \mathrm{v}$ *se* $I_V(P) = \mathrm{v}$ e $I_V(Q) = \mathrm{v}$,
(iv) $I_V(P \vee Q) = \mathrm{v}$ *se* $I_V(P) = \mathrm{v}$ ou $I_V(Q) = \mathrm{v}$.

2.15 Teorema.
Se P, Q são I-fórmulas, então

- $I_V(P \leftrightarrow Q) = \begin{cases} \mathrm{v}, \textit{ se } I_V(P) = I_V(Q), \\ \mathrm{f}, \textit{ se } I_V(P) \neq I_V(Q). \end{cases}$
- $I_V(P \,\bar{\vee}\, Q) = \begin{cases} \mathrm{v}, \textit{ se } I_V(P) \neq I_V(Q), \\ \mathrm{f}, \textit{ se } I_V(P) = I_V(Q). \end{cases}$

- $I_V(P \downarrow Q) = \begin{cases} \text{v, } \textit{se } I_V(P) = I_V(Q) = \text{f}, \\ \text{f, } \textit{em caso contrário.} \end{cases}$
- $I_V(P \uparrow Q) = \begin{cases} \text{v, } \textit{se } I_V(P) = \text{f } \textit{ou } I_V(Q) = \text{f}, \\ \text{f, } \textit{se } I_V(P) = I_V(Q) = \text{v}. \end{cases}$

Prova:
$I_V(P \leftrightarrow Q) = \text{v}$ sss $I_V(P \to Q) = \text{v}$ e $I_V(Q \to P) = \text{v}$ sss $I_V(P) \leq I_V(Q)$ e $I_V(Q) \leq I_V(P)$ sss $I_V(P) = I_V(Q)$. □

2.16 Escólio.
Se I é uma LPC*-interpretação e, P e Q são I-fórmulas, temos que:*

(i) $I_V(P \leftrightarrow Q) = \text{v}$ equivale *a dizer que* $I_V(P) = \text{v}$ *sss* $I_V(Q) = \text{v}$,
(ii) $I_V(P \,\bar{\vee}\, Q) = \text{v}$ *sss* ou $I_V(P) = \text{v}$, ou $I_V(Q) = \text{v}$,
(iii) $I_V(P \downarrow Q) = \text{v}$ *sss* nem $I_V(P) = \text{v}$, nem $I_V(Q) = \text{v}$,
(iv) $I_V(P \uparrow Q) = \text{v}$ *sss* não é o caso que $I_V(P) = \text{v}$ e $I_V(Q) = \text{v}$.

2.17 Definição. (**Satisfabilidade de fórmula e de coleção de fórmulas em** LPC)

Sejam $\begin{cases} I \text{ uma LPC-interpretação,} \\ P \text{ uma } I\text{-fórmula,} \\ \Gamma \text{ uma } I\text{-coleção de fórmulas.} \end{cases}$

I satisfaz uma fórmula P em LPC, ou P é LPC-*satisfatível*, se $I_V(P)$ é um valor distinguido em LPC, ou seja, $I_V(P) = \text{v}$.
I satisfaz uma coleção Γ de fórmulas em LPC, ou Γ é LPC-*satistatível*, se I satisfaz cada fórmula de Γ em LPC.
Caso contrário, P é LPC-insatisfatível e Γ é LPC-insatisfatível.

2.18 Definição. (**Validade de fórmula e de coleção de fórmulas em** LPC)

(i) *I valida Γ em* LPC se cada fórmula de Γ é uma I-fórmula e se existe uma fórmula de Γ tal que I satisfaz essa fórmula.
(ii) P é LPC-*válido* se P é uma fórmula em LPC e toda LPC-interpretação para P satisfaz P, caso contrário, P é dito ser LPC-inválido.
(iii) Γ é LPC-*válido* se toda LPC-interpretação para Γ valida Γ, caso contrário Γ é dito ser LPC-inválido.

2.19 Definição. (LPC-**Consequência Semântica**)

P é consequência semântica de Γ em LPC, $\Gamma \models_{\text{LPC}} P$, se toda LPC-interpretação para $\Gamma \cup \{P\}$ que satisfaz Γ, também satisfaz P.

2.20 Leituras. $\Gamma \models_{\overline{\text{LPC}}} P$

- 'De Γ afirma-se P em LPC',
- 'Γ, portanto P em LPC',
- 'P é consequência semântica de Γ em LPC',
- 'P é teorema de Γ em LPC',
- 'Γ acarreta semanticamente P em LPC'.

2.21 Exemplo.
Como verificar se $p \models p \vee q$?
Seja I uma LPC-interpretação para $\{p, p \vee q\}$.
Se I satisfaz $\{p\}$, então $I_V(p) = \text{v}$, daí $I_V(p \vee q) = \text{v}$, portanto $p \models p \vee q$.

2.22 Exemplo.
Será que $p \models_{\overline{\text{LPC}}} p \wedge q$?
Seja $I : \{p, q\} \to \{\text{v}, \text{f}\}$.
Seja $p \mapsto \text{v}$ e $q \mapsto \text{f}$.
Temos que $I_V(p) = \text{v}$, $I_V(q) = \text{v}$ e $I_V(p \wedge q) = \text{f}$.
Assim, I_V satisfaz $\{p\}$, mas não satisfaz $\{p \wedge q\}$, portanto, $p \not\models_{\overline{\text{LPC}}} p \wedge q$.

2.23 Exemplo.
Será que $p \models_{\overline{\text{LPC}}} p \vee q$?
Seja I uma LPC-interpretação para $\{p, p \vee q\}$.
Se I satisfaz $\{p\}$, então $I_V(p) = \text{v}$, daí $I_V(p \vee q) = \text{v}$, portanto, $p \models_{\overline{\text{LPC}}} p \vee q$.

2.24 Definição. '$\Gamma \models_{\overline{\text{LPC}}} P$' é LPC-correto se $\Gamma \models_{\overline{\text{LPC}}} P$.

2.25 Escólio. '$\models_{\overline{\text{LPC}}} P$' *é* LPC-*correto sss P é* LPC-*válido.*

2.26 Definição. (**Propriedades de Fórmulas**)
indexpropriedades de fórmulas

- P é LPC-tautologia (é LPC-válida) $\rightleftharpoons$ toda LPC-interpretação para P satisfaz P.
- P é LPC-contingência $\rightleftharpoons$ há uma LPC-interpretação que satisfaz P e há uma LPC-interpretação que não satisfaz P.
- P é LPC-absurdo[6] $\rightleftharpoons$ toda LPC-interpretação para P não satisfaz P.

[6] Também pode ser chamado de inconsistente.

2.2 Semântica Clássica para Conectivos

2.27 Definição. (Tabela Veritativa)
A tabela-veritativa é um método de prova para LPC onde é possível determinar se uma dada fórmula é válida ou um dado sequente é correto. Para uma fórmula com n letras sentenciais, a tabela-verdade terá 2^n linhas.

As tabelas veritativas dos principais conectivos de LPC são apresentadas:

Tabela 4.1: *Tabela Veritativa da Negação*

P	**¬P**
v	f
f	v

Tabela 4.2: *Tabela Veritativa da Conjunção*

P	**Q**	**P ∧ Q**
v	v	v
v	f	f
f	v	f
f	f	f

Tabela 4.3: *Tabela Veritativa da Disjunção*

P	**Q**	**P ∨ Q**
v	v	v
v	f	v
f	v	v
f	f	f

Tabela 4.4: *Tabela Veritativa da Implicação*

P	**Q**	**P → Q**
v	v	v
v	f	f
f	v	v
f	f	v

Tabela 4.5: *Tabela Veritativa do Verum*

$\top$	v

Tabela 4.6: *Tabela Veritativa do Falsum*

$\bot$	f

§3. Um Cálculo de Sequentes para LPC

Esta seção apresenta um cálculo de sequentes para LPC.
Ele é definido pelas seguintes leis de LPC:

$$\text{(i) Leis Primitivas} \begin{cases} \text{Leis Estruturais,} \\ \text{Leis de Introdução e Eliminação de Conectivos.} \end{cases}$$

$$\text{(ii) Leis Derivadas} \begin{cases} \text{Leis Básicas dos Conectivos,} \\ \text{Leis Complementares dos Conectivos.} \end{cases}$$

As leis estruturais dizem respeito à própria inferência[7] em si, não dizem respeito aos conectivos, aos quantificadores ou aos outros sinais específicos de uma dada lógica. As mesmas são válidas em quase todos os sistemas lógicos usuais, exceto a lei da monotonicidade, que geralmente não é válida para as lógicas ditas *não monotônicas*.

Com o auxílio destas leis é possível demonstrar a validade de um grande número de argumentos complexos.

[7] Isto é, executar os "passos" de uma dedução ou demonstração.

Esta seção trata somente da Lógica Proposicional Clássica, assim, P é consequência de Γ em LPC é notado por $\Gamma \vdash P$[8].

Leis Estruturais

3.1. **Esquema da Reflexividade** Se $P \in \Gamma$, então $\Gamma \vdash P$.

3.2. **Regra da Transitividade**

Se $\begin{cases} \Gamma \vdash P_1, \\ \cdots \\ \Gamma \vdash P_n, \end{cases}$ e $P_1, \ldots, P_n \vdash Q$, então $\Gamma \vdash Q$.

3.3. **Regra da Monotonicidade**[9] Se $\Gamma \vdash P$ e $\Gamma \subseteq \Gamma'$, então $\Gamma' \vdash P$[10].

Leis de Introdução e Eliminação de Conectivos

3.4. **Regra da Dedução** Se $\Gamma, P \vdash Q$, então $\Gamma \vdash P \to Q$.[11]

3.5 Exemplo.

Verifique se a regra da dedução é correta semanticamente, $\dfrac{\Gamma, P \models Q}{\Gamma \models P \to Q}$.

Suponha que $\Gamma, P \models Q$. (1)
Seja I uma LPC-interpretação para $\Gamma \cup \{P \to Q\}$ tal que I satisfaz Γ.
Se $I_V(P) = \text{v}$, então I satisfaz $\Gamma \cup \{P\}$, daí por (1), $I_V(Q) = \text{v}$, logo $I_V(P \to Q) = \text{v}$.
Se $I_V(P) = \text{f}$ então $I_V(P \to Q) = \text{v}$.
Em qualquer caso, $I_V(P) = \text{v}$.
Ou seja, toda LPC-interpretação para $\Gamma \cup \{P \to Q\}$ que satisfaz Γ, também satisfaz $P \to Q$, portanto $\Gamma \models P \to Q$.

3.6. **Modus Ponens** $P, P \to Q \vdash Q$.

Permite deduzir a conclusão Q a partir das premissas P e $P \to Q$.

[8] Ou seja, fica implícito que $\Gamma \vdash_{\text{LPC}} P$.
[9] As lógicas ditas *não monotônicas* ou *indutivas* não aceitam esta regra como válida.
[10] A inclusão de novas premissas não altera conclusões já obtidas.
[11] As *lógicas relevantes* não aceitam esta regra sem o estabelecimento de certas conexões entre P e Q.

3.7 Exemplo.
Houve um grande terremoto. (P)
Se houve um grande terremoto, então houve muita destruição. ($P \to Q$)
Conclui-se que houve muita destruição. (Q)

3.8. $\wedge$**-Introdução** $P, Q \vdash P \wedge Q$

De duas fórmulas P, Q é possível deduzir a sua conjunção, $P \wedge Q$ ou $Q \wedge P$.

3.9. $\wedge$**-Eliminação** $\begin{cases} \text{(i)} & P \wedge Q \vdash P, \\ \text{(ii)} & P \wedge Q \vdash Q. \end{cases}$

Da conjunção entre duas fórmulas $P \wedge Q$, é possível deduzir cada uma das fórmulas, P ou Q.

3.10. $\vee$**-Introdução** $\begin{cases} \text{(i)} & P \vdash P \vee Q, \\ \text{(ii)} & Q \vdash P \vee Q. \end{cases}$

Dado uma fórmula P, dela se pode deduzir a sua disjunção com qualquer outra fórmula.

Damos a seguir, uma lei de introdução para o conectivo $\top$ e uma lei de eliminação para o conectivo $\bot$. Temos que o conectivo $\top$ não possui lei de eliminação, e o conectivo $\bot$ não possui lei de introdução.

3.11. $\top$**-Introdução**[12]: $\Gamma \vdash \top$

3.12. $\bot$**-Eliminação**[13]: $\bot \vdash P$

3.13 Convenção. Consideraremos, doravante, para todos os esquemas de sequentes de cálculos possuindo a regra da dedução, cujos exemplares só possuem uma premissa, as implicações correspondentes dos mesmos.

3.14 Exemplo. O esquema do $\wedge$-eliminação fornece, aplicando-se a Regra da Dedução, as implicações $\vdash P \wedge Q \to P$ e $\vdash P \wedge Q \to Q$. Segundo a convenção acima, podemos usar diretamente estas implicações em quaisquer provas, bastando citar o seu esquema correspondente, o qual é, neste caso, o $\wedge$-eliminação.

3.15 Escólio. *É dado a seguir uma lista dos principais esquemas em* LPC *na forma de implicação:*

(i) $\wedge$-*el:* $\vdash P \wedge Q \to P$.

(ii) $\wedge$-*el:* $\vdash P \wedge Q \to Q$.

(iii) $\vee$-*int:* $\vdash P \to P \vee Q$.

(iv) $\vee$-*int:* $\vdash Q \to P \vee Q$.

[12] Também poderia ser escrito assim $\varnothing \vdash \top$.
[13] Ou seja, $\bot$ acarreta qualquer coisa.

(v) *CI:* $\vdash Q \to (P \to Q)$.
(vi) *AI:* $\vdash \neg P \to (P \to Q)$.
(vii) *DN:* $\vdash \neg\neg \to P$.
(viii) *DN:* $\vdash P \to \neg\neg P$.
(ix) *Ctp:* $\vdash (P \to Q) \to (\neg Q \to \neg P)$.
(x) *Ctp:* $\vdash (\neg Q \to \neg P) \to (P \to Q)$.

3.16. **Esquema da Prova por Casos**[14] $P \vee Q, P \to R, Q \to R \vdash R$.

3.17. **Regras da Redução ao Absurdo** [15]

- $\neg$-Introdução: Se $\begin{cases} \Gamma, P \vdash Q, \\ \Gamma, P \vdash \neg Q, \end{cases}$ então $\Gamma \vdash \neg P$.
- $\neg$-Eliminação: Se $\begin{cases} \Gamma, \neg P \vdash Q, \\ \Gamma, \neg P \vdash \neg Q, \end{cases}$ então $\Gamma \vdash P$.[16]

Provas em Cálculos de Sequentes no estilo de Fitch

A partir desta seção faremos diversas deduções de leis corretas em alguma lógica.

O tipo mais elementar de dedução em um cálculo de sequentes é a demonstração, que só usa leis primitivas deste cálculo. Obviamente, todas as leis de um cálculo de sequentes podem ser deduzidas apenas por demonstrações, mas o uso exclusivo destas seria inviável, na prática, porque as mesmas ficariam muito grandes na dedução de quase todas as leis do cálculo. Por isto, na prática, usa-se leis derivadas nas deduções.

3.18 Definição. (Prova)
Seja $\mathcal{L}$ um cálculo de sequentes no qual, em um dado contexto, são conhecidas uma lista de leis derivadas do mesmo. Uma *prova em* $\mathcal{L}$ é uma lista de sequentes em $\mathcal{L}$ na qual cada sequente satisfaz uma das duas seguintes condições:
- Este sequente é exemplar de um esquema primitivo ou derivado em $\mathcal{L}$.
- Este sequente é consequência de sequentes anteriores nesta lista de uma aplicação de uma regra primitiva ou derivada em $\mathcal{L}$.

Na escrita de uma prova de um sequente de um cálculo de sequentes utiliza-se três colunas, onde a primeira coluna dá o número de cada sequente

[14] Corresponde a um $\vee$-eliminação.
[15] As lógicas *paraconsistentes* e/ou *paracompletas* não aceitam, em geral, tais regras.
[16] A *Lógica Intuicionista* não aceita a segunda regra da redução ao absurdo como válida.

desta prova, a segunda coluna contém os próprios sequentes, cuja lista constitui a prova em si, e a terceira coluna fornece as justificativas do porquê cada sequente figura na prova. Cada justificativa pode conter os nomes das leis primitivas ou derivadas utilizadas, bem como os números dos sequentes ou sequentes anteriores utilizados, se este for o caso.

Cada justificativa pode ser de uma das seguintes formas:

- uma abreviatura do nome do esquema primitivo ou derivado utilizado, caso o sequente correspondente seja exemplar de um esquema primitivo ou derivado do cálculo de sequentes.
- uma abreviatura do nome da regra primitiva ou derivada utilizada, caso o sequente correspondente seja a conclusão de uma aplicação desta regra a partir de sequentes anteriores na lista, e os números dos sequentes anteriores utilizados.
- 'ant', que é uma abreviatura de "conhecimento anterior", caso o sequente considerado seja um exemplar de uma lei derivada sem um nome específico.
- 'hip', que é uma abreviatura de hipótese, que contém alguma especificação adicional do sequente a ser provado e que não está explícita neste sequente.
- Em alguns casos, para dar um ritmo mais rápido à prova, pode-se utilizar uma combinação dos tipos de justificativas anteriores.

A seguir, apresentamos exemplos de provas completas de alguns dois sequentes.

3.19 Exemplo.
Seja TNI um sistema lógico pelo qual são deduzidas as propriedades básicas dos números inteiros. Então, prove que $\vdash_{\mathrm{TNI}}$ ímpar$(x) \rightarrow$ ímpar(x^2).

1	ímpar$(x) \vdash_{\mathrm{TNI}}$ ímpar(x)	Ref
2	ímpar$(x) \vdash_{\mathrm{TNI}} x = 2n + 1$, para algum n	1, def
3	$x = 2n + 1 \vdash_{\mathrm{TNI}} x^2 = (2n + 1)^2$	ant
4	ímpar$(x) \vdash_{\mathrm{TNI}} x^2 = (2n + 1)^2$	2, 3, Tran
5	$x^2 = (2n + 1)^2 \vdash_{\mathrm{TNI}} x^2 = 4n^2 + 4n + 1$	ant
6	ímpar$(x) \vdash_{\mathrm{TNI}} x^2 = 4n^2 + 4n + 1$	4, 5, Tran
7	$x^2 = 4n^2 + 4n + 1 \vdash_{\mathrm{TNI}} x^2 = 2(2n^2 + 2n) + 1$	ant
8	ímpar$(x) \vdash_{\mathrm{TNI}} x^2 = 2(2n^2 + 2n) + 1$	6, 7, Tran
9	ímpar$(x) \vdash_{\mathrm{TNI}}$ ímpar(x^2)	8, def
10	$\vdash_{\mathrm{TNI}}$ ímpar$(x) \rightarrow$ ímpar(x^2)	9, RD

3.20 Exemplo.
Seja $\mathcal{L}_0$ uma lógica com estas duas leis primitivas, onde a primeira é o esquema da reflexividade e a segunda é chamada de regra da generalização, as quais são comuns a quase todos os cálculos de sequentes dados neste trabalho, exceto LPC:

- Ref: Se $P \in \Gamma$, então $\Gamma \vdash_{\mathcal{L}_0} P$ é um exemplar de Ref.
- Gen: Se x não é livre em Γ, então $\dfrac{`\Gamma \vdash_{\mathcal{L}_0} P'}{\Gamma \vdash_{\mathcal{L}_0} \forall x}$' é uma aplicação de Gen.

Prove que, se x não é livre em P, então $P \vdash_{\mathcal{L}_0} \forall xP$.

1	x não é livre em P	hip
2	$P \vdash_{\mathcal{L}_0} P$	Ref
3	$P \vdash_{\mathcal{L}_0} \forall xP$	1, 2, Gen

A escrita de provas de sequentes feita conforme os exemplos 3.19 e 3.20 é extensa, pois diversas premissas são repetidas em vários passos, daí usaremos um estilo compacto de escrita de provas de sequentes em cálculos de sequentes, devido a Fitch[17].

Este estilo apresenta as seguintes características, considerando uma prova de um sequente '$\Gamma \vdash P$':

- As premissas de cada sequente considerado não são escritas explicitamente, mas podem, em cada passo, ser determinadas.
- Cada fórmula de Γ, ao ser considerada, é justificada como sendo uma *premissa global*, pois a mesma é considerada em toda a prova, no sentido em que ela é premissa em todos os sequentes da prova. Neste caso, tal justificativa é dada por 'pr'.
- Cada fórmula adicional adotada como premissa em uma parte da prova, mas não em toda a prova, é uma *premissa local* ou *suposição* da mesma. Neste caso, a sua justificativa é dada por 'sup'.
- Quando uma ou mais premissas locais são adicionadas à coleção existente de premissas, então, para delimitar a sua duração na prova, usa-se uma linha vertical no lado esquerdo de cada fórmula.
- Aplicações das regras estruturais (Mon e Tran) dos cálculos de sequentes dados neste trabalho não são citadas explicitamente, a fim de evitar uma escrita demasiadamente detalhista das justificativas das provas.

Usamos, especialmente em aulas orais dos conteúdos apresentados nesta dissertação, os verbos "supor" ou "carregar" no sentido de acrescen-

[17]Método criado por Frederic Brenton Fitch [42]. Na língua inglesa, este estilo é conhecido por *Fitch-style proof*.

tar uma ou mais premissas à coleção de premissas previamente existente na prova. O ato inverso é o de retirar uma ou mais premissas da coleção vigente de premissas; neste caso, usamos o verbo "descarregar" para expressar este ato.

A prova compacta no estilo de Fitch do exemplo 3.19 é dada a seguir.

1	$\vert$ ímpar(x)	sup
2	$\vert$ $x = 2n + 1,$ para algum n	1, def
3	$\vert$ $x^2 = (2n+1)^2$	2, ant
4	$\vert$ $x^2 = 4n^2 + 4n + 1$	3, ant
5	$\vert$ $x^2 = 2(2n^2 + 2n) + 1$	4, ant
6	$\vert$ ímpar(x^2)	5, def
7	ímpar$(x) \to$ ímpar(x^2)	1, 6, RD

A prova compacta no estilo de Fitch do exemplo 3.20 é dada a seguir.

1	x não é livre em Gen	hip
2	P	pr
3	$\forall x P$	2, 1, Gen

A seguir, relacionamos a prova do sequente $\neg Q, P \to Q \vdash \neg P$ no estilo de Fitch e no modo completo.

1	$\neg Q, P \to Q \vdash \neg Q$	Ref
2	$\neg Q, P \to Q \vdash P \to Q$	Ref
3	$\neg Q, P \to Q, P \vdash P$	Ref
4	$\neg Q, P \to Q, P \vdash Q$	3, 2, MP
5	$\neg Q, P \to Q, P \vdash \neg Q$	Ref
6	$\neg Q, P \to Q \vdash \neg P$	3, 4, 5, $\neg$-int

1	$\neg Q$	pr
2	$P \to Q$	pr
3	$\quad$ | P	sup
4	$\quad$ | Q	3, 2, MP
5	$\quad$ | $\neg Q$	1
6	$\neg P$	3, 4, 5, $\neg$-int

Uma apresentação complementar sobre a prova compacta no estilo de Fitch também pode ser encontrada em [7].

Leis Básicas dos Conectivos

3.21. **Reflexividade da Implicação** $\vdash P \to P$.

Prova:

1	P	Ref
2	$\vdash P \to P$	1, RD

□

3.22. **Não Contradição**[18] $\begin{cases} \text{(i)} & P, \neg P \vdash Q, \\ \text{(ii)} & \vdash \neg(P \wedge \neg P). \end{cases}$

Prova de (i):

1	P	pr
2	$\neg P$	pr
3	$\quad$ | $\neg Q$	sup
4	$\quad$ | P	1
5	$\quad$ | $\neg P$	2
6	Q	3, 4, 5, $\neg$-el

□

Prova de (ii):

1	| $P \wedge \neg P$	sup
2	| P	1, $\wedge$-el
3	| $\neg P$	1, $\wedge$-el
4	$\neg(P \wedge \neg P)$	1,2,3,$\neg$-int

□

[18] A Lei da Não-Contradição também pode ser chamada de *Princípio de Duns Scot*, de onde afirma-se que uma contradição implica qualquer proposição.

3.23. **Terceiro Excluído** $\vdash P \vee \neg P$.

Prova:

$$
\begin{array}{lll}
1 & \quad \mid \neg(P \vee \neg P) & \text{sup} \\
2 & \quad \mid \quad \mid P & \text{sup} \\
3 & \quad \mid \quad \mid P \vee \neg P & 2, \vee\text{-int} \\
4 & \quad \mid \quad \mid \neg(P \vee \neg P) & 1, \text{Mon} \\
5 & \quad \mid \neg P & 2, 3, 4, \neg\text{-int} \\
6 & \quad \mid P \vee \neg P & 5, \vee\text{-int} \\
7 & \quad P \vee \neg P & 1, 6, \neg\text{-el}
\end{array}
$$

□

3.24. **Consequente da Implicação** $Q \vdash P \to Q$.

Prova:

$$
\begin{array}{lll}
1 & \quad Q & \text{pr} \\
2 & \quad \mid P & \text{sup} \\
3 & \quad \mid Q & 1, \text{Mon} \\
4 & \quad P \to Q & 2, 3, \text{RD}
\end{array}
$$

□

3.25. **Antecedente da Implicação** $\neg P \vdash P \to Q$.

Prova:

$$
\begin{array}{lll}
1 & \quad \neg P & \text{pr} \\
2 & \quad \mid P & \text{sup} \\
3 & \quad \mid \neg P & 1, \text{Mon} \\
4 & \quad \mid Q & 2, 3, \text{NC} \\
5 & \quad P \to Q & 2, 4, \text{RD}
\end{array}
$$

□

3.26. **Silogismo Hipotético** $P \to Q, Q \to R \vdash P \to R$.

Prova:

1	$P \to Q$	pr
2	$Q \to R$	pr
3	$\quad\mid P$	sup
4	$\quad\mid Q$	3, 1, MP
5	$\quad\mid R$	4, 2, MP
6	$P \to R$	3, 5, RD

□

3.27. **Regra Recíproca da Dedução** Se $\Gamma \vdash P \to Q$, então $\Gamma, P \vdash Q$.

Prova:

1	$\Gamma \vdash P \to Q$	hip
2	$\Gamma, P \vdash P$	Ref
3	$\Gamma, P \vdash P \to Q$	1, Mon
4	$P, P \to Q \vdash P$	2, 3, MP
5	$\Gamma, P \vdash Q$	2, 3, 4, Tran

□

3.28. **Dupla Negação** $\begin{cases} \text{(i)} & \neg\neg P \vdash P, \\ \text{(ii)} & P \vdash \neg\neg P. \end{cases}$

Prova de (i):

1	$\neg\neg P$	pr
2	$\quad\mid \neg P$	sup
3	$\quad\mid \neg\neg P$	1, Mon
4	P	2, 3, $\neg$-el

□

Prova de (ii):

1	P	pr
2	$\quad\mid \neg P$	sup
3	$\quad\mid P$	1, Mon
4	$\neg\neg P$	2, 3, $\neg$-int

□

3.29. **Modus Tollens** $\neg Q, P \to Q \vdash \neg P$.

Permite deduzir a negação do antecedente a partir das premissas que possuam uma implicação e a negação do consequente.

Prova:

1	$\neg Q$	pr
2	$P \to Q$	pr
3	$\quad P$	sup
4	$\quad Q$	3, 2, MP
5	$\quad \neg Q$	1, Mon
6	$\neg P$	3, 4, 5, $\neg$-int

$\square$

3.30. **Contraposição** $\begin{cases} \text{(i)} & P \to Q \vdash \neg Q \to \neg P, \\ \text{(ii)} & \neg Q \to \neg P \vdash P \to Q. \end{cases}$

3.31. **Silogismo Disjuntivo** $\begin{cases} \text{(i)} & \neg P, P \vee Q \vdash Q, \\ \text{(ii)} & \neg Q, P \vee Q \vdash P. \end{cases}$

3.32. **Negação da Implicação** $\begin{cases} \text{(i)} & \neg(P \to Q) \vdash P \wedge \neg Q, \\ \text{(ii)} & P \wedge \neg Q \vdash \neg(P \to Q). \end{cases}$

3.33. **Negação da Conjunção** $\begin{cases} \text{(i)} & \neg(P \wedge Q) \vdash P \to \neg Q, \\ \text{(ii)} & P \to \neg Q \vdash \neg(P \wedge Q), \\ \text{(iii)} & \neg(P \wedge Q) \vdash Q \to \neg P, \\ \text{(iv)} & Q \to \neg P \vdash \neg(P \wedge Q). \end{cases}$

3.34. **De Morgan** $\begin{cases} \text{(i)} & \neg(P \vee Q) \vdash \neg P \wedge \neg Q, \\ \text{(ii)} & \neg P \wedge \neg Q \vdash \neg(P \vee Q), \\ \text{(iii)} & \neg(P \wedge Q) \vdash \neg P \vee \neg Q, \\ \text{(iv)} & \neg P \vee \neg Q \vdash \neg(P \wedge Q). \end{cases}$

3.35. **Implicação Material** $\begin{cases} \text{(i)} & P \to Q \vdash \neg P \vee Q, \\ \text{(ii)} & \neg P \vee Q \vdash P \to Q. \end{cases}$

3.36. **Redução da Disjunção**[19] $\begin{cases} \text{(i)} \;\; P \vee Q \vdash \neg P \to Q, \\ \text{(ii)} \;\; \neg P \to Q \vdash P \vee Q, \\ \text{(iii)} \;\; P \vee Q \vdash \neg Q \to P, \\ \text{(iv)} \;\; \neg Q \to P \vdash P \vee Q. \end{cases}$

3.37. **$\leftrightarrow$-Introdução** $P \to Q, Q \to P \vdash P \leftrightarrow Q$.

3.38. **$\leftrightarrow$-Eliminação** $\begin{cases} \text{(i)} \;\; P \leftrightarrow Q \vdash P \to Q, \\ \text{(ii)} \;\; P \leftrightarrow Q \vdash Q \to P. \end{cases}$

3.39. **Modus Ponens da Equivalência** $\begin{cases} \text{(i)} \;\; P, P \leftrightarrow Q \vdash Q, \\ \text{(ii)} \;\; Q, P \leftrightarrow Q \vdash P. \end{cases}$

3.40. **Reflexividade da Equivalência** $\vdash P \leftrightarrow P$.

3.41. **Equivalência Absurda** $\vdash \neg(P \leftrightarrow \neg P)$.

3.42. **Comutatividade da Equivalência** $\vdash (P \leftrightarrow Q) \leftrightarrow (Q \leftrightarrow P)$.

3.43. **Transitividade da Equivalência** $P \leftrightarrow Q, Q \leftrightarrow R \vdash P \leftrightarrow R$.

3.44 Escólio. *As leis da dupla negação, contraposição, negação da implicação, negação da conjunção, De Morgan, implicação material e redução da disjunção podem ser reescritas como equivalências. Veja a seguir:*

(i) Dupla Negação (iii): $\vdash \neg\neg P \leftrightarrow P$.
(ii) Contraposição (iii): $\vdash (P \to Q) \leftrightarrow (\neg Q \to \neg P)$.
(iii) Negação da Implicação (iii): $\vdash \neg(P \to Q) \leftrightarrow (\neg Q \to \neg P)$.
(iv) Negação da Conjunção (v): $\vdash \neg(P \wedge Q) \leftrightarrow (P \to \neg Q)$.
(v) Negação da Conjunção (vi): $\vdash \neg(P \wedge Q) \leftrightarrow (Q \to \neg P)$.
(vi) De Morgan (v): $\vdash \neg(P \vee Q) \leftrightarrow \neg P \wedge \neg Q$.
(vii) De Morgan (vi): $\vdash \neg(P \wedge Q) \leftrightarrow \neg P \vee \neg Q$.
(viii) Implicação Material (iii): $\vdash (P \to Q) \leftrightarrow (\neg P \vee Q)$.
(ix) Redução da Disjunção (v): $\vdash P \vee Q \leftrightarrow \neg P \to Q$.
(x) Redução da Disjunção (vi): $\vdash P \vee Q \leftrightarrow \neg Q \to P$.

3.45. **Lema da Substituição para Conectivos da Equivalência**
(i) $P_1 \leftrightarrow P_2 \vdash \neg P_1 \leftrightarrow \neg P_2$,
(ii) $P_1 \leftrightarrow P_2 \vdash (P_1 \to Q) \leftrightarrow (P_2 \to Q)$,
(iii) $P_1 \leftrightarrow P_2 \vdash (Q \to P_1) \leftrightarrow (Q \to P_2)$,

[19] Uma disjunção é equivalente a negação de um de seus disjuntores, implicando outro disjuntor.

(iv) $P_1 \leftrightarrow P_2 \vdash (P_1 \wedge Q) \leftrightarrow (P_2 \wedge Q)$,
(v) $P_1 \leftrightarrow P_2 \vdash (Q \wedge P_1) \leftrightarrow (Q \wedge P_2)$,
(vi) $P_1 \leftrightarrow P_2 \vdash (P_1 \vee Q) \leftrightarrow (P_2 \vee Q)$,
(vii) $P_1 \leftrightarrow P_2 \vdash (Q \vee P_1) \leftrightarrow (Q \vee P_2)$,
(viii) $P_1 \leftrightarrow P_2, Q_1 \leftrightarrow Q_2 \vdash (P_1 \rightarrow Q_1) \leftrightarrow (P_2 \rightarrow Q_2)$,
(ix) $P_1 \leftrightarrow P_2, Q_1 \leftrightarrow Q_2 \vdash (P_1 \wedge Q_1) \leftrightarrow (P_2 \wedge Q_2)$,
(x) $P_1 \leftrightarrow P_2, Q_1 \leftrightarrow Q_2 \vdash (P_1 \vee Q_1) \leftrightarrow (P_2 \vee Q_2)$.

Leis Complementares dos Conectivos

3.46. **Idempotência** $\begin{cases} \text{(i)} \vdash P \wedge P \leftrightarrow P, \\ \text{(ii)} \vdash P \vee P \leftrightarrow P. \end{cases}$

3.47 Exemplo.
- $1 \times 1 = 1$ (é idempotente).
- $3 \times 1 \neq 1$ (não é idempotente).

3.48. **Membros da Equivalência** $\begin{cases} \text{(i)} \ P, Q \vdash P \leftrightarrow Q, \\ \text{(ii)} \ \neg P, \neg Q \vdash P \leftrightarrow Q, \\ \text{(iii)} \ P, \neg Q \vdash \neg(P \leftrightarrow Q), \\ \text{(iv)} \ \neg P, Q \vdash \neg(P \leftrightarrow Q). \end{cases}$

3.49. **Equivalência Material** $\begin{cases} \text{(i)} \ (P \leftrightarrow Q) \leftrightarrow (P \wedge Q) \vee (\neg P \wedge \neg Q), \\ \text{(ii)} \ \neg(P \leftrightarrow Q) \leftrightarrow (P \wedge \neg Q) \vee (\neg P \wedge Q), \\ \text{(iii)} \ (P \leftrightarrow Q) \leftrightarrow (P \vee \neg Q) \wedge (\neg P \vee Q), \\ \text{(iv)} \ \neg(P \leftrightarrow Q) \leftrightarrow (P \vee Q) \wedge (\neg P \vee \neg Q). \end{cases}$

3.50. **Negação da Equivalência** $\begin{cases} \text{(i)} \vdash \neg(P \leftrightarrow Q) \leftrightarrow (\neg P \leftrightarrow Q), \\ \text{(ii)} \vdash \neg(P \leftrightarrow Q) \leftrightarrow (P \leftrightarrow \neg Q). \end{cases}$

A lei de substituição para equivalência em LPC será enunciada aqui, mas detalhada e provada em §9.2.

3.51. **Substituição para Equivalência**
- $P_1 \leftrightarrow P_2 \vdash Q(S \| P_1) \leftrightarrow Q(S \| P_2)$.

3.52. **Comutatividade** $\begin{cases} \text{(i)} \vdash P \wedge Q \leftrightarrow Q \wedge P, \\ \text{(ii)} \vdash P \vee Q \leftrightarrow Q \vee P, \\ \text{(iii)} \vdash (P \leftrightarrow Q) \leftrightarrow (Q \leftrightarrow P). \end{cases}$

3.53. **Associatividade** $\begin{cases} \text{(i)} \vdash P \wedge (Q \wedge R) \leftrightarrow (P \wedge Q) \wedge R, \\ \text{(ii)} \vdash P \vee (Q \vee R) \leftrightarrow (P \vee Q) \vee R, \\ \text{(iii)} \vdash (P \leftrightarrow (Q \leftrightarrow R)) \leftrightarrow ((P \leftrightarrow Q) \leftrightarrow R). \end{cases}$

3.54. **Absorção** $\begin{cases} \text{(i)} & \vdash P \wedge (P \vee Q) \leftrightarrow P, \\ \text{(ii)} & \vdash P \vee (P \wedge Q) \leftrightarrow P, \\ \text{(iii)} & \vdash P \wedge (\neg P \vee Q) \leftrightarrow P \wedge Q, \\ \text{(iv)} & \vdash P \vee (\neg P \wedge Q) \leftrightarrow P \vee Q. \end{cases}$

3.55. **Dilema Construtivo** $P \vee Q, P \rightarrow R, Q \rightarrow S \vdash R \vee S$.

3.56. **Dilema Destrutivo** $\neg R \vee \neg S$, $P \rightarrow R$, $Q \rightarrow S \vdash \neg P \vee \neg Q$.

3.57. **Divisão e Conquista**
$R \rightarrow P \vee Q$, $R \wedge P \leftrightarrow P'$, $R \wedge Q \leftrightarrow Q'$, $P' \vee Q' \leftrightarrow S \vdash R \leftrightarrow S$.

3.58. **Distributividade** $\begin{cases} \text{(i)} & \vdash P \wedge (Q \vee R) \leftrightarrow (P \wedge Q) \vee (P \wedge R), \\ \text{(ii)} & \vdash P \vee (Q \wedge R) \leftrightarrow (P \vee Q) \wedge (P \vee R). \end{cases}$

3.59. **Importação e Exportação** $\vdash P \rightarrow (Q \rightarrow R) \leftrightarrow (P \wedge Q) \rightarrow R$.[20]

3.60 Exemplo.

Se houver um ataque terrorista (P), então, se houver muitas vítimas (Q), então haverá guerra (R).

$$\downarrow$$
$$P \rightarrow (Q \rightarrow R)$$
$$\updownarrow$$
$$P \wedge Q \rightarrow R^{21}$$
$$\downarrow$$

Se houver um ataque terrorista (P) e houver muitas vítimas (Q), então haverá guerra (R).

A segunda proposição do esquema abaixo diz que toda fórmula verdadeira em um dado contexto equivale, neste contexto, a $\top$.

3.61. **Veracidade** $\begin{cases} \text{(i)} & \vdash \top, \\ \text{(ii)} & P \vdash P \leftrightarrow \top. \end{cases}$

Analogamente, a segunda proposição do esquema abaixo diz que toda fórmula falsa em um dado contexto equivale, neste contexto, a $\bot$.

3.62. **Falsidade** $\begin{cases} \text{(i)} & \vdash \neg\bot, \\ \text{(ii)} & \neg P \vdash P \leftrightarrow \bot. \end{cases}$

[20] O esquema '$\vdash (P \rightarrow (Q \rightarrow R)) \rightarrow ((P \wedge Q) \rightarrow R)$' é chamado de *lei da exportação*. Sequentes que possuem o conectivo da equivalência e não possuem premissas podem ter as fórmulas trocadas facilmente.

[21] Aplicando a lei de importação.

3.63 Definição.
Uma fórmula em LPC é dita veritativa se ela pode ser construída usando somente as fórmulas $\top$ e $\bot$, isto é, conforme as seguintes cláusulas:
- $\top$ e $\bot$ são veritativas.
- Se P é veritativa, então é $\neg P$ veritativa.
- Se P e Q são veritativas então é $P \# Q$ é veritativa.

3.64. **Lei da Formação Veritativa**
Se P é veritativa, então $\vdash P \leftrightarrow \top$ ou $P \leftrightarrow \bot$.

3.65 Escólio. *A escolha da letra sentencial usada para definir os conectivos '$\top$' e '$\bot$' é irrelevante, no que diz respeito à validade das leis em* LPC *concernentes a estes conectivos, em virtude do seguinte fato:*

3.66 Fato. *Se $\top'$ e $\bot'$ são respectivamente abreviaturas de '$p_1 \rightarrow p_1$' e de '$\neg(p_1 \rightarrow p_1)$', onde p_1 é uma letra sentencial qualquer, então as seguintes proposições são válidas:*
- $\vdash \top \leftrightarrow \top'$,
- $\vdash \bot \leftrightarrow \bot'$.

Estratégias de Prova

Apresentamos algumas sistematizações de estratégias clássicas de provas comumente utilizadas.

Como provar que $P \rightarrow Q$

- Prova Direta (Utilizando a regra da dedução):
  Inicia por assumir que P (a hipótese) é verdadeira. Então, tenta-se mostrar que Q (a conclusão) também é verdadeira. Conclui-se a prova aplicando a Regra da Dedução entre a hipótese (antecedente da implicação) e a conclusão (consequente da implicação).

1	$\quad P$	
2	$\quad \vdots$	
3	$\quad Q$	
4	$P \rightarrow Q$	1, 3, RD

- Prova Indireta (Utilizando a regra da dedução e a lei da contraposição):
  Pela lei da contraposição (i), sabe-se que $P \to Q$ é logicamente equivalente a $\neg Q \to \neg P$. Dessa forma, nesta segunda estratégia de prova para $P \to Q$, inicialmente assume-se que $\neg Q$ é verdadeiro (ou seja, que Q é falso) e, então, tenta-se mostrar que $\neg P$ é verdadeiro (ou seja, que P é falso).

$$\begin{array}{r|ll}
1 & \neg Q & \\
2 & \vdots & \\
3 & \neg P & \\
\multicolumn{1}{r}{4} & \neg Q \to \neg P & 1, 3, \text{RD} \\
\multicolumn{1}{r}{5} & P \to Q & 4, \text{Ctp}
\end{array}$$

- Utilizando a lei do consequente da implicação:
  Caso o Q já exista no ambiente de prova, pode-se dizer, pela lei do consequente da implicação, que $P \to Q$.

$$\begin{array}{r|ll}
1 & \vdots & \\
2 & Q & \\
3 & \vdots & \\
\multicolumn{1}{r}{4} & P \to Q & 2, \text{CI}
\end{array}$$

- Utilizando a lei do antecedente da implicação:
  Caso a negação de P já exista no ambiente de prova, $\neg P$, pode-se dizer, pela lei do antecedente da implicação, que $P \to Q$.

$$\begin{array}{r|ll}
1 & \vdots & \\
2 & \neg P & \\
3 & \vdots & \\
\multicolumn{1}{r}{4} & P \to Q & 2, \text{AI}
\end{array}$$

Como provar $P \wedge Q$

- Prove cada um dos conjuntores, separadamente, em qualquer ordem, e então faça a conjunção deles com a introdução do $\wedge$ ($\wedge$-int).

$$\begin{array}{lll}
1 & \vert\ P & \\
2 & \vert\ \vdots & \\
3 & \vert\ Q & \\
4 & P \wedge Q & 1, 3, \wedge\text{-int}
\end{array}
\qquad\qquad
\begin{array}{lll}
1 & \vert\ Q & \\
2 & \vert\ \vdots & \\
3 & \vert\ P & \\
4 & Q \wedge P & 1, 3, \wedge\text{-int}
\end{array}$$

Como provar $P \vee Q$

- Se o primeiro disjuntor estiver disponível no ambiente de prova, pela introdução do $\vee$ (ou $\vee$-int) têm-se a prova da disjunção.

$$\begin{array}{lll}
1 & \vert\ P & \\
2 & \vert\ \vdots & \\
3 & P \vee Q & 1, \vee\text{-int}
\end{array}$$

- Se o segundo disjuntor estiver disponível no ambiente de prova, pela introdução do $\vee$ (ou $\vee$-int) têm-se a prova da disjunção.

$$\begin{array}{lll}
1 & \vert\ Q & \\
2 & \vert\ \vdots & \\
3 & P \vee Q & 1, \vee\text{-int}
\end{array}$$

- Supor a negação do primeiro disjuntor para tentar provar o segundo, ou vice-versa. (Utilizando a regra da dedução e a lei da redução da disjunção):

$$
\begin{array}{lll}
1 & \quad\vline\; \neg P & \\
2 & \quad\vline\; \vdots & \\
3 & \quad\vline\; Q & \\
4 & \neg P \to Q & \text{1, 3, RD} \\
5 & P \vee Q & \text{4, RDj}
\end{array}
\qquad
\begin{array}{lll}
1 & \quad\vline\; \neg Q & \\
2 & \quad\vline\; \vdots & \\
3 & \quad\vline\; P & \\
4 & \neg Q \to P & \text{1, 3, RD} \\
5 & P \vee Q & \text{4, RDj}
\end{array}
$$

- Supor a negação de toda a disjunção para chegar a uma contradição. É uma Prova não-construtiva (prova por absurdo).

$$
\begin{array}{lll}
1 & \quad\vline\; \neg(P \vee Q) & \\
2 & \quad\vline\; \vdots & \\
3 & \quad\vline\; R & \\
4 & \quad\vline\; \vdots & \\
5 & \quad\vline\; \neg R & \\
6 & P \vee Q & \text{1, 3, 5, } \neg\text{-el}
\end{array}
\qquad
\begin{array}{lll}
1 & \quad\vline\; \neg(P \vee Q) & \\
2 & \quad\vline\; \vdots & \\
3 & \quad\vline\; \neg R & \\
4 & \quad\vline\; \vdots & \\
5 & \quad\vline\; R & \\
6 & P \vee Q & \text{1, 3, 5, } \neg\text{-el}
\end{array}
$$

Como provar $\neg P$

- Para provar $\neg P$, assumir que P é verdade e tentar chegar a uma contradição.

$$
\begin{array}{lll}
1 & \quad\vline\; P & \\
2 & \quad\vline\; \vdots & \\
3 & \quad\vline\; Q & \\
4 & \quad\vline\; \vdots & \\
5 & \quad\vline\; \neg Q & \\
6 & \neg P & \text{1, 3, 5 } \neg\text{-int}
\end{array}
\qquad
\begin{array}{lll}
1 & \quad\vline\; P & \\
2 & \quad\vline\; \vdots & \\
3 & \quad\vline\; \neg Q & \\
4 & \quad\vline\; \vdots & \\
5 & \quad\vline\; Q & \\
6 & \neg P & \text{1, 3, 5, } \neg\text{-int}
\end{array}
$$

- Provar $\neg P$ através do $\neg$-int, quando $P = Q$.

$$
\begin{array}{l|ll}
1 & P & \\
2 & \vdots & \\
3 & \neg P & \\
\multicolumn{1}{l}{4} & \neg P & \quad 1, 3, \neg\text{-int}
\end{array}
$$

Como provar que $P \leftrightarrow Q$

- Utilizando a regra do $\wedge$-introdução.

$$
\begin{array}{l|ll}
1 & P \to Q & \\
2 & \vdots & \\
3 & Q \to P & \\
4 & \vdots & \\
\multicolumn{1}{l}{5} & P \leftrightarrow Q & \quad 1, 3, \wedge\text{-int}
\end{array}
\qquad
\begin{array}{l|ll}
1 & Q \to P & \\
2 & \vdots & \\
3 & P \to Q & \\
4 & \vdots & \\
\multicolumn{1}{l}{5} & P \leftrightarrow Q & \quad 1, 3, \wedge\text{-int}
\end{array}
$$

- Utilizando a lei dos membros da equivalência (i).

$$
\begin{array}{l|ll}
1 & P & \\
2 & \vdots & \\
3 & Q & \\
4 & \vdots & \\
\multicolumn{1}{l}{5} & P \leftrightarrow Q & \quad 1, 3, \text{ME}
\end{array}
$$

- Utilizando a lei dos membros da equivalência (ii).

1	$\quad \neg P$	
2	$\quad \vdots$	
3	$\quad \neg Q$	
4	$\quad \vdots$	
5	$P \leftrightarrow Q$	1, 3, ME

A **Prova por absurdo**: É um recurso bastante utilizado para provar qualquer proposição. Para provar P, assuma que $\neg P$ é verdadeiro e encontre uma contradição. O terceiro caso é utilizado quando $P = Q$.

1	$\quad \neg P$	
2	$\quad \vdots$	
3	$\quad Q$	
4	$\quad \vdots$	
5	$\quad \neg Q$	
6	P	1, 3, 5, $\neg$-el

1	$\quad \neg P$	
2	$\quad \vdots$	
3	$\quad \neg Q$	
4	$\quad \vdots$	
5	$\quad Q$	
6	P	1, 3, 5, $\neg$-el

1	$\quad \neg P$	
2	$\quad \vdots$	
3	$\quad P$	
4	P	1, 3, $\neg$-el

Como usar

Demonstramos estratégias de como usar as sentenças abaixo durante uma prova:

- $P \rightarrow Q$,
- $P \wedge Q$,
- $P \vee Q$,
- $P \leftrightarrow Q$,
- $\neg P$,
- $\neg(P \vee Q)$,
- $\neg\neg P$,
- $\neg(P \leftrightarrow Q)P$.

Como usar $P \to Q$

$$
\begin{array}{ll|ll}
1 & & P & \\
2 & & \vdots & \\
3 & & P \to Q & \\
4 & Q & & 1, 3, \text{MP}
\end{array}
\qquad\qquad
\begin{array}{ll|ll}
1 & & \neg Q & \\
2 & & \vdots & \\
3 & & P \to Q & \\
4 & \neg P & & 1, 3, \text{MT}
\end{array}
$$

Como usar $P \wedge Q$

$$
\begin{array}{ll|ll}
1 & & P \wedge Q & \\
2 & & \vdots & \\
3 & & P & \\
4 & P & & 1, 3, \wedge\text{-el}
\end{array}
\qquad\qquad
\begin{array}{ll|ll}
1 & & P \wedge Q & \\
2 & & \vdots & \\
3 & & Q & \\
4 & Q & & 1, 3, \wedge\text{-el}
\end{array}
$$

Como usar $P \vee Q$

- Utilizando o esquema da prova por casos.

$$
\begin{array}{ll|ll}
1 & & P \vee Q & \\
2 & & \vdots & \\
3 & & P \to R & \\
4 & & \vdots & \\
5 & & Q \to R & \\
6 & & \vdots & \\
7 & R & & 1, 3, 5, \text{PC}
\end{array}
$$

- Lei do silogismo disjuntivo pela negação do primeiro disjuntor e posteriormente, pela negação do segundo disjuntor.

$$\begin{array}{r|ll}
1 & P \vee Q & \\
2 & \vdots & \\
3 & \neg P & \\
\multicolumn{1}{r}{4} & Q & \text{1, 3, SD}
\end{array}
\qquad\qquad
\begin{array}{r|ll}
1 & P \vee Q & \\
2 & \vdots & \\
3 & \neg Q & \\
\multicolumn{1}{r}{4} & P & \text{1, 3, SD}
\end{array}$$

- Lei do dilema construtivo (em qualquer ordem).

$$\begin{array}{r|ll}
1 & P \vee Q & \\
2 & \vdots & \\
3 & P \to R & \\
4 & \vdots & \\
5 & Q \to S & \\
\multicolumn{1}{r}{6} & R \vee S & \text{1, 3, 5, DC}
\end{array}$$

Como usar $P \leftrightarrow Q$

- Utilizando a eliminação da equivalência ($\leftrightarrow$-el) pelo primeiro membro, posteriormente pelo segundo membro.

$$\begin{array}{r|ll}
1 & P & \\
2 & \vdots & \\
3 & P \leftrightarrow Q & \\
4 & \vdots & \\
\multicolumn{1}{r}{5} & Q & \text{1, 3, }\leftrightarrow\text{-el}
\end{array}
\qquad\qquad
\begin{array}{r|ll}
1 & Q & \\
2 & \vdots & \\
3 & P \leftrightarrow Q & \\
4 & \vdots & \\
\multicolumn{1}{r}{5} & P & \text{1, 3, }\leftrightarrow\text{-el}
\end{array}$$

- Utilizando a lei da substituição da equivalência, primeiramente trocando o primeiro membro pelo segundo, posteriormente, trocando o segundo membro pelo primeiro.

$$
\begin{array}{l|ll}
1 & P_1 \leftrightarrow P_2 & \\
2 & \vdots & \\
3 & Q(S||P_1) & \\
4 & \vdots & \\
\multicolumn{1}{l}{5} & \multicolumn{1}{l}{Q(S||P_2)} & \text{1, 3, SE}
\end{array}
\qquad
\begin{array}{l|ll}
1 & P_1 \leftrightarrow P_2 & \\
2 & \vdots & \\
3 & Q(S||P_2) & \\
4 & \vdots & \\
\multicolumn{1}{l}{5} & \multicolumn{1}{l}{Q(S||P_1)} & \text{1, 3, SE}
\end{array}
$$

Como usar $\neg P$

- Utilizando a lei da não-contradição.

$$
\begin{array}{l|ll}
1 & P & \\
2 & \vdots & \\
3 & \neg P & \\
4 & \vdots & \\
\multicolumn{1}{l}{5} & \multicolumn{1}{l}{Q} & \text{1, 3, NC}
\end{array}
$$

- Utilizando a lei da negação da implicação

$$
\begin{array}{l|ll}
1 & \neg(P \rightarrow Q) & \\
2 & \vdots & \\
\multicolumn{1}{l}{3} & \multicolumn{1}{l}{P \wedge \neg Q} & \text{1, NI}
\end{array}
$$

- Utilizando Lei De Morgan.

$$
\begin{array}{l|ll}
1 & \neg(P \wedge Q) & \\
2 & \vdots & \\
\multicolumn{1}{l}{3} & \multicolumn{1}{l}{\neg P \vee \neg Q} & \text{1, DM}
\end{array}
$$

Como usar $\neg(P \lor Q)$

- Utilizando a lei de negação da conjunção (i) e (ii)

$$
\begin{array}{lll}
1 & \quad \neg(P \lor Q) & \\
2 & \quad \vdots & \\
3 & P \rightarrow \neg Q & 1, \text{NCj(i)}
\end{array}
$$

$$
\begin{array}{lll}
1 & \quad \neg(P \lor Q) & \\
2 & \quad \vdots & \\
3 & Q \rightarrow \neg P & 1, \text{NCj(ii)}
\end{array}
$$

- Utilizando a lei de De Morgan (i)

$$
\begin{array}{lll}
1 & \quad \neg(P \lor Q) & \\
2 & \quad \vdots & \\
3 & \neg P \land \neg Q & 1, \text{DM(i)}
\end{array}
$$

Como usar $\neg\neg P$

- Utilizando a lei da dupla negação.

$$
\begin{array}{lll}
1 & \quad \neg\neg P & \\
2 & \quad \vdots & \\
3 & P & 1, \text{DN}
\end{array}
$$

Como usar $\neg(P \leftrightarrow Q)$

- Utilizando a lei da negação da equivalência (i) (negação do primeiro membro).

$$
\begin{array}{lll}
1 & \quad \neg(P \leftrightarrow Q) & \\
2 & \quad \vdots & \\
3 & (\neg P \leftrightarrow Q) & 1, \text{NEq}
\end{array}
$$

- Utilizando a lei da negação da equivalência (ii) (negação do segundo membro).

$$
\begin{array}{lll}
1 & \quad \neg(P \leftrightarrow Q) & \\
2 & \quad \vdots & \\
3 & (\neg P \leftrightarrow Q) & 1, \text{NEq}
\end{array}
$$

Capítulo 5

Lógica Quantificacional Clássica elementar

Este capítulo apresenta a *Lógica Quantificacional Clássica* ou simplesmente LQC. Esta lógica estuda as propriedades clássicas dos conectivos e de certas variações internas em fórmulas, expressas pelos *quantificadores*.

Todas as convenções e definições dadas anteriormente continuam valendo neste capítulo, com as devidas adaptações, sempre que necessário.

§1. Uma Linguagem para LQC

1.1 Definição.
Um *alfabeto quantificacional* contém os seguintes tipos de sinais, mutuamente disjuntos dois a dois:

- *constantes* – são nomes de objetos definidos do universo de discurso. A coleção de constantes pode ser eventualmente vazia. Exemplos: '2', '0', '4', '$\mathbb{N}$', 'π', 'e'.
- *variáveis* – são nomes de objetos em geral indefinidos do universo de discurso ou expressam alguma quantificação; juntamente com as constantes, constituem a coleção dos *termos atômicos*[1] em uma Lógica Quantificacional. A coleção de variáveis deve ser infinitamente enumerável. Adotamos como variáveis as letras x, y, z, w, seguidas ou não de subíndices: $x, y, z, w, x_1, y_1, z_1, w_1, x_2, y_2, z_2, w_2, x_3, y_3, z_3, w_3, \ldots$

[1] São termos que não contém outros termos.

- *sinais funcionais* – formam termos a partir de uma lista não vazia de termos[2]; se um sinal funcional usa n termos para formar um termo, dizemos que n é uma *aridade*[3] *deste sinal*. A coleção de sinais funcionais pode ser eventualmente vazia. Exemplos: $+$, x, $-$, $/$, !, $\int$. No termo '$2+3$', '2' e '3' também são termos e '$+'$ é um sinal funcional. No termo '4!'[4], '4' também é termo e ! é um sinal funcional.
- sinais predicativos – formam fórmulas a partir de uma lista eventualmente vazia de termos; se um sinal predicativo usa n termos para formar uma fórmula, dizemos que n é uma *aridade*[5] *deste sinal*[6]. Todo alfabeto quantificacional deve possuir pelo menos um sinal predicativo de aridade positiva (> 0). Exemplos: $\in, =, >$. Na fórmula '$2 \in \mathbb{N}$', '2' e '$\mathbb{N}$' são termos e '$\in$' é um sinal predicativo. Na fórmula 'par(8)', '8' é termo e 'par()' é um sinal predicativo de aridade 1. Na fórmula 'pai(José, Jesus)', 'José' e 'Jesus' são termos e 'pai()' é um sinal predicativo de aridade 2. E na fórmula '4+3=2+5', dizemos que '$4+3$' e '$2+5$' são termos e '$=$' é um sinal predicativo.
- *conectivos* – formam fórmulas a partir de fórmulas; se um conectivo usa n fórmulas para formar uma fórmula, dizemos que n é a *aridade* deste sinal; os conectivos são em geral de aridades 1 ou 2. Todo alfabeto quantificacional deve possuir pelo menos um conectivo. Exemplos: $\neg, \rightarrow, \wedge, \vee, \top, \bot$.
- *quantificadores* – formam fórmulas a partir de listas não vazias de variáveis e de fórmulas. Os quantificadores mais comuns só utilizam uma variável e uma fórmula. Todo alfabeto quantificacional deve possuir pelo menos um quantificador. Exemplos: $\forall$, $\exists$, $\exists!$, dentre outros.
- *sinais de pontuação* – são o parêntese de abertura '(', o parêntese de fechamento ')' e a vírgula ','.

1.2 Definição. Um *sinal lógico* é um conectivo, um quantificador, uma variável ou um sinal de pontuação.

1.3 Definição. Um *sinal não lógico* é uma constante, um sinal funcional ou um sinal predicativo.

Estendemos abaixo a definição 4.1.7, considerando também, além das aridades de conectivos, as aridades de sinais funcionais e de sinais predicativos.

---

[2] Pode-se também dizer que as constantes são sinais funcionais de aridade 0, isto é, formam termos a partir de uma lista vazia de termos.

[3] Admitimos que cada sinal funcional possa conter mais de uma aridade.

[4] Fatorial.

[5] Admitimos que cada sinal predicativo possa conter mais de uma aridade.

[6] Os sinais predicativos de aridade zero são também chamados de *letras sentenciais*.

1.4 Definição.
Seja s um conectivo, um sinal funcional ou um sinal predicativo. Se s possui aridade n, dizemos também que s é n*-ário* ou n*-ádico*. Se s possui uma das aridades 1, 2 ou 3, então s é também chamado de *monádico*, *diádico* ou *triádico*, respectivamente.

1.5 Convenção. Um dado sinal não lógico deve possuir a mesma função sintática[7] em qualquer linguagem de uma certa lógica. Por exemplo, o sinal '$+$' é um sinal funcional de aridade dois em qualquer linguagem L de LQC.

1.6 Definição.
Os dois quantificadores mais comuns são:
- '$\forall$', dito o *quantificador universal*;
- '$\exists$', dito o *quantificador existencial*.

1.7 Definição.
Um *alfabeto* para LQC é um alfabeto quantificacional possuindo os conectivos '$\rightarrow$', '$\wedge$','$\vee$', '$\neg$', '$\perp$', '$\top$' e os quantificadores '$\forall$' e '$\exists$'.

1.8 Notação. Além das convenções estabelecidas em 3.1.8, as seguintes referências adicionais serão usadas, para as listas de letras descritas abaixo, seguidas ou não de plicas ou subíndices, a não ser que o contrário seja dito:
- a, b, c – denotam constantes,
- x, y, z, w – denotam variáveis[8],
- f, g, h – denotam sinais funcionais,
- p, q, r – denotam sinais predicativos,
- t, u, v – denotam termos,
- P, Q, R, S – denotam fórmulas,
- D, E, F – denotam designadores,
- Γ, Φ – denotam coleções de fórmulas,
- Ω – denota uma coleção de designadores.

1.9 Definição.
Termos em LQC são samblagens satisfazendo as seguintes condições:
- Toda constante do alfabeto é um termo, dito um *termo atômico*.
- Toda variável do alfabeto é um termo, dito um *termo atômico*.
- Se $t_1, \ldots, t_n$ são termos e f é um sinal funcional n-ário do alfabeto, então f$(t_1, \ldots, t_n)$ é um termo, dito um *termo funcional*.

[7] Ou funcionalidade.

[8] Ou seja, as letras latinas minúsculas x, y, z, w, escritas em itálico, seguidas ou não de plicas ou subíndices, são nomes de variáveis, as quais são as letras latinas minúsculas x,y,z,w, aprumadas, seguidas ou não de subíndices. Daí não pode haver confusão entre as próprias variáveis e os nomes das mesmas.

1.10 Definição.
Fórmulas em LQC são samblagens satisfazendo as seguintes condições:
- Se p é um sinal predicativo de aridade n (ou n-ário) e $t_1, \ldots, t_n$ são termos, então p($t_1, \ldots, t_n$) é uma fórmula, dita *fórmula atômica*.
- $\top$ e $\bot$ são fórmulas.
- Se P é uma fórmula, então $\neg P$ é uma fórmula.
- Se P e Q são fórmulas, então $(P \to Q)$, $(P \land Q)$ e $(P \lor Q)$ são fórmulas.
- Se x é uma variável e P é uma fórmula, então:
  - $\forall x P$ é uma fórmula, dita *fórmula universal*, cujo corpo é o P.
  - $\exists x P$ é uma fórmula, dita *fórmula existencial*, cujo corpo é o P.

1.11 Definição.
Um *designador em* LQC é um termo em LQC ou uma fórmula em LQC.

1.12 Notação. A partir de agora, a menos que o contrário seja dito, adotamos as seguintes convenções adicionais:
- As letras Ψ e Υ, seguidas ou não de subíndices, referem-se a um dos quantificadores '$\forall$' ou '$\exists$'.
- Quando Ψ' aparecer no mesmo contexto que Ψ, $\Psi' = \begin{cases} \exists, \text{ se } \Psi = \forall, \\ \forall, \text{ se } \Psi = \exists, \end{cases}$ e Υ' é definido analogamente.
- O mesmo vale para Ψ'_i e Υ'_i, quando aparecerem respectivamente no mesmo contexto que Ψ_i e Υ_i, onde i é um inteiro positivo.

1.13. **Escrita informal de termos e fórmulas em LQC**
Além das convenções estipuladas em 4.1.16, adotamos as seguintes convenções adicionais, desde que não haja margem para quaisquer confusões:
- Se f é um sinal funcional monádico de escrita pré-fixada, notamos $f(t)$ por (ft).
  Quando (ft) não está no interior de outro termo, podemos prescindir do seu par exterior de parênteses.[9]
- Se f é um sinal funcional monádico de escrita pós-fixada, notamos $f(t)$ por (tf).
  Quando (tf) não está no interior de outro termo, podemos prescindir do seu par exterior de parênteses.[10]
- Se f é um sinal funcional diádico, notamos $f(t_1, t_2)$ por ($t_1\ f\ t_2$).
  Quando ($t_1\ f\ t_2$) não estiver escrito como subtermo de outro termo, pode-

[9] Por exemplo, as samblagens '(-x)' e '(+y)' representam respectivamente os termos '-(x)' e '+(y)'. Quando '(-x)' e '(+y)' não estiverem dentro de outro termo, os mesmos podem ser escritos como '-x' e '+y'.

[10] Por exemplo, se x denota um número natural, notamos o fatorial de x como '(x!)'. Se '(x!)' não está no interior de outro termo, o mesmo pode ser escrito como 'x!'.

mos prescindir do par exterior de parênteses.[11]

- Quando o mesmo sinal funcional diádico ocorrer em um termo, podemos suprimir no mesmo todos os pares internos de parênteses, considerando a parentetização implícita da esquerda para a direita.[12]
- Se p é um sinal predicativo de aridade 2, notamos $p(t_1, t_2)$ por $(t_1\, p\, t_2)$. Quando $(t_1\, p\, t_2)$ não estiver escrito como subfórmula de outra fórmula, podemos prescindir do par exterior de parênteses.[13]
- Quando $(t_1\, p\, t_2)$ for um dos componentes de uma fórmula formada por um conectivo diádico, podemos prescindir do seu par exterior de parênteses.[14]

1.14. **Leitura de fórmulas quantificadas**

- '$\forall x P$' pode ser lido de uma das seguintes formas:
  - 'Para todo x, P',
  - 'Para cada x, P',
  - 'Para qualquer x, P',
  - 'Qualquer que seja x, P'.
- '$\exists x P$' pode ser lido de uma das seguintes formas:
  - 'Existe x tal que P',
  - 'Existe pelo menos um x tal que P',
  - 'Para algum x, P'.

Existem dois sinais predicativos de aridade 'dois' especialmente importantes, respectivamente para a *Lógica Equacional* e a *Teoria dos Conjuntos*, a saber:

- '=' : exemplo '$t = u$'; é lido como 't é igual a u' ou 't é idêntico a u';
- '$\in$' : exemplo '$t \in A$'; é lido como 't pertence ao conjunto A'.

O seguinte lema define a cardinalidade da coleção de fórmulas de uma linguagem L para LQC e será útil para demonstrar a completude do cálculo de LQC apresentada em 10.3.2.

---

[11] Por exemplo, a samblagem '$(x + y)$' representa o termo '+(x, y)'. Se '$(x + y)$' não está no interior de outro termo, este pode ser escrito como 'x + y'.

[12] Por exemplo, a samblagem '$x + y + z + w$' representa o termo '$((x + y) + z) + w$'.

[13] Por exemplo, a samblagem '$(x \in A)$' representa a fórmula '$\in (x, A)$'. Se '$(x \in A)$' não está no interior de outra fórmula, esta pode ser escrita como '$x \in A$'.

[14] Por exemplo, a samblagem '$x \in A \to x \in B$' representa a fórmula '$(x \in A) \to (x \in B)$'. No entanto, quando $(t_1\ p\ t_2)$ for uma fórmula negada ou for o corpo de uma fórmula universal ou existencial, o par exterior de parênteses é indispensável para uma boa legibilidade da fórmula completa. Por exemplo, nas fórmulas $\neg(x \in A)$ e $\exists x(x \in A)$, se fosse suprimido o par exterior de parênteses de '$(x \in A)$', o resultado seriam as expressões '$\neg x \in A$' e '$\exists x\ x \in A$', as quais não apresentam uma boa legibilidade.

1.15 Lema. ***(Da cardinalidade da coleção de fórmulas de uma linguagem*** L ***para*** LQC*)*

Sejam $\begin{cases} \eta \text{ a cardinalidade da coleção de sinais não lógicos}^a \text{ de } L. \\ \#(L) \text{ a cardinalidade da coleção de fórmulas em } L. \end{cases}$

[a] As constantes, os sinais funcionais e os sinais predicativos são considerados *sinais não lógicos*. Eles podem variar entre as linguagens de LQC. Os outros sinais são comuns a qualquer linguagem para LQC.

Então,

(i) *Se* η *é finito ou* $\eta = \aleph_0$*, então* $\#(L) = \omega$.
(ii) *Se* $\eta \geq \aleph_0$*, então* $\#(L) = \eta$.

Se existe uma linguagem L cuja quantidade de sinais não lógicos é finita, ou seja, a cardinalidade dos sinais não lógicos é finita, então existe um número finito de constantes, sinais funcionais ou sinais predicativos. Neste caso, a quantidade de fórmulas dessa linguagem vai ser infinita e igual à ω, ou seja, igual à cardinalidade da coleção dos números naturais $\mathbb{N}$.
Exemplo: Seja $\{p, c_1, c_2\}$ a coleção finita de sinais não lógicos de L em $\mathcal{L}$.
A quantidade de fórmulas pode ser infinita:
$p(c_1); p(c_1) \wedge p(c_2); p(c_1) \wedge p(c_1) \wedge p(c_1) \wedge \ldots; p(c_2)$, dentre outras.
Prova-se o lema acima através de recursos de teoria dos conjuntos.

Abaixo é formulada, em termos precisos, a não ambiguidade concernente a qualquer linguagem para LQC.

1.16. **Teorema da Legibilidade Única**
Cada termo e cada fórmula em LQC só podem ser lidos de uma única forma, isto é:

- As coleções de constantes, variáveis, termos funcionais, fórmulas atômicas, negações, implicações, conjunções, disjunções, fórmulas universais e fórmulas existenciais são duas a duas disjuntas.
- Se $f(t_1, \ldots, t_n)$ e $g(u_1, \ldots, u_p)$ são termos funcionais idênticos, então $f = g$, $n = p$; e, para cada $i \in \{1, \ldots, n\}$, $t_i = u_i$.
- Se $p(t_1, \ldots, t_n)$ e $q(u_1, \ldots, u_p)$ são fórmulas atômicas idênticas, então $p = q$, $n = p$; e, para cada $i \in \{1, \ldots, n\}$, $t_i = u_i$.
- $\neg P$ e $\neg Q$ são negações idênticas, então $P = Q$.
- Se $(P \rightarrow Q)$ e $(R \rightarrow S)$ são implicações idênticas, então $P = R$ e $Q = S$.
- Se $(P \wedge Q)$ e $(R \wedge S)$ são conjunções idênticas, então $P = R$ e $Q = S$.
- Se $(P \vee Q)$ e $(R \vee S)$ são disjunções idênticas, então $P = R$ e $Q = S$.
- Se $\forall x P$ e $\forall y Q$ são fórmulas universais idênticas, então $x = y$ e $P = Q$.
- Se $\exists x P$ e $\exists y Q$ são fórmulas existenciais idênticas, então $x = y$ e $P = Q$.

Variáveis

Existem dois conceitos sintáticos essenciais para a formulação de regras para uma Lógica Quantificacional: a condição de uma dada *variável* ser *livre em uma fórmula* e a *instanciação de uma variável por um termo em uma fórmula*.

1.17 Definição. (**Ocorrências livres ou ligadas**)
Uma *ocorrência de uma variável* em um termo ou uma fórmula é uma cópia específica da mesma neste termo ou fórmula.[15] Todas as *ocorrências de uma variável* em um termo são ditas serem *livres neste termo*.[16] Uma *ocorrência de uma variável* x em uma fórmula P é dita ser *ligada em* P se esta figurar em uma subfórmula de P de uma das formas $\forall x Q$ ou $\exists x Q$; caso contrário esta *ocorrência* é dita ser *livre em* P.

1.18 Definição. (**Variáveis livres ou ligadas**)

- Uma *variável* é dita ser *livre em um termo* se esta possuir pelo menos uma ocorrência neste termo.[17]
- Uma *variável* é dita ser *ligada em uma fórmula* se esta possuir pelo menos uma ocorrência ligada nesta fórmula. Da mesma forma, uma *variável* é dita ser *livre em uma fórmula* se esta possuir pelo menos uma ocorrência livre nesta fórmula.

1.19 Definição. As cláusulas abaixo definem ocorrências livres e ligadas de uma variável em um designador, e quando uma variável é livre em um designador ou em uma coleção de designadores.

- Uma *ocorrência de uma variável* x em um designador D é dita ser *ligada* em D, caso esta ocorrência ocorra dentro de um subdesignador de D de uma das formas $\forall x P$ ou $\exists x P$, caso contrário, esta ocorrência é dita *livre em* D.
- Uma *variável* x é dita *livre em um designador* D se esta possui ao menos uma ocorrência livre em D.
- Uma *variável* x é dita *livre em uma coleção* Ω *de designadores*, se existe $D \in \Omega$ tal que x é livre em D.

1.20 Definição. (**Termo fechado**)
São termos que não contém variáveis livres.

---

[15] Por exemplo, a fórmula $p(x, g(x, y))$ possui duas ocorrências de x e somente uma ocorrência de y.

[16] Isto não acontece nas Lógicas Descritivas, onde ocorrências de variáveis sucedendo qualificadores ou certas variáveis no corpo de descrições não são livres nos termos que as contém.

[17] Em linguagens nas quais termos podem possuir variáveis ligadas, como, por exemplo, nas linguagens para a Lógica Descritiva Clássica ou para a Lógica das Descrições Indefinidas, dadas adiante, uma variável pode ocorrer em um termo e ainda assim não ser livre neste termo.

1.21 Definição. Um *designador* é dito *fechado* se ele não contém variáveis livres.

Se x é livre em uma fórmula P, isto significa, intuitivamente, que P *fala do* objeto designado por x através de suas ocorrências livres em P. Se x não é livre em P, então *não fala* do objeto *designado* por x através de suas ocorrências livres em P, mesmo que x ocorra em P.

Por exemplo, considere a fórmula $\forall x\ (x \in \mathbb{Z} \wedge par(x^2) \rightarrow par(x))$. Esta fórmula não fala do objeto designado por x através de suas ocorrências livres, pelo fato de que x não é livre na mesma. Podemos reescrevê-la por '$\forall y\ (y \in \mathbb{Z} \wedge par(y^2) \rightarrow par(y))$', sem alterar o seu significado original. Temos assim que a fórmula $\forall x\ (x \in \mathbb{Z} \wedge par(x^2) \rightarrow par(x))$ não possui essencialmente nenhuma relação especial com a variável x.

1.22 Fato. *y não é livre em $\forall x R$ sss $y = x$ ou y não é livre em R.*

1.23 Fato.
As seguintes cláusulas especificam recursivamente quando uma variável é livre em uma fórmula:

- *x é livre em $p(t_1, \ldots, t_n)$ sss existe $i \in \{1, \ldots, n\}$ tal que x é livre em t_i.*
- *x é livre em $\neg P$ sss x é livre em P.*
- *x é livre em $P \# Q$ sss x é livre em P ou x é livre em Q.*
- *x não é livre em $\Psi x P$.*
- *x é livre em $\Psi y P$ sss $x \neq y$ e x é livre em P.*

1.24 Definição. Em $\forall x\, r(z), \exists y\, r(z) \vdash \exists z \forall x t(z, x)$, a primeira ocorrência do quantificador universal é vácua pois é seguida de uma fórmula, $r(z)$, onde z é livre em $\forall x$. O mesmo ocorre para a primeira ocorrência do quantificador existencial.

Instanciação Simples e Congruência

O conceito de *instanciação de variáveis por termos* é essencial para a formulação de três das quatro *leis de introdução e eliminação de quantificadores*, as quais integram a definição do cálculo de sequentes para LQC, dado na seção seguinte.

Queremos definir a *instanciação de x por t em P*, que passaremos a notar por $P(x|t)$, de modo que $P(x|t)$ fale do objeto designado por t através de todas as novas ocorrências de t em $P(x|t)$, da mesma forma que P fala do objeto designado por x.

A partir do que foi dito há pouco, poderíamos definir $P(x|t)$ como sendo a fórmula obtida de P substituindo todas as ocorrências livres de x por t. Tal definição leva a certos problemas, como veremos a seguir. Por exemplo, sabemos que, em um *universo de discurso*[18] com pelo menos dois objetos, considerando a interpretação usual para o sinal '$\neq$', a fórmula $\forall x \exists y(x \neq y)$ é verdadeira, isto é, para qualquer objeto x temos que $\exists y(x \neq y)$, daí a fórmula $\exists y(x \neq y)$ deveria continuar a ser verdadeira se nela a variável x for instanciada por um termo qualquer. Se instanciarmos x por y em $\exists y(x \neq y)$ segundo esta nossa primeira abordagem, concluiríamos erradamente que $\exists y(y \neq y)$ é verdadeiro, o que é absurdo. O problema todo se origina ao instanciar x por uma ocorrência de y que passou a ser ligada na fórmula $\exists y(y \neq y)$, o que altera completamente a estrutura da fórmula original.

Para contornar isto, podemos observar que a fórmula $\exists z(x \neq z)$ é equivalente à fórmula original $\exists y(x \neq y)$, dada inicialmente, no sentido que a segunda afirma exatamente o que dizia a primeira. Assim, para instanciar x por y em $\exists y(x \neq y)$, devemos primeiro alterar a variável quantificada na fórmula original para uma nova variável não ocorrendo nem na fórmula $(x \neq y)$ nem no termo y, a qual pode ser a variável z, obtendo assim a fórmula $\exists z(x \neq z)$. Instanciando x por y nesta nova fórmula, obtemos a fórmula $\exists z(y \neq z)$, a qual é obviamente verdadeira no contexto por nós considerado.

Assim, conforme a segunda abordagem, $(\exists y(x \neq y))(x|y)$ é a fórmula $\exists z(y \neq z)$. Isto conduz a um novo problema, pois a variável z pode ser escolhida de uma infinidade de maneiras, pois existe um número infinito de variáveis não ocorrendo em $(x \neq y)$ nem em y, daí a nossa nova definição de $(\exists y(x \neq y))(x|y)$ leva a uma infinidade de fórmulas, o que é uma ambiguidade.

Para evitar isto, iremos considerar uma ordenação fixa na coleção das variáveis, e iremos sempre escolher, para fins de instanciação, a primeira variável preenchendo as condições requeridas. Veremos posteriormente que a escolha da variável não altera o significado original da fórmula instanciada.

1.25 Convenção. A lista infinita de variáveis '$x, y, z, w, x_1, y_1, z_1, w_1, x_2, y_2, z_2, w_2, x_3, y_3, z_3, w_3, \ldots$' dá a ordenação que será considerada ao longo de todo este trabalho.

1.26 Exemplo.
Conforme a ordenação fixa das variáveis considerada a partir de agora, temos que, como z é a primeira variável não ocorrendo em $(x \neq y)$ nem em y, $(\exists y(x \neq y))(x|y)$ é a fórmula $\exists z(y \neq z)$.

1.27 Definição.
Os enunciados abaixo especificam a instanciação de variáveis por termos em termos, em fórmulas e em coleções de fórmulas:

[18] O universo de discurso é a coleção subentendida de valores que as variáveis podem assumir em um dado contexto.

(i) A *instanciação de x por t em u*, notada por $u(x|t)$, é o termo obtido de u substituindo todas as ocorrências de x por t.

(ii) A instanciação de x por t em P, notada por $P(x|t)$, é a fórmula obtida de P substituindo todas as ocorrências livres de x por t, se P não possuir quantificadores. Caso houverem quantificadores na fórmula envolvida, então tal instanciação é definida conforme as seguintes cláusulas, onde x e y são variáveis distintas:

- $(\Psi x P)(x|t) = \Psi x P$;
- $(\Psi y P)(x|t)$ [19] [20] [21] $= \begin{cases} * & \Psi y P(x|t), \text{ se } x \text{ não é livre em } P \text{ ou } y \text{ não é livre em } t, \\ * & \Psi z P(y|z)(x|t), \text{ em caso contrário, onde } z \text{ é a primeira variável não livre em } \{t, P\}. \end{cases}$

(iii) A *instanciação de x por t em Γ*, notada por $\Gamma(x|t)$, é a coleção $\{P(x|t) | P \in \Gamma\}$.

1.28 Lema.

- *Se* $\begin{cases} x \neq y, \\ x \text{ é livre em } P, \end{cases}$ *então* $(\Psi y P)(x|t) = \Psi z P(y|z)(x|t)$*, onde*
$\begin{cases} * & z = y, \text{ se } y \text{ não é livre em } t, \\ * & z \text{ é a primeira variável não livre em } \{t, P\}, \text{ em caso contrário.} \end{cases}$

1.29 Lema.

- *Se* $\begin{cases} x \neq y, \\ x \text{ é livre em } P, \end{cases}$ *então existe uma variável z tal que*
$\begin{cases} z \text{ não é livre em } \{x, t, \forall y P\}, \\ (\Psi y P)(x|t) = \Psi z P(y|z)(x|t). \end{cases}$

1.30 Fato. *$P(x|t) = P$ se, e somente se, $x = t$ ou x não é livre em P.*

1.31 Fato. *y é livre em $P(x|t)$ se, e somente se, y é livre em $\forall x P$ ou* $\begin{cases} x \text{ é livre em } P, \\ y \text{ é livre em } t. \end{cases}$

Para provar os resultados básicos de instanciação, há uma relação sintática fundamental, que é a *aceitação de uma variável por um termo em uma fórmula*. Esta nova relação é baseada por sua vez em uma outra ideia ainda mais básica, a do *escopo forte de uma variável*.

---

[19] O caso em que y não é livre em t contém a possibilidade de que $x = t$, pois estamos considerando que $x \neq y$.

[20] A primeira condição desta cláusula pode ser desdobrada em duas condições, e daí a formulação seria desta forma: $(\Psi y P)(x|t) = \Psi y P$, se x não é livre em P, e $(\Psi y P)(x|t) = \Psi y P(x|t)$, se y não é livre em t.

[21] No segundo caso, como x é livre em P e z não é livre em P, temos que $z \neq x$, ou seja, z não é livre em $\{x, t, P\}$.

1.32 Definição.
Dizemos que x *está no escopo forte de* y *em* P se x possuir uma ocorrência livre em P figurando em alguma subfórmula de P de uma das formas $\forall yQ$ ou $\exists yQ$.

1.33 Escólio. *Se* y *não é ligado em* P*, então* x *não está no escopo forte de* y *em* P.

1.34 Escólio. *Se* x *está no escopo forte de* y *em* P*, então* x *e* y *são variáveis distintas.*

1.35 Notação. Dadas duas variáveis x e y e uma fórmula P, a expressão '$escf(x, y, P)$' abrevia a proposição 'x está no escopo forte de y em P'.

1.36 Fato. *As cláusulas a seguir especificam recursivamente quando uma variável está no escopo forte de outra variável em uma fórmula:*
(i) *Não é o caso que* $escf(x, y, p(t_1, \ldots, t_n))$.[22]
(ii) $escf(x, y, \neg P)$ *sss* x $escf(x, y, P)$.
(iii) $escf(x, y, P\#Q)$ *sss* $escf(x, y, P)$ *ou* $escf(x, y, Q)$.
(iv) $escf(x, y, \Psi zP)$ *sss* $\begin{cases} x \neq z, \\ y = z \text{ implica que } x \text{ é livre em } P, \\ y \neq z \text{ implica que } escf(x, y, P). \end{cases}$

1.37 Definição. Dizemos que x *aceita* t *em* P se x não estiver em P no escopo forte de alguma variável livre em t.

1.38 Escólio. *Se nenhuma variável livre em* t *é ligada em* P*, então* x *aceita* t *em* P.

1.39 Notação. Dada uma variável x, um termo t e uma fórmula P, a expressão '$ac(x, t, P)$' abrevia a proposição 'x aceita t em P'.

1.40 Fato.
As cláusulas a seguir especificam recursivamente quando uma variável aceita um termo em uma fórmula:
(i) $ac(x, t, p(t_1, \ldots, t_n))$.
(ii) $ac(x, t, \neg P)$ *sss* $ac(x, t, P)$.
(iii) $ac(x, t, P\#Q)$ *sss* $ac(x, t, P)$ *e* $ac(x, t, Q)$.
(iv) $ac(x, t, \Psi yP)$ *sss* $x = y$ *ou* y *não é livre em* P *ou* $\begin{cases} y \text{ não é livre em } t, \\ ac(x, t, P). \end{cases}$

[22] Isto não é verdadeiro em linguagens nas quais termos possuem variáveis ligadas, como é o caso nas Lógicas Descritivas.

1.41 Lema.
As seguintes leis fundamentam as relações entre a aceitação e a instanciação simples:

(i) *Se* $\begin{cases} y \text{ não é livre em } \forall x P, \\ x \text{ aceita } y \text{ em } P, \end{cases}$ *então* $ac(x, t, P)$ *sss* $ac(y, t, P(x|y))$.

(ii) *Se* y *não é livre em* $\forall x P$, *então* $ac(x, y, P)$ *sss* $ac(y, t, P(x|y))$.

(iii) *Se* $\begin{cases} x \neq y, \\ x \text{ não é livre em } u, \\ y \text{ aceita } u \text{ em } P, \end{cases}$ *então* $ac(x, t, P)$ *sss* $ac(x, t, P(y|u))$.

Damos a seguir algumas leis sintáticas fundamentais a respeito da instanciação simples, as quais são requisitos para a dedução de outras leis mais avançadas concernentes à instanciação e à congruência.

1.42 Lema.

(i) *Se* $\begin{cases} y \text{ não é livre em } \forall x P, \\ x \text{ aceita } y \text{ em } P, \\ x \text{ aceita } t \text{ em } P, \end{cases}$ *então* $P(x|y)(y|t) = P(x|t)$.

(ii) *Se* $\begin{cases} y \text{ não é livre em } \forall x P, \\ x \text{ aceita } y \text{ em } P, \end{cases}$ *então* $P(x|y)(y|x) = P$.

(iii) *Se* $\begin{cases} x \neq y, \\ x \text{ não é livre em } u, \\ y \text{ não é livre em } t, \\ x \text{ aceita } t \text{ em } P, \\ y \text{ aceita } u \text{ em } P, \end{cases}$ *então* $P(x|t)(y|u) = P(y|u)(x|t)$.

Há uma relação entre fórmulas que é fundamental para a enunciação de vários resultados sintáticos concernentes à instanciação de variáveis, que é a relação de *congruência entre fórmulas*.

1.43 Definição.
As condições abaixo definem a *relação de congruência entre duas fórmulas em* LQC, notada por '$\approx_c$' :

- $p(t_1, \ldots, t_n) \approx_c q(u_1, \ldots, u_p)$ se, e somente se, $p = q$, $n = p$ e $t_i = u_i$, para qualquer $i = 1, \ldots, n$.[23]
- $\neg P_1 \approx_c \neg P_2$ se, e somente se, $P_1 \approx_c P_2$.
- $P_1 \# Q_1 \approx_c P_2 \# Q_2$ se, e somente se, $P_1 \approx_c P_2$ e $Q_1 \approx_c Q_2$.
- $\Psi x P_1 \approx_c \Psi y P_2$ se, e somente se,$\begin{cases} y \text{ não é livre em } \Psi x P_1, \\ P_1(x|y) \approx_c P_2. \end{cases}$.
- Se P_1 e P_2 são fórmulas de tipos diferentes, então $P_1 \not\approx_c P_2$.[24]

As seguintes proposições concernentes à relação de congruência entre fórmulas são válidas.

[23] Ou seja, duas fórmulas atômicas são congruentes se, e somente se, elas forem iguais.

[24] Isto é, não é verdade que $P_1 \approx_c P_2$.

1.44. **Leis sintáticas fundamentais da congruência de fórmulas**

(i) $P \approx_c P$.
(ii) Se $P \approx_c Q$, então P e Q possuem as mesmas variáveis livres.
(iii) Se $P \approx_c Q$, então $gr(P) = gr(Q)$.
(iv) Existe P' tal que $P \approx_c P'$ e $x_1, \ldots x_n$ aceitam respectivamente $t_1, \ldots t_n$ em P'.
(v) Se $P \approx_c Q$, então $Q \approx_c P$.
(vi) Se $P \approx_c Q$ e $Q \approx_c R$, então $P \approx_c R$.
(vii) Se $P \approx_c Q$, então $P(x|t) \approx_c Q(x|t)$.

Finalmente, com base nas leis acima, podemos deduzir as seguintes leis sintáticas da instanciação:

1.45. **Leis fundamentais da instanciação**

(i) Se y não é livre em $\forall x P$, então $P(x|y)(y|t) \approx_c P(x|t)$.
(ii) Se y não é livre em $\forall x P$, então $P(x|y)(y|x) \approx_c P$.
(iii) Se $\begin{cases} x \text{ e } y \text{ são variáveis distintas,} \\ x \text{ não é livre em } u, \\ y \text{ não é livre em } t, \end{cases}$ então $P(x|t)(y|u) \approx_c P(y|u)(x|t)$.
(iv) Se $\begin{cases} x \text{ e } y \text{ são variáveis distintas,} \\ z \text{ não é livre em } \{x, t, \forall y P\}, \end{cases}$
então $\Psi y P(x|t) \approx_c \Psi z P(y|z)(x|t)$.

Instanciação X Substituição

Além da *instanciação de variáveis por termos em fórmulas*, existe outra operação importante de manipulação de fórmulas, que é a *substituição de fórmulas por fórmulas em fórmulas*. Enquanto a primeira operação sintática é fundamental para a formulação de três das quatro leis de introdução e eliminação de quantificadores e suas consequências, a segunda operação é essencial para a enunciação das leis de substituição para equivalência. O conceito de substituição é tratado de forma mais detalhada na seção § 10.1.

A nosso pensar, existe uma certa confusão entre os conceitos de instanciação e substituição em diversos livros-textos sobre lógica. A par disso, apresentamos alguns exemplos e uma tabela comparativa com o resumo das características relacionadas aos mesmos.

1º exemplo:

- Instanciação:
  $(\forall x\, p(x) \wedge q(x))(x|f(y, z)) = \forall x\, p(x) \wedge q(f(y, z))$.
- Substituição:
  $(\forall x\, p(x) \wedge q(x))(x||f(y, z)) = \forall x\, p(f(y, z)) \wedge q(f(y, z))$.

2º exemplo:

- Instanciação:

$$(\forall x\, p(x,y) \wedge \exists y\, q(z,y,x))(x|f(y,z))$$
$$=$$
$$(\forall x\, p(x,y) \wedge \exists w\, q(z,w,x))(x|f(y,z))$$
$$=$$
$$\forall x\, p(x,y) \wedge \exists w\, q(z,w,f(y,z)).$$

- Substituição:

$$(\forall x\, p(x,y) \wedge \exists y\, q(z,y,x))(x||f(y,z))$$
$$=$$
$$\forall x\, p(f(y,z),y) \wedge \exists y\, q(z,y,f(y,z)).$$

3º exemplo:

- Instanciação:

$$(\forall x\, p(x,y))(y|f(x,z))$$
$$=$$
$$(\forall w\, p(w,y))(y|f(x,z))$$
$$=$$
$$\forall w\, p(w,f(x,z)).$$

- Substituição:

$$(\forall x\, p(x,y))(y||f(x,z))$$
$$=$$
$$\forall x\, p(x,f(x,z)).$$

Tabela 5.1: Comparativo - Instanciação X Substituição

	INSTANCIAÇÃO	SUBSTITUIÇÃO
Âmbito	variável por termo	termo por termo, fórmula por fórmula
Aplicação	leis de introdução e eliminação de quantificadores e leis derivadas	leis de substituição em geral
Renomeação de variáveis	sim	não
Definição	troca de ocorrências livres de variáveis com eventuais renomeações	troca de ocorrências reais de termos ou fórmulas sem renomeações
Símbolo	(x | y)	(x || y)

§2. Uma Semântica para LQC

Esta seção define uma *semântica para* LQC.

2.1 Definição. Os *valores veritativos em* LQC são v e f, onde o único *valor distinguido* é v. Uma ordem é considerada na coleção de valores veritativos em LQC, em que $\mathrm{f} < \mathrm{v}$.

2.2 Definição. Um *universo* é um conjunto não vazio.

2.3 Notação. A menos que seja dito o contrário:
- Δ denota um universo.
- d denota um elemento de Δ.

2.4 Definição.
Um *mundo sobre* Δ é uma função w, cujo domínio – notado aqui por $\mathcal{D}(w)$ – é a coleção de sinais não lógicos[25] satisfazendo as seguintes condições:
- se c é uma constante e $c \in \mathcal{D}(w)$, então $w(c) \in \Delta$.
- se f é um sinal funcional n-ário tal que $f \in \mathcal{D}(w)$, então $w(f)$ é uma função de Δ^n para Δ.
- se p é um sinal predicativo n-ário tal que $p \in \mathcal{D}(w)$, então $w(p) \in \Delta^n$.

2.5 Exemplo.
Seja w um mundo sobre {Mercúrio, Vênus, ..., Netuno}.
Seja o domínio $\mathcal{D}(w)$={{Mercúrio, ..., Netuno}, maior, mais distante, seguinte}.
Considerando que {Mercúrio,..., Netuno} são constantes, *maior* e *mais distante* são sinais predicativos diádicos e *seguinte* é um sinal funcional monádico:
w('Mercúrio') = Mercúrio.
w('Netuno') = Netuno.
w(maior)={$\langle x, y\rangle | x, y$ são planetas $\wedge$ x é maior que y}.
w(mais distante) = {$\langle x, y\rangle | x, y$ são planetas $\wedge$ x é mais distante do sol que y}.
Uma interpretação para este último caso seria $\langle Netuno, Saturno\rangle \in w$(mais distante).

2.6 Definição.
As cláusulas abaixo explicam o que significa um mundo para um designador e um mundo para uma coleção de designadores:
- w é um mundo se w é um mundo sobre Δ, para algum universo Δ.
- w é um mundo para um designador D se cada constante, sinal funcional ou sinal predicativo ocorrendo em D pertencem ao domínio de w.

[25] Considere a definição 1.3.

- w é um mundo para uma coleção Ω de designadores se w é um mundo para D, para cada $D \in \Omega$.

2.7 Definição. (**Δ-atribuição para variáveis**)
Uma Δ-*atribuição para variáveis* s é uma função da coleção de variáveis para Δ.

2.8 Exemplo. Seja s uma $\mathbb{N}$ atribuição definida por $s(x) = 1, s(y) = 2, \ldots$

2.9 Notação. A menos que seja dito o contrário, s é uma Δ-*atribuição para variáveis*, para algum universo Δ.

2.10 Definição.
Seja s uma Δ-atribuição para variáveis e $d \in \Delta$. $s(x|d)$ é uma Δ-atribuição para variáveis como é especificado abaixo:
$s(x|d)(y) = \begin{cases} s(y), \text{ se } x \neq y. \\ d, \text{ se } x = y. \end{cases}$
$s(x|d)$ pode diferir de s, possivelmente, apenas em relação ao valor de x com respeito a s.

2.11 Exemplo. Considere o exemplo 2.8. Então para $s(x_1|7)$ temos que $s(x) = 1$ e $s(x_1) = 7$.

2.12 Definição. (LQC-**interpretação**)
Uma LQC-*interpretação* é uma tripla $\langle \Delta, w, s \rangle$, tal que Δ é um universo de discurso, w é um mundo sobre Δ e s é uma Δ-atribuição para variáveis.

2.13 Notação. $I = \langle \Delta, w, s \rangle$.

2.14 Definição.
$I(x|d) = \langle \Delta, w, s(x|d) \rangle$.
$I(x|d)$ é uma LQC-interpretação que pode diferir de I, possivelmente, apenas no que diz respeito à sua Δ-atribuição para variáveis.

2.15 Definição.
- I é uma LQC-*interpretação para um designador* D se w é um mundo para D.
- I é LQC-*interpretação para uma coleção de designadores* Ω se w é um mundo para Ω.
- t é um I-*termo* se t é termo e I é uma LQC-interpretação para t.
- P é uma I-*fórmula* se P é fórmula e I é uma LQC-interpretação para P.
- D é um I-*designador* se D é um designador e I é uma LQC-interpretação para D.

2.16 Definição.
Considerando uma LQC-interpretação $I = \langle \Delta, w, s \rangle$, as seguintes cláusulas especificam duas funções I_D e I_V, as quais são respectivamente a LQC-*denotação definida por* I e a LQC-*valoração definida por* I:

- I_D é uma função da coleção de todos os I-termos para Δ.
- Se c é uma constante e $c \in \mathcal{D}(w)$, então $I_D(c) = w(c)$.
- Se x é uma variável então $I_D(x) = s(x)$.
- Se f é um sinal funcional n-ário, $f \in \mathcal{D}(w)$ e $t_1, \ldots, t_n$ são I-termos, então $I_D(f(t_1, \ldots, t_n)) = w(f)(I_D(t_1), \ldots, I_D(t_n))$.
- I_V é uma função da coleção de todas as I-fórmulas para {v,f}.
- Se p é um sinal predicativo n-ário, $p \in \mathcal{D}(w)$ e $t_1, \ldots, t_n$ são I-termos, então $I_V(p(t_1, \ldots, t_n)) =$ v sss $\langle I_D(t_1), \ldots, I_D(t_n) \rangle \in w(p)$.

Se P e Q são I-fórmulas, então:

- $I_V(\neg P) \neq I_V(P)$.
- $I_V(P \to Q) = \begin{cases} I_V(Q), \text{ se } I_V(P) = \text{v}, \\ \text{v, se } I_V(P) = \text{f}. \end{cases}$
- $I_V(P \wedge Q) = \min\{I_V(P), I_V(Q)\}$.
- $I_V(P \vee Q) = \max\{I_V(P), I_V(Q)\}$.
- $I_V(\top) = \text{v}$.
- $I_V(\bot) = \text{f}$
- $I_V(\forall x P) = \min\{I(x|d)_V(P) \mid d \in \Delta\}$.
- $I_V(\exists x P) = \max\{I(x|d)_V(P) \mid d \in \Delta\}$.

2.17 Definição.

- $I(x|d)_V(P) \rightleftharpoons$ o valor veritativo de P (com respeito a I) quando a x é atribuído o valor d.
- $I(x|d)_D(u) \rightleftharpoons$ o objeto associado a u (por I) quando a x é atribuído o valor d, sendo $d \in \Delta$.

2.18 Proposição.
Se y não é livre em $\forall x P$, então $I(y|d)_V(P(x|y)) = I(x|d)_V(P)$.

2.19 Definição. (Funções Booleanas)

Sejam $\begin{cases} \bullet\ \mathcal{D}_{LC} \text{ o domínio veritativo da Lógica Clássica,} \\ \bullet\ A \text{ um subconjunto de } \mathcal{D}_{LC} \text{ que também é elemento de } \wp(\mathcal{D}_{LC}), \\ \bullet\ b_1, b_2 \text{ representações de valores booleanos,} \\ \bullet\ \max \text{ e } \min\text{, um valor máximo e um valor mínimo, respectivamente.} \end{cases}$

Considere as seguintes definições booleanas[26]

[26] Chamamos essa função de 'B' em correspondência ao 'B' de George *B*oole. Tais definições são úteis para simplificar alguns lemas da correção de LQC.

- $B_\neg : \{v, f\} \to \{v, f\}$[27].
  $B_\neg(b) \neq B(b)$.
- $B_\to : \{v, f\}^2 \to \{v, f\}$[28].
  $B_\to(b_1, b_2) = \begin{cases} b_2, \text{ se } b_1 = \text{v}, \\ \text{v, caso contrário.} \end{cases}$
- $B_\wedge : \{v, f\}^2 \to \{v, f\}$.
  $B_\wedge : B_\wedge(b_1, b_2) = \min\{b_1, b_2\}$[29].
- $B_\vee : \{v, f\}^2 \to \{v, f\}$.
  $B_\vee : B_\vee(b_1, b_2) = \max\{b_1, b_2\}$[30].
- $B_\forall : \wp(\mathcal{D}_{LC}) \to \{v, f\}$.
  $A \mapsto \min(A)$.
- $B_\exists : \wp(\mathcal{D}_{LC}) \to \{v, f\}$.
  $A \mapsto \max(A)$.

2.20 Escólio.
- $I_V(\neg P) = B_\neg(P)$.
- $I_V(P \to Q) = B_\to(I_V(P), I_V(Q))$.
- $I_V(P \wedge Q) = B_\wedge(I_V(P), I_V(Q))$.
- $I_V(P \vee Q) = B_\vee(I_V(P), I_V(Q))$.
- $I_V(\forall xP) = B_\forall(\{I(x|d)_V(P)|d \in \Delta\})$.
- $I_V(\exists xP) = B_\exists(\{I(x|d)_V(P)|d \in \Delta\})$.

2.21 Escólio.
Se x, y são variáveis distintas, então $I(x|d_1)(y|d_2) = I(y|d_2)(x|d_1)$.

2.22 Definição.
Considere $I = \langle \Delta, w, s \rangle$ e $I' = \langle \Delta, w', s' \rangle$.
I e I' coincidem em um designador D se as seguintes condições são satisfeitas:
- I e I' são interpretações para D.
- Para cada variável livre x em D, $s(x) = s'(x)$.
- Para cada sinal lógico $\int$ ocorrendo em D, $w(\int) = w'(\int)$.

I e I' *coincidem em uma coleção* Ω de designadores se, para cada $D \in \Omega$, I e I' coincidem em D.

2.23 Definição. (**Coincidência entre** LQC**-interpretações**)
Se I e I' são LQC-interpretações, então as seguintes proposições são válidas:
- Se I e I' coincidem em t, então $I_D(t) = I'_D(t)$.
- Se I e I' coincidem em P, então $I_V(P) = I'_V(P)$.

---
[27] Uma função de $\{v, f\}$ para $\{v, f\}$, que a v associa f e a f associa v.
[28] A dois valores booleanos v, f associa um valor booleano v, f.
[29] Valor mínimo de b_1, b_2.
[30] Valor máximo de b_1, b_2.

- Se I e I' coincidem em P, então I satisfaz P sss I' satisfaz P.
- Se I e I' coincidem em Γ, então I satisfaz Γ se I' satisfaz Γ.

2.24 Lema. ***(Instanciação de variáveis por termos fechados em um termo ou uma fórmula)***

- *Se I é uma* LQC*-interpretação para $\{u, t\}$, então $I_D u(x|t) = I(x|I_D(t))_D(u)$.*
- *Se I é uma* LQC*-interpretação para $\{P, t\}$, então $I_V P(x|t) = I(x|I_D(t))_V(P)$.*

2.25 Definição. (**Satisfabilidade de fórmula e de coleção de fórmulas em** LQC)

Sejam $\begin{cases} I \text{ uma LQC-interpretação,} \\ P \text{ uma } I\text{-fórmula,} \\ \Gamma \text{ uma } I\text{-coleção de fórmulas.} \end{cases}$

I satisfaz uma fórmula P em LQC, ou P é LQC-*satisfatível*, se $I_V(P)$ é um valor distinguido em LQC, ou seja, $I_V(P) = \mathrm{v}$.

I satisfaz uma coleção Γ de fórmulas em LQC, ou Γ é LQC-*satisfatível*, se I satisfaz cada fórmula de Γ em LQC.

Caso contrário, P é LQC-insatisfatível e Γ é LQC-insatisfatível.

2.26 Definição. (**Validade de fórmula e de coleção de fórmulas em** LQC)

(i) *I valida Γ em* LQC se cada fórmula de Γ é uma I-fórmula e se existe uma fórmula de Γ tal que I satisfaz essa fórmula.

(ii) P é LQC-*válido* se P é uma fórmula em LQC e toda LQC-interpretação para P satisfaz P, caso contrário, P é dito ser LQC-inválido.

(iii) Γ é LQC-*válido* se toda LQC-interpretação para Γ valida Γ, caso contrário Γ é dito ser LQC-inválido.

2.27 Definição. (LQC**-Consequência Semântica**)

P é consequência semântica de Γ em LQC, $\Gamma \models_{\mathrm{LQC}} P$, se toda LQC-interpretação para $\Gamma \cup \{P\}$ que satisfaz Γ também satisfaz P.

2.28 Leituras. $\Gamma \models_{\mathrm{LQC}} P$

- ‘De Γ afirma-se P em LQC’,
- ‘Γ, portanto P em LQC’,
- ‘P é consequência semântica de Γ em LQC’,
- ‘P é teorema de Γ em LQC’,
- ‘ Γ acarreta semanticamente P em LQC’.

2.29 Exemplo. Quando Γ acarreta P semanticamente em LQC:

- $\{\forall x P, \forall x(P \to Q)\} \models_{LQC} \forall x Q$.
- $\{\forall x P, \exists x Q\} \models_{LQC} P \wedge Q$.

§3. Um Cálculo de Sequentes para LQC

Nesta seção nós falaremos somente da Lógica Quantificacional Clássica. Assim, para dizer que P é consequência de Γ em LQC, notaremos isto por $\Gamma \vdash P$.

Damos abaixo um cálculo de sequentes para LQC. Este se constitui de todas as leis primitivas de LPC, traduzidas para a linguagem de LQC, mais as *Leis de Introdução e Eliminação de Quantificadores*.

Todas as *leis derivadas de* LPC citadas na parte expositiva do capítulo anterior, devidamente traduzidas para a linguagem de LQC, também são válidas em LQC, com exceção do *Esquema da Substituição para Equivalência*, dado na página 193, o qual não é válido nesta forma em LQC, por supor implicitamente um fato não necessariamente presente em LQC, devido à sua distinta estrutura sintática.

Leis de Introdução e Eliminação de Quantificadores

3.1. **Generalização** Se $\begin{cases} \Gamma \vdash P, \\ x \text{ não é livre em } \Gamma, \end{cases}$ então $\Gamma \vdash \forall x P$.

3.2. **$\forall$-Eliminação** $\forall x P \vdash P(x|t)$.

3.3 Escólio. ***(Exemplar do esquema $\forall$-el: $\forall x P \vdash P$)***

1	$\forall x P$	*pr*
2	$P(x\|x)$	*1, $\forall$-el*
3	P	*2*

3.4. **$\exists$-Introdução** $P(x|t) \vdash \exists x P$.

3.5 Escólio. ***(Exemplar do esquema $\exists$-int: $P \vdash \exists x P$)***

1	P	*pr*
2	$P(x\|x)$	*1*
3	$\exists x P$	*2, $\exists$-int*

3.6. **$\exists$-Eliminação**

Se $\begin{cases} \Gamma \vdash \exists x P, \\ \Gamma, P(x|y) \vdash Q, \\ y \text{ não é livre em } \Gamma \cup \{\exists x P, Q\}, \end{cases}$ então $\Gamma \vdash Q$.[31]

[31]Se x não for livre em $\Gamma \cup \{Q\}$ (obviamente, x já não é livre em $\exists x P$), então a variável y

3.7. **$\exists$-Eliminação (versão que usa uma nova constante)**

Se $\begin{cases} \Gamma \vdash \exists x P, \\ \Gamma, P(x|c) \vdash Q, \\ c \text{ não figura em } \Gamma,\ P,\ Q, \end{cases}$ então $\Gamma \vdash Q$.

3.8 Notação. Considere que, no restante deste trabalho, TNI[32] é uma axiomática[33] para os números inteiros.

A seguir damos um exemplo mostrando por que a restrição de x não ser livre em Γ, na regra da generalização, é importante.

3.9 Exemplo. Temos que $\mathrm{TNI}, x = 5 \vdash$ primo(x), e daí, aplicando Gen sem atender à restrição de que x não deve ser livre em Γ, temos que $\mathrm{TNI}, x = 5 \vdash \forall x$ primo(x), e daí $\mathrm{TNI} \vdash x = 5 \to \forall x$ primo(x). Aplicando a este último sequente sucessivamente a Gen e o esquema do $\forall$-eliminação, temos que $\mathrm{TNI} \vdash 5 = 5 \to \forall x$ primo(x), e daí $\mathrm{TNI} \vdash \forall x$ primo(x), o que evidentemente não é verdadeiro.

3.10 Exemplo. Mostre que $\vdash_{\mathrm{TNI}} \forall x$ par$(x + x)$.

Prova:

1	$x + x = 2x$	ant
2	$par(2x)$	ant
3	$par(x + x)$	2, 1
4	$\forall x$ par$(x + x)$	3, Gen

□

O seguinte exemplo mostra a importância da restrição de que y não deve ser livre em $\exists x P$ na regra do $\exists$-eliminação. Nele, essa restrição é violada e as demais são respeitadas.

3.11 Exemplo. Considere que o universo de discurso implícito é a coleção dos números inteiros. Temos que $\mathrm{TNI} \vdash \exists x(x > y)$. Se aplicarmos a Regra do $\exists$-Eliminação sem atender à restrição de que y não deve ser livre em $\exists x P$, temos que $\exists x(x > y) \vdash y > y$, o que acarreta, pela Regra do $\exists$-Introdução, em $\exists x(x > y) \vdash \exists y(y > y)$, e daí $\mathrm{TNI} \vdash \exists y(y > y)$, o que não é verdadeiro.

---

a ser utilizada pode ser a própria variável x. Adiante é dada uma versão simplificada desta regra levando isto em conta.

[32] 'TNI' é uma sigla para 'teoria dos números inteiros'.

[33] Isto é, uma coleção de fórmulas descrevendo as propriedades básicas dos números inteiros.

Existe uma forma mais simples da lei do $\exists$-eliminação comumente utilizada:

3.12 Corolário. *Se* $\begin{cases} \Gamma \models \exists x P, \\ \Gamma, P \models Q, \\ x \text{ não é livre em } \Gamma \cup \{Q\}, \end{cases}$ *então* $\Gamma \models Q$.

Prova:

1	$\Gamma \models \exists x P$	hip
2	$\Gamma, P \models Q$	hip
3	x não é livre em Γ, Q	hip
4	$\Gamma, P(x\|x) \models Q$	2
5	x não é livre em $\exists x P$	ant
6	x não é livre em $\Gamma, \exists x P, Q$	3, 5
7	$\Gamma \models Q$	1, 4, 6, $\exists$-el

$\square$

O próximo exemplo mostra a importância da restrição de x não ser livre em Γ nesta segunda versão da regra do $\exists$-eliminação. Esta é violada, mas a outra restrição é respeitada.

3.13 Exemplo. Temos que TNI $\models$ $\exists x$ ímpar(x), e daí TNI, $par(x) \models \exists x$ ímpar(x). Aplicando a versão simplificada da regra do $\exists$-eliminação sem atender à restrição de que x não deve ser livre em Γ, acabamos concluindo que TNI $\models$ $\exists x(par(x)\wedge$ ímpar(x)), o que não é correto.

No exemplo seguinte mostramos a importância específica da restrição de x ser livre em Q, na formulação desta segunda versão da regra do $\exists$-eliminação.

3.14 Exemplo. Temos que TNI $\models \exists x$ perfeito(x).[34] Aplicando esta segunda versão da regra do $\exists$-eliminação sem atender à restrição de que x não deve ser livre em Q, temos que $\exists x$ perfeito$(x) \models$ perfeito(x), e daí, aplicando a regra da generalização, temos que $\exists x$ perfeito$(x) \models \forall$ perfeito(x), o que nos leva a concluir que TNI $\models \forall x$ perfeito(x), o que obviamente é falso.

[34]Um número inteiro positivo é dito perfeito se a soma dos seus divisores positivos, distintos dele próprio, for igual a este número. Por exemplo, 6 e 28 são ambos números perfeitos.

Outro contra-exemplo ou falácia é apresentado a seguir.

Mostre que $\vdash_{\text{TNI}} \exists y(y > y)$.

Prova:

1	$y + 1 > y$	ant
2	$(x > y)(x\|y + 1)$	1
3	$\exists x(x > y)$	2, $\exists$-int
4	$\quad\vert\ (x > y)(x\|y)$	sup
5	$\quad\vert\ y > y$	4
6	$\quad\vert\ \exists y(y > y)$	5, $\exists$-int
7	$\exists y(y > y)$	3, 4, 6, $\exists$-el

$\square$

Novamente aplicando a segunda versão da Regra do $\exists$-Eliminação e sem verificar que y é livre em $\exists x(x > y)$, temos que $\vdash_{\text{TNI}} \exists y(y > y)$, o que é incorreto.

Leis Básicas dos Quantificadores

3.15. **Negação de Fórmula Existencial** $\vdash \neg\exists xP \leftrightarrow \forall x\neg P$.

Prova da ida:

1	$\vert\ \neg\exists xP$	sup
2	$\vert\ \vert\ P$	sup
3	$\vert\ \vert\ \exists xP$	2,$\exists$-int
4	$\vert\ \vert\ \neg\exists xP$	1
5	$\vert\ \neg P$	2,3,4,$\neg$-int
6	$\vert\ \forall x\neg P$	5,Gen
7	$\neg\exists xP \to \forall x\neg P$	1,6,RD

$\square$

Prova da volta:

1	$\vert\ \forall x\neg P$	sup
2	$\vert\ \neg P$	1,$\forall$-el
3	$\vert\ \vert\ \exists xP$	sup
4	$\vert\ \vert\ \vert\ P$	sup
5	$\vert\ \vert\ \vert\ \neg P$	2
6	$\vert\ \vert\ \vert\ \neg\exists xP$	4,5,NC
7	$\vert\ \vert\ \neg\exists xP$	3,4,6,$\exists$-el
8	$\vert\ \neg\exists xP$	3,7,$\neg$-int
9	$\forall x\neg P \to \neg\exists xP$	1,8,RD

$\square$

3.16 Exemplo.

$$\begin{gathered}\text{Não existe algo imortal.}\\ \downarrow \\ \neg\exists x \text{ imortal}(x) \\ \updownarrow \\ \forall x \neg \text{ imortal}(x) \\ \downarrow \\ \text{Tudo não é mortal.}\end{gathered}$$

3.17 Exemplo.

$$\begin{gathered}\text{Não existe Doutor iletrado.}\\ \downarrow \\ \neg\exists x(\text{ Doutor}(x) \wedge \text{iletrado}(x)) \\ \updownarrow \\ \forall x \neg(\text{Doutor}(x) \wedge \text{ iletrado}(x)) \\ \updownarrow \\ \forall x(\text{Doutor}(x) \to \neg\text{iletrado}(x)) \\ \downarrow \\ \text{Todo Doutor não é iletrado.}\end{gathered}$$

3.18. **Negação de Fórmula Universal** $\vdash \neg\forall x P \leftrightarrow \exists x \neg P$.

Prova da ida:

1	$\neg\exists x \neg P$	sup
2	$\forall x \neg\neg P$	1,NE
3	$\neg\neg P$	2,$\forall$-el
4	P	3,DN
5	$\forall x P$	4,Gen
6	$\neg\neg\forall x P$	5,DN
7	$\neg\exists x \neg P \to \neg\neg\forall x P$	1,6,RD
8	$\neg\forall x P \to \exists x \neg P$	7, Ctp

□

Prova da volta:

1	$\neg\neg\forall x P$	sup
2	$\forall x P$	1,DN
3	P	2,$\forall$-el
4	$\neg\neg P$	3,DN
5	$\forall x \neg\neg P$	4,Gen
6	$\neg\exists x \neg P$	5,NE
7	$\neg\neg\forall x P \to \neg\exists\neg P$	1,6,RD
8	$\exists x \neg P \to \neg\forall x P$	7, CTP

□

3.19 Exemplo.

$$\begin{array}{c}\text{Nem todo leão é feroz.}\\ \downarrow \\ \neg\forall x(\text{leão}(x) \rightarrow \text{ feroz}(x)) \\ \updownarrow \\ \exists x\neg(\text{leão}(x) \rightarrow \text{ feroz}(x)) \\ \updownarrow \\ \exists x(\text{leão}(x) \wedge \neg \text{ feroz}(x)) \\ \downarrow \\ \text{Algum leão não é feroz.}\end{array}$$

3.20. **Instanciação**[35] Se $\begin{cases}\Gamma \vdash P, \\ x \text{ não é livre em } \Gamma,\end{cases}$ então $\Gamma \vdash P(x|t)$.

3.21. **Vacuidade**[36] Se x não é livre em P, então $\begin{cases}\text{(i)} \ \vdash \forall xP \leftrightarrow P, \\ \text{(ii)} \ \vdash \exists xP \leftrightarrow P.\end{cases}$

Prova de (i):

1	x não é livre em P	hip
2	$\forall xP \rightarrow P$	$\forall$-el
3	$\mid$ P	sup
4	$\mid$ $\forall xP$	3,1, Gen
5	$P \rightarrow \forall xP$	3,4,RD
6	$\forall xP \leftrightarrow P$	2,5,$\wedge$-int

$\square$

Prova de (ii):

1	x não é livre em P	hip
2	$P \rightarrow \exists xP$	$\exists$-int
3	$\mid$ $\exists xP$	sup
4	$\mid$ $\mid$ P	sup
5	$\mid$ $\mid$ P	4
6	$\mid$ P	3,4,5,1,$\exists$-el
7	$\exists xP \rightarrow P$	3,6,RD
8	$\exists xP \leftrightarrow P$	7,2,$\wedge$- int

$\square$

[35] É uma combinação entre a Regra da Generalização e a Regra do $\forall$-eliminação.
[36] É uma condição de uma ocorrência de quantificador ser vácua.

3.22 Corolário. ***(Lei de Negação de Fórmula Existencial e Negação de Fórmula Universal)***

(i) $\vdash \forall xP \leftrightarrow \exists x\neg P$,
(ii) $\vdash \exists xP \leftrightarrow \neg\forall x\neg P$.

Prova de (i):

1	$\neg\forall xP \leftrightarrow \exists x\neg P$	NU
2	$\neg\neg\forall xP \leftrightarrow \neg\exists x\neg P$	1, LSC
3	$\forall xP \leftrightarrow \neg\neg\forall xP$	DN
4	$\forall xP \leftrightarrow \neg\exists x\neg P$	3, 2, TE

□

Prova de (ii):

1	$\neg\exists xP \leftrightarrow \forall x\neg P$	NE
2	$\neg\neg\exists xP \leftrightarrow \neg\forall x\neg P$	1, LSC
3	$\exists xP \leftrightarrow \neg\neg\exists xP$	DN
4	$\exists xP \leftrightarrow \neg\forall x\neg P$	3, 2, TE

□

3.23. **Congruência**

Se y não é livre em $\forall xP$, então $\begin{cases} \text{(i)} \vdash \exists xP \leftrightarrow \exists yP(x|y), \\ \text{(ii)} \vdash \forall xP \leftrightarrow \forall yP(x|y). \end{cases}$

3.24 Exemplo.

Todo número primo maior e igual a 3 é impar.
$\forall x(\text{primo}(x) \wedge x \geq 3 \rightarrow \text{ ímpar}(x))$. (1)
$\forall y(\text{primo}(y) \wedge y \geq 3 \rightarrow \text{ ímpar}(y))$. (2)

Com essa lei é possível renomear a variável quantificada sem alterar o significado da fórmula[37]. Isso pode ser útil em algum processamento no qual seja necessário renomear variáveis, por exemplo, as leis de substituição.

3.25. **Lema da Substituição para Quantificadores**
$\begin{cases} \text{(i)} \ \forall x(P \leftrightarrow Q) \vdash \forall xP \leftrightarrow \forall xQ, \\ \text{(ii)} \ \forall x(P \leftrightarrow Q) \vdash \exists xP \leftrightarrow \exists xQ. \end{cases}$

Leis Complementares dos Quantificadores

3.26. **Comutatividade** $\begin{cases} \text{(i)} \vdash \forall x\forall yP \leftrightarrow \forall y\forall xP, \\ \text{(ii)} \vdash \exists x\exists yP \leftrightarrow \exists y\exists xP. \end{cases}$

---

[37](1) e (2) não são iguais, mas são congruentes.

3.27. $\exists$**-Importação**[38] $\vdash \exists x \forall y P \rightarrow \forall y \exists x P$.

Prova:

1	$\mid\ \exists x \forall y P$	sup
2	$\mid\ \mid\ \forall y P$	sup
3	$\mid\ \mid\ P$	2,$\forall$-el
4	$\mid\ \mid\ \exists x P$	3,$\exists$-int
5	$\mid\ \mid\ \forall y \exists x P$	4,Gen
6	$\mid\ \forall y \exists x P$	1,2,5,$\exists$-el
7	$\exists x \forall y P \rightarrow \forall y \exists x P$	1,6,RD

□

Para mostrar que a volta não é correta, tentaremos provar que $\vdash \forall y \exists x P \rightarrow \exists x \forall y P$.

Prova:

1	$\mid\ \forall y \exists x P$	sup
2	$\mid\ \mid\ \exists x P$	1,$\forall$-el
3	$\mid\ \mid\ P$	sup
4	$\mid\ \mid\ \forall y P x$	

□

A regra da generalização não pode ser aplicada em (4) pois é impossível saber se y não é livre em Γ, ou seja, nas premissas existentes no ambiente de hipótese: y não é livre em (1) mas pode ser livre em P. Dessa forma, não é verdade que $\vdash \forall y \exists x P \rightarrow \exists x \forall y P$.

3.28 Exemplo.
Sejam $\Delta = \mathbb{N}$ e $w(<) = \{\langle x, y\rangle | x, y \in \mathbb{N} \wedge x < y\}$.
$I_V(\forall y \exists x(y < x)) =$ v, mas $I_V(\exists x \forall y(y < x)) =$ f.

3.29 Definição.
Sejam Ψ um quantificador e $\#$ um conectivo diádico.
Dizemos que:

(i) Ψ distribui com respeito a $\#$ em $\mathcal{L}$ se $\Psi x(P\#Q) \vdash_{\mathcal{L}} \Psi xP\#\Psi xQ$,

(ii) Ψ fatora com respeito a $\#$ em $\mathcal{L}$ se $\Psi xP\#\Psi xQ \vdash_{\mathcal{L}} \Psi x(P\#Q)$.

[38] Ou $\forall$-Exportação.

3.30. **Distributividade e Fatorabilidade de Quantificadores**
Sejam Ψ um quantificador, $\mathcal{L}$ uma lógica e $\#$ um conectivo diádico em $\mathcal{L}$:

- Ψ é distributivo (em $\mathcal{L}$) em relação a $\#$: $\Psi x(P\#Q) \vdash_{\mathcal{L}} \Psi xP\#\Psi xQ$.
- Ψ é fatorativo (em $\mathcal{L}$) com respeito a $\#$: $\Psi x(P\#Q) \vdash_{\mathcal{L}} \Psi x(P\#Q)$.

(i) $\vdash \forall x(P \to Q) \to (\forall xP \to \forall xQ)$,
(ii) $\vdash \forall x(P \land Q) \leftrightarrow (\forall xP \land \forall xQ)$,
(iii) $\vdash \forall xP \lor \forall xQ \to \forall x(P \lor Q)$,
(iv) $\vdash \forall x(P \leftrightarrow Q) \to (\forall xP \leftrightarrow \forall xQ)$,
(v) $\vdash (\exists xP \to \exists xQ) \to \exists x(P \to Q)$,
(vi) $\vdash \exists x(P \land Q) \to \exists xP \land \exists xQ$,
(vii) $\vdash \exists x(P \lor Q) \leftrightarrow \exists xP \lor \exists xQ$.

No item (iii), a volta não é correta, ou seja, não é verdade que $\vdash \forall x(P \lor Q) \to \forall xP \lor \forall xQ$.
Veja este contra-exemplo:

3.31 Contra-Exemplo. $I_V(\forall x(\leq 0 \lor x > 0)) = \text{v}$

$$\left.\begin{array}{l} I_V(\forall x(x \leq 0)) = \text{f} \\ I_V(\forall x(x > 0)) = \text{f} \end{array}\right\} I_V(\forall x(x \leq 0) \lor \forall x(x > 0)) = \text{f}$$

No item (iv), o quantificador universal não fatora com respeito à equivalência, ou seja, não é verdade que $\vdash (\forall xP \leftrightarrow \forall xQ) \to \forall x(P \leftrightarrow Q)$.
Veja o seguinte contra-exemplo:

3.32 Contra-Exemplo. $\vdash_{\text{TNI}} \neg\forall x\ \text{par}(x)$.
$\neg\forall x\ \text{primo}(x)$.
$\forall x\ \text{par}(x) \leftrightarrow \forall x\ \text{primo}(x)$.
$\neg\forall x(\text{par}(x) \leftrightarrow \text{primo}(x))$.

A lei de substituição da equivalência para LQC será enunciada aqui, mas detalhada e provada em § 10.1.

3.33. **Regra da Substituição para Equivalência**

$$\text{Se} \begin{cases} \Gamma \vdash P_1 \leftrightarrow P_2, \\ S \text{ não está, em } Q\text{, no escopo de nenhuma variável livre em } \Gamma \text{ e em } \{P_1, P_2\}, \end{cases}$$

então $\Gamma \vdash Q(S\|P_1) \leftrightarrow Q(S\|P_2)$.

3.34. **Distributividade e Fatorabilidade Degeneradas de Quantificadores**

$$\text{Sejam} \begin{cases} \Psi, \Psi' \text{ quantificadores,} \\ \mathcal{L} \text{ uma lógica,} \\ \# \text{ conectivos diádicos em } \mathcal{L}. \end{cases}$$

Ψ distribui de forma degenerada (em $\mathcal{L}$) com respeito a $\#$ se e somente se:

- existe $\Psi' \neq \Psi$ tal que $\Psi x(P\#Q) \vdash_{\mathcal{L}} \Psi xP\#\Psi' xQ$, ou

- existe $\Psi' \neq \Psi$ tal que $\Psi x(P\#Q) \models_{\mathcal{L}} \Psi' xP\#\Psi xQ$, ou
- existem $\Psi', \Psi'' \neq \Psi$ tal que $\Psi x(P\#Q) \models_{\mathcal{L}} \Psi' xP\#\Psi'' xQ$.

Ψ fatora de forma degenerada (em $\mathcal{L}$) com respeito a $\#$ se e somente se:

- existe $\Psi' \neq \Psi$ tal que $\Psi' xP\#\Psi xQ \models_{\mathcal{L}} \Psi x(P\#Q)$, ou
- existe $\Psi' \neq \Psi$ tal que $\Psi xP \neq \Psi' xQ \models_{\mathcal{L}} \Psi x(P\#Q)$, ou
- existem $\Psi', \Psi'' \neq \Psi$ tal que $\Psi' xP\#\Psi'' xQ \models_{\mathcal{L}} \Psi x(P\#Q)$.

(i) $\models \forall x(P \to Q) \to (\exists xP \to \exists xQ)$,
(ii) $\models (\exists xP \to \forall xQ) \to \forall x(P \to Q)$,
(iii) $\models \forall x(P \vee Q) \to \forall xP \vee \exists xQ$,
(iv) $\models \forall x(P \vee Q) \to \exists xP \vee \forall xQ$,
(v) $\models \forall x(P \leftrightarrow Q) \to (\exists xP \leftrightarrow \exists xQ)$,
(vi) $\models \exists x(P \to Q) \leftrightarrow \forall xP \to \exists xQ$,
(vii) $\models \forall xP \wedge \exists xQ \to \exists x(P \wedge Q)$,
(viii) $\models \exists xP \wedge \forall xQ \to \exists x(P \wedge Q)$,
(ix) $\models (\forall xP \leftrightarrow \exists xQ) \to \exists x(P \leftrightarrow Q)$,
(x) $\models (\exists xP \leftrightarrow \forall xQ) \to \exists x(P \leftrightarrow Q)$.

3.35 Contra-Exemplo.

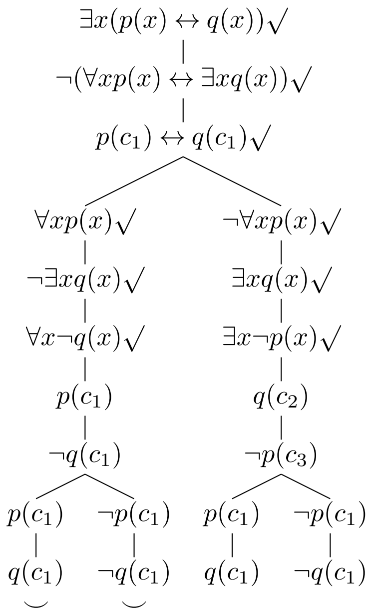
No item (ix), a volta não é correta, ou seja, não é verdade que $\models \exists x(P \leftrightarrow Q) \to (\forall xP \leftrightarrow \exists xQ)$.

Sejam $\Delta = \{c_1, c_2, c_3\}$, $w(p) = \{c_1\}$ e $w(q) = \{c_1, c_2\}$.

$$\left.\begin{array}{l} I_V(p(c_1) \leftrightarrow q(c_1)) = \text{v} \\ I_V(\exists x(p(x) \leftrightarrow q(x))) = \text{v} \end{array}\right\} I_V(\exists x(P \leftrightarrow Q)) = \text{v}.$$

$$\left.\begin{array}{l} I_V(\forall xp(x)) = \text{f} \\ I_V(\exists xq(x)) = \text{v} \\ I_V(\forall xp(x) \leftrightarrow \exists xq(x)) = \text{f} \end{array}\right\} I_V(\forall xP \leftrightarrow \exists xQ) = \text{f}.$$

Desta forma $I_V(\exists x(P \leftrightarrow Q) \to (\forall xP \leftrightarrow \exists xQ)) = \text{f}$.



3.36. **Transporte de Quantificadores**.

Todas as leis abaixo são indispensáveis para se fazer uma prova por indução com redução à forma prenex. Além disso, a redução a forma prenex é essencial para diversos métodos de automatização do raciocínio. Redução à forma conjuntiva normal, redução à forma disjuntiva normal e redução à forma prenex, são ferramentas indispensáveis para a automatização do raciocínio, além da lei da vacuidade (eliminação de quantificadores vácuos). Métodos de resolução sempre fazem redução à forma prenex e os métodos de tablô também costumam fazê-la.

- Se x não é livre em P, então $\begin{cases} \text{(i)} & \vdash \forall x(P \to Q) \leftrightarrow (P \to \forall x Q), \\ \text{(ii)} & \vdash \exists x(P \to Q) \leftrightarrow (P \to \exists x Q), \\ \text{(iii)} & \vdash \forall x(P \wedge Q) \leftrightarrow P \wedge \forall x Q, \\ \text{(iv)} & \vdash \exists x(P \wedge Q) \leftrightarrow P \wedge \exists x Q, \\ \text{(v)} & \vdash \forall x(P \vee Q) \leftrightarrow P \vee \forall x Q, \\ \text{(vi)} & \vdash \exists x(P \vee Q) \leftrightarrow P \vee \exists x Q. \end{cases}$

Prova de (i):

x não é livre em P	(1), hip
$\forall x(P \to Q)$	
$\leftrightarrow$	IM, SE
$\forall x(\neg P \lor Q)$	
$\leftrightarrow$	1, TQ(v)
$\neg P \lor \forall x Q$	
$\leftrightarrow$	IM
$P \to \forall x Q$	(8)
$\forall x(P \to Q) \leftrightarrow P \to \forall x Q$	1, 8, TE

Prova de (iii):

1	x não é livre em P	hip
2	$\forall x(P \land Q) \leftrightarrow \forall x P \land \forall x Q$	DFQ
3	$\forall x(P \land Q) \leftrightarrow P \land \forall x Q$	1,Vc,SE

Prova de (v):

1	x não é livre em P	hip
2	$\quad\vert\ \forall x(P \lor Q)$	sup
3	$\quad\vert\ \exists x P \lor \forall x Q$	2, DFDQ
4	$\quad\vert\ P \lor \forall x Q$	3, 1, Vc, SE
5	$\forall x(P \lor Q) \to P \lor \forall x Q$	2, 4, RD
6	$\quad\vert\ P \lor \forall x Q$	sup
7	$\quad\vert\ \forall x P \lor \forall x Q$	6, 1, Vc, SE
8	$\quad\vert\ \forall x(P \lor Q)$	7, DFQ
9	$P \lor \forall x Q \to \forall x(P \lor Q)$	6, 8, RD
10	$\forall x(P \lor Q) \leftrightarrow P \lor \forall x Q$	5, 9, $\land$-int

Abstração das leis de i a vi:
$\vdash \Psi x(P \# Q) \leftrightarrow P \# \Psi x Q$, onde $\Psi \in \{\forall, \exists\}$ e $\# \in \{\rightarrow, \wedge, \vee\}$.

Nas seguintes leis, os quantificadores mudam ou se invertem se são transportados para o antecedente da implicação (casos i e ii). Nos outros casos (transporte para o consequente da implicação, conjunção e disjunção) eles permanecem os mesmos .

- Se x não é livre em Q, então $\begin{cases} \text{(i)} & \vdash \forall x(P \rightarrow Q) \leftrightarrow (\exists x P \rightarrow Q), \\ \text{(ii)} & \vdash \exists x(P \rightarrow Q) \leftrightarrow (\forall x P \rightarrow Q), \\ \text{(iii)} & \vdash \forall x(P \wedge Q) \leftrightarrow \forall x P \wedge Q, \\ \text{(iv)} & \vdash \exists x(P \wedge Q) \leftrightarrow \exists x P \wedge Q, \\ \text{(v)} & \vdash \forall x(P \vee Q) \leftrightarrow \forall x P \vee Q, \\ \text{(vi)} & \vdash \exists x(P \vee Q) \leftrightarrow \exists x P \vee Q. \end{cases}$

3.37. **$\exists$-Exportação**
Se $\begin{cases} x \text{ não é livre em } Q, \\ y \text{ não é livre em } P, \end{cases}$ ou $\begin{cases} x \text{ não é livre em } P, \\ y \text{ não é livre em } Q, \end{cases}$
então
(i) $\vdash \forall x \exists y(P \rightarrow Q) \leftrightarrow \exists y \forall x(P \rightarrow Q)$;
(ii) $\vdash \forall x \exists y(P \wedge Q) \leftrightarrow \exists y \forall x(P \wedge Q)$;
(iii) $\vdash \forall x \exists y(P \vee Q) \leftrightarrow \exists y \forall x(P \vee Q)$.

Quantificadores Típicos

Embora definimos fórmulas universais e existenciais em LQC para quaisquer corpos, o seu maior uso prático se dá com implicações e conjunções, como corpos de fórmulas universais e de fórmulas existenciais, respectivamente.

Uma fórmula universal com um corpo qualquer pode significar algo bem improvável de ser verdadeiro. Por exemplo, considere a fórmula $\forall x$ efêmero(x). Pelo significado que normalmente atribuímos a ser efêmero, temos que esta fórmula é falsa, mas existem várias restrições possíveis para o intervalo de valores da variável x que poderiam torná-la verdadeira. Poderíamos, por exemplo, restringir os valores de x para a classe das borboletas, e expressar facilmente esta restrição escrevendo $\forall x(borboleta(x) \rightarrow$ efêmero$(x))$.

De uma forma análoga, uma fórmula existencial com um corpo qualquer pode significar algo tão genérico que é muito improvável de ser falso, assim a informação contida em tal fórmula é em geral desprovida de utilidade. Por exemplo, considere a fórmula $\exists x$ homem(x). Tal informação é tão genérica e óbvia que normalmente não possui utilidade prática. Restringindo o domínio de valores possíveis da variável x, poderíamos expressar algo bem mais interessante. Poderíamos por exemplo limitar os valores de x para a classe dos centenários, e expressar com facilidade tal restrição escrevendo $\exists x(\text{centenário}(x) \wedge \text{homem}(x))$.

Para um estudo sistemático de tais fórmulas quantificadas, utilizamos as abreviaturas dadas a seguir.

3.38 Definição. Adotamos as seguintes abreviaturas para fórmulas universais e existenciais cujos corpos são respectivamente implicações e conjunções:
- $\forall RxP \rightleftharpoons \forall x(R \to P)$;
- $\exists RxP \rightleftharpoons \exists x(R \wedge P)$.

As expressões '$\forall R$' e '$\exists R$' são chamadas de *quantificadores típicos*, ou melhor, *quantificadores universais típicos* e *quantificadores existenciais típicos*, respectivamente. As fórmulas $\forall RxP$ e $\exists RxP$ são também chamadas de *universais típicas* e *existenciais típicas*, onde R é sua *restrição* e P é seu *corpo*.

Dizer $\forall RxP$ significa afirmar que cada objeto x que satisfaz R satisfaz P, isto é, a propriedade universal $\forall xP$ torna-se restrita aos objetos x do *tipo* ou do *gênero* que satisfazem R. Da mesma forma, dizer $\exists RxP$ significa afirmar que existe um objeto que satisfaz R e P, isto é, a propriedade existencial $\exists xP$ torna-se restrita aos objetos x do *tipo* ou do *gênero* que satisfazem R.

Podemos generalizar esses conceitos de quantificação típica e definir uma convenção para o sinal predicativo diádico e o sinal predicativo monádico.

3.39 Convenção. (**Para o sinal predicativo diádico**)
Seja pt um sinal predicativo diádico.
- $\forall\, x\, pt\, P \rightleftharpoons \forall(x\, pt)x\, P \rightleftharpoons \forall x(x\, pt \to P)$,
- $\exists\, x\, pt\, P \rightleftharpoons \exists(x\, pt)x\, P \rightleftharpoons \exists x(x\, pt \wedge P)$.

3.40 Exemplo.
a) $\forall x > 0(2x > 0)$, onde $>$ é o sinal predicativo p, 0 é o termo e $2x > 0$ é o corpo P.
b) $\exists x > 0(x^2 < 100)$, onde $>$ é o sinal predicativo p, 0 é o termo e $x^2 < 100$ é o corpo P.

3.41 Convenção. (**Para o sinal predicativo monádico**)
Seja p um sinal predicativo monádico.
- $\forall\, x\, p\, P \rightleftharpoons \forall p(x)x\, P \rightleftharpoons \forall x(p(x) \to P)$,
- $\exists\, x\, p\, P \rightleftharpoons \exists p(x)x\, P \rightleftharpoons \exists x(p(x) \wedge P)$.

3.42 Exemplo.
a)

$$\begin{gathered}\forall x \text{ positivo}(x^2 > 0)\\ \rightleftharpoons\\ \forall \text{ positivo}(x)x(x^2 > 0)\\ \rightleftharpoons\\ \forall x(\text{positivo}(x) \to (x^2 > 0))\end{gathered}$$

b)

$$\begin{gathered}\exists x \text{ político honesto}(x)\\ \rightleftharpoons\\ \exists \text{ político}(x)x \text{ honesto}(x)\\ \rightleftharpoons\\ \exists x(\text{ político}(x) \wedge \text{ honesto}(x))\end{gathered}$$

c) $\exists x$ real$(x^2 = 2) \rightleftharpoons \exists$ real$(x)x(x^2 = 2) \rightleftharpoons \exists x($ real$(x) \wedge (x^2 = 2))$.

Empreendemos a seguir um estudo mais detalhado destes quantificadores. Conforme veremos, muitas das propriedades dos quantificadores típicos assemelham-se bastante às dos quantificadores comuns, sendo que a única diferença está em eventuais restrições adicionais.

3.43. **Regra da Generalização Típico**

Se $\begin{cases}\Gamma, R \vdash P,\\ x \text{ não é livre em } \Gamma, \text{ então } \Gamma \vdash \forall RxP.\end{cases}$

Prova:

1	$\Gamma, R \vdash P$	hip
2	x não é livre em Γ	hip
3	$\Gamma \vdash R \to P$	1, RD
4	$\Gamma \vdash \forall x(R \to P)$	3, Gen
5	$\Gamma \vdash \forall RxP$	4, def

□

3.44. **Esquema do $\forall$-eliminação Típico** $\forall RxP, R(x|t) \vdash P(x|t)$.

Prova:

1	$\forall RxP$	pr
2	$R(x|t)$	pr
3	$\forall x(R \to P)$	1, def
4	$R(x|t) \to P(x|t)$	3, $\forall$-el
5	$P(x|t)$	2, 4, MP

$\square$

3.45 Exemplo.
$\forall$ homem$(x)x$ mortal, homem(Sócrates) $\vdash$ mortal(Sócrates).

3.46 Escólio. *São equivalentes:*
(i) *y é livre em $\forall xP$,*
(ii) *y é livre em $\exists xP$.*

3.47. **Esquema do $\exists$-introdução Típico** $R(x|t), P(x|t) \vdash \exists RxP$.

Prova:

1	$R(x|t)$	pr
2	$P(x|t)$	pr
3	$R(x|t) \wedge P(x|t)$	1, 2, $\wedge$-int
4	$(R \wedge P)(x|t)$	3
5	$\exists x(R \wedge P)$	4, $\exists$-int
6	$\exists RxP$	5, def

$\square$

3.48 Exemplo.
árvore(mangueira), frondosa(mangueira) $\vdash \exists$ árvore $(x)x$ frondosa(x).

3.49. **Regra do $\exists$-eliminação Típico**

Se $\begin{cases} \Gamma \vdash \exists RxP, \\ \Gamma, R(x|t), P(x|t) \vdash Q, \\ y \text{ não é livre em } \Gamma, \exists xR, \exists xP, Q, \end{cases}$ então $\Gamma \vdash Q$.

Prova:

1	$\Gamma \vdash \exists RxP$	hip
2	$\Gamma, R(x\|y), P(x\|y) \vdash Q$	hip
3	y não é livre em Γ, $\exists xR$, $\exists xP$, Q	hip
4	$\Gamma \vdash \exists x(R \wedge P)$	1, def
5	para todo $S \in \Gamma, \Gamma \vdash S$	Ref
6	$(R \wedge P)(x\|y) \vdash R(x\|y)$	$\wedge$-el
7	$(R \wedge P)(x\|y) \vdash P(x\|y)$	$\wedge$-el
8	$\Gamma, (R \wedge P)(x\|y) \vdash R(x\|y)$	6, Mon
9	$\Gamma, (R \wedge P)(x\|y) \vdash P(x\|y)$	7, Mon
10	para todo $S \in \Gamma$, Γ, $(R \wedge P)(x\|y) \vdash S$	5, Mon
11	$\Gamma, (R \wedge P)(x\|y) \vdash Q$	10, 9, 8, 2, TG
12	y não é livre em $\Gamma, \exists x(R \wedge P)$, Q	3
13	$\Gamma \vdash Q$	4, 11, 12, $\exists$-el

$\square$

3.50. **Regra do $\exists$-eliminação Típico (forma simplificada)**

Se $\begin{cases} \Gamma \vdash \exists RxP, \\ \Gamma, R, P \vdash Q, \\ x \text{ não é livre em } \Gamma, Q, \end{cases}$ então $\Gamma \vdash Q$.

Prova:

1	$\Gamma \vdash \exists RxP$	hip
2	$\Gamma, R, P \vdash Q$	hip
3	x não é livre em Γ, Q	hip
4	$R(x\|x) = R$	ant
5	$P(x\|x) = P$	ant
6	$\Gamma, R(x\|x), P(x\|x) \vdash Q$	2, 4, 5
7	x não é livre em $\exists xR, \exists xP$	ant
8	x não é livre em $\Gamma, \exists xR, \exists xP, Q$	3, 7
9	$\Gamma \vdash Q$	1, 6, 8

$\square$

3.51 Lema. $\vdash (R \vee S \to P) \leftrightarrow (R \to P) \wedge (S \to P)$.

3.52. **Bifurcação** $\begin{cases} \text{(i)} \vdash \forall(R \vee S)xP \leftrightarrow \forall RxP \wedge \forall SxP, \\ \text{(ii)} \vdash \exists(R \vee S)xP \leftrightarrow \exists RxP \vee \exists SxP. \end{cases}$

Prova de (i):

1	$\forall(R \vee S)xP \leftrightarrow \forall x(R \vee S \to P)$	def
2	$\forall(R \vee S)xP \leftrightarrow \forall x((R \to P) \wedge (S \to P))$	1, SE, Lema 3.51
3	$\forall(R \vee S)xP \leftrightarrow \forall x(R \to P) \wedge \forall x(S \to P)$	2, DFQ, SE
4	$\forall(R \vee S)xP \leftrightarrow \forall RxP \wedge \forall SxP$	3, def

$\square$

3.53 Exemplo.

$$\begin{array}{c}
\text{Todo ser humano é mortal.} \\
\downarrow \\
\forall x(\text{humano}(x) \rightarrow \text{ mortal}(x)) \\
\leftrightarrow \\
\forall \text{ humano}(x) x \text{ mortal}(x) \\
\leftrightarrow \\
\forall(\text{ homem}(x) \vee \text{ mulher}(x)) x \text{ mortal}(x) \\
\leftrightarrow \\
\forall \text{ homem}(x) x \text{ mortal}(x) \wedge \forall \text{ mulher}(x) x \text{ mortal}(x) \\
\downarrow \\
\text{Todo homem é mortal e toda mulher é mortal.}
\end{array}$$

3.54 Exemplo.

$$\begin{array}{c}
\vdash \exists(\text{leão}(x) \vee \text{ hiena}(x)) x \text{ manso}(x) \\
\leftrightarrow \\
\exists \text{ leão}(x) x \text{ manso}(x) \vee \exists \text{ hiena}(x) x \text{ manso}(x)
\end{array}$$

Generalizando a bifurcação, temos o seguinte:

- $\vdash \forall(R_1 \vee \ldots \vee R_n)xP \leftrightarrow \forall R_1 xP \wedge \ldots \wedge \forall R_n xP$,
- $\vdash \exists(R_1 \vee \ldots \vee R_n)xP \leftrightarrow \exists R_1 xP \vee \ldots \vee \exists R_n xP$.

Outro resultado específico de quantificadores típicos está na sua capacidade de retração ou de desdobramento. Antes de enunciá-lo, precisamos de uma notação correspondente.

3.55 Notação.
Adotamos as seguintes abreviaturas concernentes à quantificação típica envolvendo duas variáveis simultaneamente:

- $\forall RxyP \rightleftharpoons \forall x \forall y(R \rightarrow P)$,
- $\exists RxyP \rightleftharpoons \exists x \exists y(R \wedge P)$.

ou, de forma semelhante temos que:

- $\forall Rx_1, \ldots, x_n P \rightleftharpoons \forall x_1 \ldots \forall x_n(R \rightarrow P)$,
- $\exists Rx_1, \ldots, x_n P \rightleftharpoons \exists x_1 \ldots \exists x_n(R \wedge P)$.

3.56. **Aninhamento** Se y não é livre em R, então

$$\begin{cases} \text{(i)} \vdash \forall(R \wedge S)x, yP \leftrightarrow \forall Rx \forall SyP, \\ \text{(ii)} \vdash \exists(R \wedge S)x, yP \leftrightarrow \exists Rx \exists SyP. \end{cases}$$

3.57. **Aninhamento Generalizado**
Considerando que, para quaisquer $i, j \in \{1, \ldots, n\}$, se $i > j$, então x_i não é livre em R_j, os seguintes esquemas, concernentes ao *aninhamento de quantificadores típicos*, são válidos:

- $\vdash \forall(R_1 \wedge \ldots \wedge R_n)x_1, \ldots, x_nP \leftrightarrow \forall R_1x_1 \ldots \forall R_nx_nP$,
- $\vdash \exists(R_1 \wedge \ldots \wedge R_n)x_1, \ldots, x_nP \leftrightarrow \exists R_1x_1 \ldots \exists R_nx_nP$.

3.58. **Negação de Fórmula Universal Típica** $\vdash \neg\forall RxP \leftrightarrow \exists Rx\neg P$.

Prova:

1	$\neg\forall RxP \leftrightarrow \neg\forall x(R \to P$	def
2	$\neg\forall RxP \leftrightarrow \exists x\neg(R \to P)$	1, NU, SE
3	$\neg\forall RxP \leftrightarrow \exists x(R \wedge \neg P)$	2, NI, SE
4	$\neg\forall RxP \leftrightarrow \exists Rx\neg P$	3, def

□

3.59. **Negação de Fórmula Existencial Típica** $\vdash \neg\exists RxP \leftrightarrow \forall Rx\neg P$.

Prova:

$\neg\exists RxP$	(1)
$\leftrightarrow$	def
$\neg\exists x(R \wedge P)$	
$\leftrightarrow$	NE
$\forall x\neg(R \wedge P)$	
$\leftrightarrow$	NCj
$\forall x(R \to \neg P)$	
$\leftrightarrow$	def
$\forall Rx\neg P$	
$\neg\exists RxP \leftrightarrow \forall Rx\neg P$	1, TE

□

3.60. **Congruência Típica**

- Se y não é livre em $\{R, P\}$, então
$$\begin{cases} \text{(i)} \vdash \exists RxP \leftrightarrow \exists R(x|y)yP(x|y), \\ \text{(ii)} \vdash \forall RxP \leftrightarrow \forall R(x|y)yP(x|y). \end{cases}$$

3.61. **Comutatividade Típica**

- Se $\begin{cases} x \text{ não é livre em } S, \\ y \text{ não é livre em } R, \end{cases}$ então $\begin{cases} \text{(i)} & \vdash \forall Rx\forall SyP \leftrightarrow \forall Sy\forall RxP, \\ \text{(ii)} & \vdash \exists Rx\exists SyP \leftrightarrow \exists Sy\exists RxP. \end{cases}$

3.62. **$\exists$-Importação Típica**

Se $\begin{cases} x \text{ não é livre em } S, \\ y \text{ não é livre em } R, \end{cases}$ então $\vdash \exists Rx\forall SyP \to \forall Sy\exists RxP.$

Prova de (i):

1	x não é livre em S	hip
2	y não é livre em R	hip
3	$\mid\ \exists Rx\forall SyP$	sup
4	$\mid\ \mid\ R, \forall SyP$	sup
5	$\mid\ \mid\ \mid\ S$	sup
6	$\mid\ \mid\ \mid\ R$	4
7	$\mid\ \mid\ \mid\ P$	4, 5, $\forall$-elt
8	$\mid\ \mid\ \mid\ \exists RxP$	6, 7, $\exists$-intt
9	$\mid\ \mid\ \forall Sy\exists RxP$	5, 8, GenT,2
10	$\mid\ \forall Sy\exists RxP$	3, 4, 9, 1, $\exists$-elt
11	$\exists Rx\forall SyP \to \forall Sy\exists RxP$	3, 10, RD

$\square$

Outra prova do $\exists$-Importação Típica:

3.63 Lema. $\vdash (R \wedge (S \to P)) \to (S \to (R \wedge P))$.

Prova:

$$
\begin{array}{l}
R \wedge (S \to P) \\
R \\
S \to P \\
\quad \left| \begin{array}{l} \underline{S} \\ P \\ R \\ R \wedge P \end{array} \right. \\
S \to R \wedge P
\end{array}
$$

□

3.64 Escólio.

- $\vdash \forall xP \leftrightarrow \forall \top xP$,
- $\vdash \exists xP \leftrightarrow \exists \top xP$.

3.65. **Distributividade e Fatorabilidade de Quantificadores Típicos**

(i) $\vdash \forall Rx(P \to Q) \to (\forall RxP \to \forall RxQ)$,
(ii) $\vdash \forall Rx(P \wedge Q) \leftrightarrow \forall RxP \wedge \forall RxQ$,
(iii) $\vdash \forall RxP \vee \forall RxQ \to \forall Rx(P \vee Q)$,
(iv) $\vdash \forall Rx(P \leftrightarrow Q) \to (\forall RxP \leftrightarrow \forall RxQ)$,
(v) $\exists xR \vdash (\exists RxP \to \exists RxQ) \to \exists Rx(P \to Q)$,
(vi) $\vdash \exists Rx(P \wedge Q) \to \exists RxP \wedge \exists RxQ$,
(vii) $\vdash \exists Rx(P \vee Q) \leftrightarrow \exists RxP \vee \exists RxQ$.

3.66. **Distributividade e Fatorabilidade Degeneradas de Quantificadores Típicos**

(i) $\vdash \forall Rx(P \to Q) \to (\exists RxP \to \exists RxQ)$,
(ii) $\vdash (\exists RxP \to \forall RxQ) \to \forall Rx(P \to Q)$,
(iii) $\vdash \forall Rx(P \vee Q) \to \forall RxP \vee \exists RxQ$,
(iv) $\vdash \forall Rx(P \vee Q) \to \exists RxP \vee \forall RxQ$,
(v) $\vdash \forall Rx(P \leftrightarrow Q) \to (\exists RxP \leftrightarrow \exists RxQ)$,
(vi) $\vdash \forall RxP \wedge \exists RxQ \to \exists Rx(P \wedge Q)$,
(vii) $\vdash \exists RxP \wedge \forall RxQ \to \exists Rx(P \wedge Q)$,

(viii) $\vdash \exists Rx(P \to Q) \leftrightarrow \forall RxP \to \exists RxQ$,
(ix) $\vdash (\forall RxP \leftrightarrow \exists RxQ) \to \exists Rx(P \leftrightarrow Q)$,
(x) $\vdash (\exists RxP \leftrightarrow \forall RxQ) \to \exists Rx(P \leftrightarrow Q)$.

3.67. **Transporte de Quantificadores Típicos**

- Se x não é livre em P, então $\begin{cases} \text{(i)} & \vdash \forall Rx(P \to Q) \leftrightarrow (P \to \forall RxQ), \\ \text{(ii)} & \exists xR \vdash \exists Rx(P \to Q) \leftrightarrow (P \to \exists RxQ), \\ \text{(iii)} & \exists xR \vdash \forall Rx(P \wedge Q) \leftrightarrow P \wedge \forall RxQ, \\ \text{(iv)} & \vdash \exists Rx(P \wedge Q) \leftrightarrow P \wedge \exists RxQ, \\ \text{(v)} & \vdash \forall Rx(P \vee Q) \leftrightarrow P \vee \forall RxQ, \\ \text{(vi)} & \exists xR \vdash \exists Rx(P \vee Q) \leftrightarrow P \vee \exists RxQ. \end{cases}$
- Se x não é livre em Q, então $\begin{cases} \text{(i)} & \vdash \forall Rx(P \to Q) \leftrightarrow (\exists RxP \to Q), \\ \text{(ii)} & \exists xR \vdash \exists Rx(P \to Q) \leftrightarrow (\forall RxP \to Q), \\ \text{(iii)} & \exists xR \vdash \forall Rx(P \wedge Q) \leftrightarrow \forall RxP \wedge Q, \\ \text{(iv)} & \vdash \exists Rx(P \wedge Q) \leftrightarrow \exists RxP \wedge Q, \\ \text{(v)} & \vdash \forall Rx(P \vee Q) \leftrightarrow \forall RxP \vee Q, \\ \text{(vi)} & \exists xR \vdash \exists Rx(P \vee Q) \leftrightarrow \exists RxP \vee Q. \end{cases}$

3.68. **$\exists$-Exportação**

Considere que $\begin{cases} x \text{ não é livre em } S, \\ y \text{ não é livre em } R. \end{cases}$

Se $\begin{cases} x \text{ não é livre em } Q, \\ y \text{ não é livre em } P, \end{cases}$ ou $\begin{cases} x \text{ não é livre em } P, \\ y \text{ não é livre em } Q, \end{cases}$

então

(i) $\exists yS \vdash \forall Rx\exists Sy(P \to Q) \leftrightarrow \exists Sy\forall Rx(P \to Q)$,
(ii) $\exists xR \vdash \forall Rx\exists Sy(P \wedge Q) \leftrightarrow \exists Sy\forall Rx(P \wedge Q)$,
(iii) $\exists yS \vdash \forall Rx\exists Sy(P \vee Q) \leftrightarrow \exists Sy\forall Rx(P \vee Q)$.

Capítulo 6

Indução Matemática

Este capítulo discorre sobre a Indução Matemática de forma sucinta, por se tratar de uma ferramenta indispensável na demonstração de diversos resultados lógicos presentes neste trabalho, tais como a correção, a completude e outros desenlaces apresentados nos capítulos que tratam da LPC e da LQC com indução[1].

Ressaltamos que a indução aqui abordada é um tipo de raciocínio dedutivo[2] e não se relaciona com a indução normalmente utilizada nas ciências empíricas[3].

As três formas básicas de indução matemática consideradas nesta exposição são: Indução em Sistemas de Peano, Indução Transfinita e Indução Estrutural.

§1. Conceitos Gerais

Em diversas aplicações de Indução Matemática deve-se provar uma propriedade $\Delta(n)$ valendo para todos os números naturais n, ou, em alguns casos, para todos os naturais a partir de um determinado número natural, o qual, em geral, é o número 1 ou o número 2.

---

[1] Sobre os tipos usuais de raciocínio, os fundamentos da indução e detalhes avançados sobre indução matemática o leitor pode consultar as referências: [5–7, 11, 43–46].

[2] Um raciocínio dedutivo parte de informações universais para extrair uma conclusão particular.

[3] O conceito de indução utilizado nas ciências empíricas relaciona-se com o raciocínio que nos leva a tirar conclusões gerais, ou a fazer previsões sobre casos não observados, a partir dos casos que já observamos. Por exemplo, considere que um copo de vidro se quebra ao ser arremessado ao chão. Logo, por este tipo de indução, todo copo de vidro se quebra ao ser arremessado ao chão.

1.1 Exemplo.
Seja $\Delta(n) = par(n)$, então temos que $\Delta(0) = par(0), \Delta(1) = par(1)$, e assim sucessivamente.

Apresentamos uma taxonomia das formas de Indução Matemática frequentemente utilizadas pelos matemáticos e abordadas neste trabalho.

1.2 Definição. (**Taxonomia da Indução Matemática**)

$$\text{Indução Matemática}\begin{cases}\text{Indução em Sistemas de Peano}\begin{cases}\text{Fraca}\\ \text{Forte}\end{cases}\\ \text{Indução Transfinita}\begin{cases}\text{Fraca}\\ \text{Forte}\end{cases}\\ \text{Indução Estrutural}\begin{cases}\text{Funcional}\\ \text{Relacional}\end{cases}\end{cases}$$

Algumas considerações sobre a taxonomia:

- A indução em Sistemas de Peano, em geral, é um caso particular da indução transfinita.
  - ∗ A indução fraca em Sistemas de Peano é um caso particular da indução transfinita fraca e da indução estrutural.
  - ∗ A indução forte em Sistemas de Peano é um caso particular da indução transfinita forte.
- A indução transfinita finita fraca[4] é um caso particular da indução estrutural.
- A indução de grau é uma variante da indução forte em Sistemas de Peano.
- Também existe a indução de grau em sistemas transfinitos, a qual é uma variante[5] da indução forte em sistemas transfinitos.
- A indução estrutural e a indução transfinita são generalizações distintas da indução em Sistemas de Peano[6].

A indução forte difere da indução fraca apenas na suposição da hipótese considerada[7]:

---

[4]Que também pode ser chamada *indução segmentária fraca.*

[5]Dizemos *variante* pois ambas são, de uma certa forma, equivalentes.

[6]A partir da indução estrutural você pode gerar um conjunto, seja funcional ou relacionalmente, a partir de um conjunto inicial e usando certas operações. No primeiro caso os elementos geradores são apenas funções, e, no segundo caso, são relações (que não são necessariamente funções, mas podem ser funções). Na indução transfinita a partida é o elemento mínimo da coleção bem ordenada, mas podem existir elementos que não são obtidos de elementos precedentes por funções ou relações, mas que são o supremo de todos os elementos precedentes, ao contrário do que acontece em uma geração estrutural, na qual todos os elementos não iniciais são obtidos por aplicação de uma lista de elementos já obtidos por uma das funções ou relações geradoras.

[7]Alguns autores, no lugar de “fraca”, dizem “simples”, e, no lugar de “forte”, dizem “completa”.

- no caso da indução forte, é preciso mostrar que a propriedade vale para todos os antecessores,
- no caso da indução fraca, basta mostrar que a propriedade vale para o primeiro antecessor.

1.3 Exemplo.
Queremos provar que uma propriedade $\Delta(n)$ vale para o caso em que $n = 8$, ou seja, $\Delta(8)$.
Utilizando a indução forte é preciso mostrar que, se esta propriedade vale para todos os números menores que 8, então ela também vale para o número 8. Através da indução fraca prova-se que, se a propriedade vale para o número 7, então ela também vale para o número 8.

§2. Indução em Sistemas de Peano

Os Sistemas de Peano[8] são muito utilizados para fazer demonstrações sobre conjuntos que são isomorfos aos números naturais. Dessa forma, um exemplo de Sistema de Peano é o próprio conjunto dos números naturais.

2.1 Definição. (Sistema de Peano)
Uma tripla $\langle N, \Theta, s\rangle$ é dita um *Sistema de Peano* se as seguintes condições forem satisfeitas:

- N é uma coleção,
- $\Theta \in N$, onde Θ é dito o *elemento inicial* do Sistema de Peano,
- s é uma função de N em N em que, dado $n \in N$, $s(n)$ é dito o *sucessor* de n no Sistema de Peano,
- $\forall n \in N(\Theta \neq s(n))$,
- $\forall m, n(m, n \in N \wedge s(m) = s(n) \rightarrow m = n)$,
- se $\Delta(n)$ é uma propriedade para elementos $n \in N$, então[9]
  $\begin{cases} \Delta(\Theta), \\ \forall n \in N(\Delta(n) \rightarrow \Delta(s(n)), \end{cases}$ implica que $\forall n \in N, \Delta(n))$.

Mais exemplos de Sistemas de Peano:

2.2 Exemplo.
$\langle \mathbb{N}, 0, s\rangle$ é um Sistema de Peano, onde:

$$\begin{aligned} s\ :&\ \mathbb{N} \rightarrow \mathbb{N} \\ &\ n \mapsto n+1 \end{aligned}$$

[8]Em homenagem ao matemático italiano Giuseppe Peano (1858-1932) que primeiramente expôs a ideia de indução.

[9]Também chamamos esta condição do Sistema de Peano de *Princípio da Indução Fraca*.

2.3 Exemplo.
$\langle \mathbb{N}^*, 1, s \rangle$ é um Sistema de Peano, onde:

$$\begin{aligned} s\ &: \mathbb{N}^* \to \mathbb{N}^* \\ n\ &\mapsto n+1 \end{aligned}$$

2.4 Exemplo.
$\langle \text{pares}, 0, s \rangle$ é um Sistema de Peano, sendo *pares*=$\{n \in \mathbb{N} \mid \text{ par } (n)\}$onde:

$$\begin{aligned} s\ &: pares \to pares \\ n\ &\mapsto n+2 \end{aligned}$$

2.5 Escólio. *Em um Sistema de Peano, o seu primeiro componente é uma cópia isomorfa do conjunto dos números naturais.*

O que comumente é utilizado neste trabalho são os Sistemas de Peano com o domínio sendo o conjunto dos números naturais, com um dos números $0, 1$ ou 2 no papel de elemento inicial. A mais importante alternativa está em adotar como domínio de um Sistema de Peano o conjunto dos números naturais a partir de um certo número natural, em geral, os números 1 ou 2. E a forma sucessora mais adotada é a operação de 'somar 1' ($n \mapsto n+1$).

2.6 Exemplo.
Uma Progressão Aritmética[10] é uma sequência de números na qual a diferença entre dois termos consecutivos é constante. Se a é uma Progressão Aritmética, notamos o seu n-ésimo termo por a_n.
Então, se a uma P.A. e r é a diferença constante, prove que $a_n = a_1 + (n-1) \cdot r$

Prova:
Seja $\Delta(n)$ a propriedade '$a_n = a_1 + (n-1) \cdot r$'.
Temos que $a_1 = a_1 + (1-1).r$, logo vale $\Delta(1)$. (1)
Suponha, por HI, que, dado $n \in \mathbb{N}^*$, $a_n = a_1 + (n-1).r.(\Delta(n))$.
$a_{n+1} = a_1 + (n-1).r + r$.
$a_{n+1} = a_1 + n.r = a_1 + ((n+1)-1).r$.
Ou seja, $a_{n+1} = a_1 + ((n+1)-1).r$, logo vale $\Delta(n+1)$.
Provamos daí que $\forall n \in \mathbb{N}^*(\Delta(n) \to \Delta(n+1))$. (2)
Portanto, de (1) e (2)), $\forall n \in \mathbb{N}^*$, $a_n = a_1 + (n-1).r$. $\square$

Em um Sistema de Peano são válidos o princípio da indução fraca[11], o princípio da indução forte e o princípio da boa ordem.

Considere que s é uma função, ou seja, s também é uma relação. Toda relação tem um fecho transitivo, definido em 2.3.52, que é a menor relação que contém s e é transitiva.

[10] Normalmente abreviada por P.A.
[11] Descrita no sexto item das condições para um Sistema de Peano, em 2.1.

2.7. **Princípio da Indução Forte em Sistemas de Peano**[12]

Sejam $\begin{cases} \langle N, \Theta, s\rangle \text{ um Sistema de Peano,} \\ \text{`$<$' o fecho transitivo de } s, \\ \Delta(n) \text{ uma propriedade para elementos } n \in N. \end{cases}$

Então, $\forall n \in N(\forall m(m < n \rightarrow \Delta(m)) \rightarrow \Delta(n))$ implica que $\forall n \in N$, $\Delta(n)$.

2.8 Exemplo.

Mostre que $\models_{\text{TNN}} \forall n(n \geq 2 \rightarrow n$ possui fatoração em primos).

Prova:

Seja $\Delta(n)$ a propriedade '$n \geq 2 \rightarrow n$ é fatorável em primos'.

Obviamente valem $\Delta(0)$ e $\Delta(1)$.

Suponha que $n \geq 2$ e considere, por HI, que, para cada $m < n, \Delta(m)$.

Temos que, primo$(n) \vee \neg$ primo(n).

Se primo$(n), n$ é um produto com um único fator, que é ele mesmo, daí não há mais o que provar.

Se $\neg$primo(n), então existem $m_1, m_2 \in \mathbb{N}$, onde $m_1, m_2 \notin \{1, n\}$, tal que $n = m_1.m_2$.

Temos que $m_1 < n$ e $m_2 < n$.

Daí, por HI, m_1 e m_2 são fatoráveis em primos, portanto n é fatorável em primos.

Logo, por PIF, $\forall n(n \geq 2 \rightarrow n$ é fatorável em primos). $\square$

2.9. **Princípio da Boa Ordem em Sistemas de Peano**

Sejam $\begin{cases} \langle N, \Theta, s\rangle \text{ um Sistema de Peano,} \\ \text{`$<$' o fecho transitivo de } s, \end{cases}$

então $\forall A(A \subseteq N$ e $A \neq \varnothing$ implica que A possui um $<$-elemento mínimo).

2.10 Definição. (**Conjunção Generalizada**)

- $\wedge() \rightleftharpoons \top$ (conjunção de uma lista vazia de fórmulas)[13].
- $\wedge(P_1) \rightleftharpoons P_1$.
- $\wedge(P_1, \ldots, P_n, P_{n+1}) \rightleftharpoons \wedge(P_1, \ldots, P_n) \wedge P_{n+1} (n \geq 1)$.

2.11 Exemplo.

Seja $\Delta(n)$ a propriedade 'para quaisquer fórmulas $P_1, \ldots, P_n$, temos que $P_1, \ldots, P_n \vdash P_1 \wedge \ldots \wedge P_n$'. Então temos que:

- $\Delta(0)$: $\varnothing \vdash \top$,
- $\Delta(1)$: para qualquer fórmula P, temos que $P \vdash P$,
- $\Delta(2)$: para quaisquer fórmulas P_1, P_2, temos que $P_1, P_2 \vdash P_1 \wedge P_2$,

[12] Considere 'PIF' como sendo a abreviação de **P**rincípio da **I**ndução **F**orte.

[13] Uma conjunção é verdadeira quando todos os seus conjuntores são verdadeiros. No caso em que a lista de fórmulas é vazia, a falsidade de sua conjunção implicaria na existência de um conjuntor da mesma, o que é absurdo.

- $\Delta(3)$: para quaisquer fórmulas P_1, P_2, P_3, temos que $P_1, P_2, P_3 \vdash P_1 \wedge P_2 \wedge P_3$,
- $\Delta(n)$: para quaisquer fórmulas $P_1, \ldots, P_n$, temos que $P_1, \ldots, P_n \vdash P_1 \wedge \ldots \wedge P_n$.

2.12 Escólio. *Se I é uma* LPC*-interpretação para $P_1, \ldots, P_n$, então:*
- $I_V(\wedge()) = I_V(\top) = \mathrm{v}$.
- *Para* $n \geq 1, I_V(\wedge(P_1, \ldots, P_n)) = \mathrm{v}$ *sss* $I_V(P_1) = \mathrm{v}$ *e* $\ldots$ *e* $I_V(P_n) = \mathrm{v}$.

2.13 Definição. (**Disjunção Generalizada**)
- $\vee() \rightleftharpoons \bot$ (disjunção de uma lista vazia de fórmulas)[14].
- $\vee(P_1) \rightleftharpoons P_1$.
- $\vee(P_1, \ldots, P_n, P_{n+1}) \rightleftharpoons \vee(P_1, \ldots, P_n) \vee P_{n+1} (n \geq 1)$.

2.14 Escólio. *Se I é uma* LPC*-interpretação para $P_1, \ldots, P_n$, então:*
- $I_V(\vee()) = I_V(\bot) = \mathrm{f}$.
- *Para* $n \geq 1, I_V(\vee(P_1, \ldots, P_n)) = \mathrm{v}$ *sss* $I_V(P_1) = \mathrm{v}$ *ou* $\ldots$ *ou* $I_V(P_n) = \mathrm{v}$.

2.15 Exemplo.
Prove a proposição abaixo, que generaliza o esquema do $\wedge$-introdução:
$P_1, \ldots, P_n \vdash P_1 \wedge \ldots \wedge P_n$.

Prova:
Seja $\Delta(n)$ a proposição '$P_1, \ldots, P_n \vdash \wedge(P_1, \ldots, P_n)$'.
Temos que $\vdash \top$, daí $\vdash \wedge()$, ou seja, vale $\Delta(0)$. (1)
Como $\wedge(P_1) = P_1$, temos que, por Ref, $\wedge(P_1) \vdash P_1$, ou seja, vale $\Delta(1)$.
Dado $n \in \mathbb{N}$ tal que $n \geq 1$, suponha, por hipótese de indução, que vale $\Delta(n)$, ou seja, $P_1, \ldots, P_n \vdash \wedge(P_1, \ldots, P_n)$. (2)
Para cada $i \in \{1, \ldots, n\}$, $P_1, \ldots, P_n, P_{n+1} \vdash P_i$. (3)
De (3), (2) e Tran, temos que $P_1, \ldots, P_n, P_{n+1} \vdash \wedge(P_1, \ldots, P_n)$. (4)
Temos também, por Ref, que $P_1, \ldots, P_n, P_{n+1} \vdash P_{n+1}$. (5)
Pelo $\wedge$-int temos que $\wedge(P_1, \ldots, P_n), P_{n+1} \vdash \wedge(P_1, \ldots, P_n) \wedge P_{n+1}$.
Ou seja, $\wedge(P_1, \ldots, P_n), P_{n+1} \vdash \wedge(P_1, \ldots, P_n, P_{n+1})$. (6)
De (4), (5), (6) e Tran temos que $P_1, \ldots, P_n, P_{n+1} \vdash \wedge(P_1, \ldots, P_n, P_{n+1})$, ou seja, vale $\Delta(n+1)$. $\square$

[14] Uma disjunção é verdadeira quando existe pelo menos um disjuntor da lista que é verdadeiro. No casa em que a lista de fórmulas é vazia, a veracidade de sua disjunção implicaria na existência de um disjuntor da mesma, o que é absurdo.

2.16 Exemplo.
Prove as proposições abaixo, as quais generalizam a lei da distributividade:
(i) $P \wedge (Q_1 \vee \ldots \vee Q_n) \leftrightarrow (P \wedge Q_1) \vee \ldots \vee (P \wedge Q_n)$.
(ii) $P \vee (Q_1 \wedge \ldots \wedge Q_n) \leftrightarrow (P \vee Q_1) \wedge \ldots \wedge (P \vee Q_n)$.

Prova de (i):
Seja $\Delta(n)$ a propriedade $\vdash P \wedge \vee(Q_1, \ldots, Q_n) \leftrightarrow \vee(P \wedge Q_1, \ldots, P \wedge Q_n)$.
Por resultado anterior, temos que $\vdash P \wedge \vee() \leftrightarrow \top$, daí $\vdash P \wedge \vee() \leftrightarrow \vee()$, ou seja, vale $\Delta(0)$. (1)
Temos que $\vdash P \wedge \vee(Q_1) \leftrightarrow P \wedge Q_1 \leftrightarrow \vee(P \wedge Q_1)$, logo $\vdash P \wedge \vee(Q_1) \leftrightarrow \vee(P \wedge Q_1)$, ou seja, vale (1). (2)
Dado $n \in \mathbb{N}^*$, suponha $\Delta(n)$, ou seja
$\vdash P \wedge \vee(Q_1, \ldots, Q_n) \leftrightarrow \vee(P \wedge Q_1, \ldots, P \wedge Q_n)$. (3)
$\vdash P \wedge \vee(Q_1, \ldots, Q_n, Q_{n+1}) \leftrightarrow P \wedge (\vee(Q_1, \ldots, Q_n) \vee Q_{n+1})$. (4)
De (7), $\vdash P \wedge \vee(Q_1, \ldots, Q_n, Q_{n+1}) \leftrightarrow (P \wedge \vee(Q_1, \ldots, Q_n)) \vee (P \wedge Q_{n+1})$. (5)
De (5), (3), $\vdash P \wedge \vee(Q_1, \ldots, Q_n, Q_{n+1}) \leftrightarrow \vee(P \wedge Q_1, \ldots, P \wedge Q_n) \vee (P \wedge Q_{n+1})$. (6)
De (6), $\vdash P \wedge \vee(Q_1, \ldots, Q_n, Q_{n+1}) \leftrightarrow \vee(P \wedge Q_1, \ldots, P \wedge Q_n, P \wedge Q_{n+1})$, ou seja, vale $\Delta(n+1)$.
Logo, pelo PIF, temos que para cada $n \in \mathbb{N}^*$, $\Delta(n)$. (7)
De (1) e (7) concluímos que para cada $n \in \mathbb{N}$, $\Delta(n)$. □

Prova de (ii): É análoga à prova de (i). □

2.1 Indução de Grau em Sistemas de Peano

É um caso particular da indução forte em Sistemas de Peano, pois é provada a partir do princípio da indução forte[15].

O que puder ser mensurado em termos de números naturais é um tipo de indução de grau em Sistemas de Peano, tais como a indução sobre o grau de fórmula e a indução sobre o grau de termo. Por exemplo, quando realiza-se a indução sobre o grau de fórmula, a mensuração não é sobre a fórmula, mas sobre o seu grau.

Uma definição específica sobre o grau de uma fórmula é apresentada em 9.1.4, para LPC, e em 10.1.3, para LQC.

2.17 Comentário. Na indução transfinita, a indução de grau (ou a mensuração) é feita sobre um conjunto cujos elementos são associados a números ordinais segundo uma correspondência biunívoca.

[15]Neste trabalho a indução de grau é utilizada, por exemplo, na prova da lei da congruência.

2.18 Definição. (**Sistema de Peano expandido**)
$\langle N, \Theta, s, < \rangle$ é um Sistema de Peano expandido se:
- $\langle N, \Theta, s \rangle$ é um Sistema de Peano,
- '$<$' é o fecho transitivo de s.

2.19. **Princípio da Indução de Grau em Sistemas de Peano**

$$\text{Sejam} \begin{cases} \langle N, \Theta, s, < \rangle \text{ um sistema de Peano expandido,} \\ A \text{ uma coleção qualquer,} \\ \mathrm{gr} : A \to N, \\ \Delta(n) \text{ uma propriedade para elementos } n \in A. \end{cases}$$

Se $\forall n \in A(\forall m \in A(gr(m) < gr(n) \to \Delta(m)) \to \Delta(n))$, então $\forall n \in A$, $\Delta(n)$.

Prova: Prova-se a partir do princípio da indução forte em Sistemas de Peano. □

2.20 Exemplo. Se L é a coleção de fórmulas em LPC, e gr é a função que associa a cada fórmula de L o número de ocorrências de conectivos desta fórmula, então, neste caso, o Sistema de Peano expandido é $\langle \mathbb{N}, 0, suc, < \rangle$, onde

$$\begin{aligned} suc \ : \mathbb{N} &\to \mathbb{N} \\ n &\mapsto n + 1 \end{aligned}$$

e $<$ é a relação de ordem estrita usual dos naturais. Daí o Princípio da Indução de Grau em Sistemas de Peano formula-se da seguinte forma, onde $\Delta(P)$ é uma propriedade para fórmulas P em LPC: para qualquer fórmula P em LPC, se a validade de $\Delta(Q)$ para qualquer fórmula Q de grau menor que P implica na validade de $\Delta(P)$, então, para qualquer fórmula P em LPC, vale $\Delta(P)$.

§3. Indução Transfinita

A indução transfinita generaliza a noção usual de indução matemática para conjuntos bem ordenados que podem ser finitos, infinitos enumeráveis ou infinitos não enumeráveis.

3.1 Definição. (**Sistema Transfinito**)
$\langle \mathcal{O}, \Theta, \prec, s \rangle$ é um sistema transfinito se:
(i) $\mathcal{O}$ é uma coleção,
(ii) Θ é o $\prec$-mínimo de $\mathcal{O}$,
(iii) $\prec$ é uma boa ordem estrita em $\mathcal{O}$,
(iv) s é a sucessão induzida por $<$.

Alguns exemplos de sistemas transfinitos:

3.2 Exemplos.

- $\langle\{n \in \mathbb{N} | n \geq 1\}, 1, \prec, s\rangle$, onde

$$\begin{aligned} s \ :& \{n \in \mathbb{N} | n \geq 1\} \rightarrow \{n \in \mathbb{N} | n \geq 1\} \\ n & \qquad\qquad\quad \mapsto n+1 \end{aligned}$$

- $\langle\{n \in \mathbb{N} | n \geq 2\}, 2, \prec, s\rangle$, onde

$$\begin{aligned} s \ :& \{n \in \mathbb{N} | n \geq 2\} \rightarrow \{n \in \mathbb{N} | n \geq 2\} \\ n & \qquad\qquad\quad \mapsto n+2 \end{aligned}$$

- $\langle$ naturais pares $, 0, \prec, s\rangle$, onde

$$\begin{aligned} s \ :& \text{ naturais pares } \rightarrow \text{ naturais pares} \\ n & \qquad\qquad\quad \mapsto n+2 \end{aligned}$$

3.3. **Princípio da Indução Transfinita Forte**[16]

Sejam $\begin{cases} \langle \mathcal{O}, \prec\rangle \text{ uma coleção estritamente bem ordenada,} \\ \Delta(n) \text{ uma propriedade para elementos } n \in \mathcal{O}. \end{cases}$

Se $\forall n \in \mathcal{O}(\forall m(m < n \rightarrow \Delta(m)) \rightarrow \Delta(n))$, então $\forall n \in \mathcal{O}$, vale $\Delta(n)$.

3.4 Definição. (Princípio da Indução Transfinita Fraca)

Sejam $\begin{cases} \langle \mathcal{O}, \Theta, <, s\rangle \text{ um sistema transfinito,} \\ \Delta(n) \text{ uma propriedade para elementos } n \in \mathcal{O}. \end{cases}$

Se $\begin{cases} \Delta(\Theta), \\ \forall n \in \mathcal{O}((n \text{ possui } <\text{-sucessor } \wedge \Delta(n)) \rightarrow \Delta(s(n))), \\ \forall n \in \lim\limits_{\prec} \mathcal{O}(\forall m(m < n \rightarrow \Delta(m)) \rightarrow \Delta(n)), \end{cases}$

então $\forall n \in \mathcal{O}$, vale $\Delta(n)$.

Conforme o princípio acima, temos que:

- A primeira condição diz que Θ deve satisfazer a propriedade, ou seja, deve valer $\Delta(\Theta)$.
- A segunda condição diz que, para todo elemento pertencente à coleção $\mathcal{O}$, possuindo um $<$-sucessor, se a propriedade vale para o mesmo, então esta propriedade vale para o seu sucessor.
- A terceira condição diz que, se para um dado $\prec$-elemento limite de $\mathcal{O}$, se a propriedade Δ vale para todos os seus $\prec$-minorantes em $\mathcal{O}$, então Δ vale também para este dado elemento.

[16] O objetivo deste princípio é provar que a propriedade vale para todos os elementos de $\mathcal{O}$.

§4. Indução Estrutural

A indução estrutural é útil quando precisamos demonstrar uma propriedade sobre estruturas definidas recursivamente.

Podemos citar dois tipos de indução estrutural: a *funcional* e a *relacional*. A *indução sobre fórmulas* é um exemplo de indução estrutural funcional e a *indução sobre sequentes corretos no cálculo* é um exemplo de indução estrutural relacional.

Estes tipos de indução se baseiam nos conceitos definidos a seguir, os quais chamamos de *geração*.

4.1 Definição. (Geração Funcional)
Uma tripla $G = \langle U, B, \mathcal{F} \rangle$ é dita uma geração funcional se as seguintes condições forem satisfeitas:
- U e B são coleções;
- $B \subseteq U$;
- $\mathcal{F}$ é uma coleção de funções tais que, para cada $f \in \mathcal{F}$, o domínio de f é uma coleção de tuplas em U e a imagem de f é um subconjunto de U.

Neste caso:
- U é dito o ambiente de G,
- cada elemento de B é dito um elemento inicial de G,
- cada elemento de $\mathcal{F}$ é dito uma função geradora em G.

4.2 Exemplo.
Ambiente: conjunto dos números reais $\mathbb{R}$. A partir de $\mathbb{R}$ é gerado o conjunto dos números naturais $\mathbb{N}$[17].
Coleção de elementos iniciais: o conjunto cujo único elemento é o zero: $\{0\}$.
Função Geradora: a operação sucessor, que é a função definida abaixo:

$$\begin{aligned} s\ :&\mathbb{R} \to \mathbb{R} \\ &x \mapsto x+1 \end{aligned}$$

4.3 Exemplo.
Ambiente: a coleção de samblagens em LPC.
Coleção de elementos iniciais: a coleção de letras sentenciais $p_1, p_2, p_3, \ldots$
Funções Geradores: as funções que a cada samblagem P ou a cada par de samblagens $\langle P, Q \rangle$ associam as samblagens $\neg P$, $P \to Q$, $P \wedge Q$ e $P \vee Q$. As mesmas são denominadas, respectivamente, $\xi_\neg, \xi_\to, \xi_\wedge$ e $\xi_\vee$.

4.4 Exemplo.
Considere como ambiente a coleção de samblagens em LQC. Neste caso, o

[17] Logo a seguir será definido o conjunto gerado em uma geração.

conjunto inicial é a coleção de fórmulas atômicas ($p(t_1, \ldots, t_n)$) e as funções geradoras são $\xi_\neg, \xi_\rightarrow, \xi_\wedge, \xi_\vee$ e, para cada variável x, $\xi_{\forall x}, \xi_{\exists x}$. Neste caso, a coleção gerada é a coleção de fórmulas em LQC.

4.5 Definição. (**Geração (Relacional)**)
Uma tripla $G = \langle U, B, \mathcal{R} \rangle$ é dita uma geração relacional se as seguintes condições forem satisfeitas:
- U e B são coleções;
- $B \subseteq U$;
- $\mathcal{R}$ é uma coleção de relações poliádicas em U.

Neste caso:
- U é dito o ambiente de G,
- cada elemento de B é dito um elemento inicial de G,
- cada elemento de $\mathcal{R}$ é dito uma relação poliádica geradora em G.

4.6 Exemplo.
Ambiente: a coleção de todos os sequentes em LQC, considerando uma dada linguagem.
Coleção de elementos iniciais: a coleção dos exemplares de esquemas do cálculo de sequentes para LQC.
Relações Geradoras: a coleção de regras do cálculo de sequentes para LQC.
Note que as regras abaixo são relações que não são funções[18]:
- Regra da Monotonicidade;
- Regra da Generalização;
- Regra do $\exists$-eliminação.

4.7 Notação. $G = \langle U, B, \mathcal{R} \rangle$ é uma geração.

4.8 Definição. (**n-segmento**)
Dado um número natural n, notamos a coleção dos números naturais positivos menores ou iguais a n por n-segmento. Chamamos tal coleção de *n-segmento*. Se uma dada coleção é o n-segmento, para algum natural n, então também chamamos tal coleção de *segmento*.

4.9 Exemplos.
- 0-segmento $= \{\}$,
- 1-segmento $= \{1\}$,
- 2-segmento $= \{1, 2\}$,
- 3-segmento $= \{1, 2, 3\}$,
- $\ldots$

Para definirmos o que é um conjunto gerado, antes é preciso definir *sequência construtiva*.

[18] As demais regras do cálculo de sequentes para LQC são funções.

4.10 Definição. (**Sequência Construtiva em uma Geração Funcional**)
Sejam $\begin{cases} G = \langle U, B, \mathcal{F} \rangle \text{ uma geração funcional,} \\ I \text{ um segmento.} \end{cases}$

Dizemos que $(x_i)_{i \in I}$ é uma sequência construtiva em G se, para cada $i \in I$, uma das duas seguintes condições é satisfeita:

- $x_i \in B$,
- existem $i_1, \ldots, i_j \in I$ tais que $i_1, \ldots, i_j$ minoram i, e existe $f \in \mathcal{F}$ tal que $\langle x_{i_1}, \ldots, x_{i_j} \rangle \in \mathcal{D}(f)$ e $x_i = f(x_{i_1}, \ldots, x_{i_j})$.

4.11 Exemplo.
Considere o exemplo 4.3.
A sequência construtiva deste exemplo é a sequência de samblagens em LPC, em que o primeiro termo da sequência é uma samblagem inicial em LPC e qualquer termo intermediário (que não seja o primeiro termo), ou é elemento inicial da samblagem em LPC ou é obtido de elementos anteriores da sequência através de uma função geradora.
Considere a sequência construtiva especificada abaixo:
$p_1, p_5, p_7, p_5 \rightarrow p_7, \neg p_5, (p_5 \rightarrow p_7) \wedge \neg p_5$.
O primeiro termo da sequência, p_1, pertence à coleção de elementos iniciais.
O mesmo acontece com p_5 e p_7.
Entretanto, o termo $p_5 \rightarrow p_7$ é obtido de p_5 e de p_7 aplicando a função geradora $\Psi_{\rightarrow}$, o termo $\neg p_5$ é obtido de p_5 aplicando a função geradora $\Psi_{\neg}$, e o termo $(p_5 \rightarrow p_7) \wedge \neg p_5$ é obtido aplicando a função geradora $\Psi_{\wedge}$ entre os novos termos $p_5 \rightarrow p_7$ e $\neg p_5$. Uma sequência é uma função que associa cada segmento a um elemento do conjunto gerado. Para o segmento $\{1, 2, 3, 4, \ldots\}$, ao 1 associa p_1, ao 2 associa p_5, ao 3 associa p_7, ao 4 associa $(p_5 \rightarrow p_7)$, ao 4 associa $\neg p_5$ e ao 5 associa $(p_5 \rightarrow p_7) \wedge \neg p_5$.

4.12 Definição. (**Sequência Construtiva em uma Geração (Relacional)**)
Sejam $\begin{cases} G = \langle U, B, \mathcal{R} \rangle \text{ uma geração (relacional),} \\ I \text{ um segmento.} \end{cases}$

Dizemos que $(x_i)_{i \in I}$ é uma sequência construtiva em G se, para cada $i \in I$, uma das duas seguintes condições é satisfeita:

- $x_i \in B$,
- existem $i_1, \ldots, i_j \in I$ tais que $i_1, \ldots, i_j$ minoram i, e existe $R \in \mathcal{R}$ tal que $\langle x_{i_1}, \ldots, x_{i_j} \rangle \in \mathcal{D}(R)$ e $\langle x_{i_1}, \ldots, x_{i_j} \rangle R\, x_i$.

4.13 Exemplo.
Para exemplificar uma sequência construtiva em uma geração relacional apresentamos a demonstração abaixo, que é uma prova de $((P \to Q) \to P \vdash P$.

Prova de $((P \to Q) \to P \vdash P$:

1	$(P \to Q) \to P \vdash (P \to Q) \to P$	Ref
2	$(P \to Q) \to P, \neg P \vdash \neg P$	Ref
3	$(P \to Q) \to P, \neg P, P \to Q \vdash P \to Q$	Ref
4	$(P \to Q) \to P, \neg P, P \to Q \vdash (P \to Q) \to P$	1,Mon
5	$P \to Q, (P \to Q) \to P \vdash P$	MP
6	$(P \to Q) \to P, \neg P, P \to Q \vdash P$	3,4,5,Tran
7	$(P \to Q) \to P, \neg P, P \to Q \vdash \neg P$	2, Mon
8	$(P \to Q) \to P, \neg P \vdash \neg(P \to Q)$	6,7,$\neg$-int
9	$(P \to Q) \to P, \neg P, P \vdash P$	Ref
10	$(P \to Q) \to P, \neg P, P, \neg Q \vdash \neg Q$	Ref
11	$(P \to Q) \to P, \neg P, P, \neg Q \vdash P$	9, Mon
12	$(P \to Q) \to P, \neg P, P, \neg Q \vdash \neg P$	2, Mon
13	$(P \to Q) \to P, \neg P, P \vdash Q$	11, 12, $\neg$-el
14	$(P \to Q) \to P, \neg P \vdash P \to Q$	13, RD
15	$(P \to Q) \to P \vdash P$	14,8,$\neg$-el
16	$\vdash ((P \to Q) \to P) \to P$	15, RD

$\square$

O último termo desta prova trata-se de um exemplar da Lei de Peirce[19].

4.14 Escólio. *Toda Geração Funcional é uma Geração Relacional, mas o contrário não é verdadeiro.*

4.15 Definição. (**Conjunto Gerado em** G)
Seja G uma geração.
N é dito o *conjunto gerado em* G se N é a coleção de termos de sequências construtivas em G. Dizemos também, neste caso, que N é gerado em U a partir de B por $\mathcal{R}$.

[19] Charles Sanders Peirce, filósofo e matemático americano (1839–1914).

4.16 Notação. Se G é uma geração, notamos o conjunto gerado em G por $\underline{G}$.

4.17 Definição.
Sejam $\begin{cases} I \text{ uma coleção,} \\ f \text{ uma função.} \end{cases}$
I é fechado com respeito a f se, para qualquer $j \in \mathbb{N}$ e para quaisquer $x_1, \ldots, x_j \in I$, se a tupla $\langle x_1, \ldots, x_j \rangle \in \mathcal{D}(f)$, então $f(x_1, \ldots, x_j) \in I$.

4.18 Exemplo.
A soma de dois números naturais é sempre um número natural.
Então, o conjunto dos números naturais é fechado em relação à adição.
Agora considere a operação de subtração sobre números naturais.
Seja a subtração $2 - 10 = -8$.
Posto isso, temos que o conjunto dos números naturais não é fechado em relação à subtração, pois a subtração de dois números naturais nem sempre é um número natural.

4.19 Definição.
Sejam $\begin{cases} I \text{ uma coleção,} \\ R \text{ uma relação.} \end{cases}$
I é fechado com respeito a R se, para qualquer $j \in \mathbb{N}$, para quaisquer $x_1, \ldots, x_j \in I$ e para qualquer m, se a tupla $\langle x_1, \ldots, x_j \rangle R\, m$, então $y \in I$.

4.20 Exemplo.
Considere a regra (não primitiva) para LQC dada por $\dfrac{\Gamma, P \vdash \bot}{\Gamma \vdash \neg P}$. Então a coleção de sequentes corretos em LQC é fechada com respeito a esta regra.

4.21 Definição. (**Conjunto Induzido em uma Geração Funcional**)
Seja $G = \langle U, B, \mathcal{F} \rangle$.
I é *induzido em* G se $\begin{cases} B \subseteq I, \\ \text{para cada } f \in \mathcal{F}, I \text{ é fechado com respeito a } f. \end{cases}$

4.22 Exemplo.
Seja $G = \langle \mathbb{R}, \{0\}, \{suc\} \rangle$, onde

$$\begin{aligned} suc : & \mathbb{R} \to \mathbb{R} \\ & x \mapsto x + 1 \end{aligned}$$

Temos que $\mathbb{Q}$ é um conjunto induzido em G.

4.23 Definição. (**Conjunto Induzido em uma Geração (Relacional)**)
Seja $G = \langle U, B, \mathcal{R} \rangle$.
I é *induzido em* G se $\begin{cases} B \subseteq I, \\ \text{para cada } R \in \mathcal{R}, I \text{ é fechado com respeito a } R. \end{cases}$

4.24 Exemplo.
Seja $G = \langle U, B, \mathcal{R}\rangle$ onde U é a coleção de sequentes em LQC, B é a coleção de exemplares de esquemas do cálculo de sequentes para LQC e $\mathcal{R}$ é a coleção de regras do cálculo de sequentes para LQC.
Então a coleção de sequentes corretos no cálculo de sequentes para LQC é um conjunto induzido em G, que também é o menor dos conjuntos induzidos em G.

4.25 Definição.
Sejam $\begin{cases} f \text{ uma função,} \\ I \text{ uma coleção,} \\ \Delta(n) \text{ uma propriedade para elementos } n \in I. \end{cases}$
Dizemos que f *preserva* Δ *em* I se, para qualquer $j \in \mathbb{N}$ e para quaisquer $n_1, \ldots, n_j \in I$, se $\Delta(n_1), \ldots, \Delta(n_j)$ e $\langle n_1, \ldots, n_j\rangle \in \mathcal{D}(f)$, então $f(n_1, \ldots, n_j) \in I$ e vale $\Delta(f(n_1, \ldots, n_j))$.

4.26 Exemplo.
Considere a coleção de fórmulas em LPC.
Seja $\Delta(Q)$ a propriedade "para quaisquer fórmulas P_1, P_2 e S, $P_1 \leftrightarrow P_2 \vdash Q(S\|P_1) \leftrightarrow Q(S\|P_2)$".
Temos que as funções $\xi_\neg, \xi_\to, \xi_\wedge$ e $\xi_\vee$ preservam Δ na coleção de fórmulas em LPC.

4.27 Definição.
Sejam $\begin{cases} R \text{ uma relação,} \\ I \text{ uma coleção,} \\ \Delta(n) \text{ uma propriedade para elementos } n \in I. \end{cases}$
Dizemos que R *preserva* Δ *em* I se, para qualquer $j \in \mathbb{N}$, para quaisquer $n_1, \ldots, n_j \in I$ e para qualquer m, se $\Delta(n_1), \ldots, \Delta(n_j)$ e $\langle\langle n_1, \ldots, n_j\rangle R\, m\rangle$, então $m \in I$ e vale $\Delta(m)$.

4.28 Exemplo.
Considere a coleção dos sequentes corretos em LQC.
Seja $\Delta(\Gamma \models_{\text{LQC}} P)$ a propriedade "existe um subconjunto Γ' finito de Γ tal que $\Gamma' \models_{\text{LQC}} P$".
Temos que as regras do cálculo de sequentes para LQC preservam Δ na coleção dos sequentes corretos em LQC.

4.29 Definição.
Se G uma geração, notamos por $\overline{G} = \bigcap\{I | I \text{ é induzido em G}\}$.

4.30 Proposição.
Se G é uma geração, então:
(i) $\overline{G}$ é um conjunto induzido em G.
(ii) $\forall G'(G'$ é o conjunto induzido em $G \to \overline{G} \subseteq G')$.
Dizemos também, simplesmente, que $\overline{G}$ é o *menor conjunto induzido em* G.

4.31 Lema.
Se $\begin{cases} G \text{ é uma geração,} \\ I \text{ é um conjunto induzido em } G, \end{cases}$ *então* $\underline{G} \subseteq I$.

Prova:
Considere $G = \langle U, B, \mathcal{R} \rangle$.
Seja $n_1, \ldots, n_l$ uma sequência construtiva em G.
Suponha por absurdo que algum termo desta sequência não pertence a I.
Seja $S = \{i | i \in \{1, \ldots, l\} \wedge n_i \notin I\}$.
Como $S \neq \varnothing$, daí S possui um elemento mínimo i.
Daí $n_i \notin I$.
Sendo n_i um termo da sequência construtiva $n_1, \ldots, n_l$, temos que uma das seguintes situações deve ocorrer:
(i) $n_i \in B$,
(ii) existem naturais $i_1, \ldots, i_j$ precedendo i e existe $R \in \mathcal{R}$ tal que $\langle n_{i_1}, \ldots, n_{i_j} \rangle R\, n_i$.
Como $I \supseteq B$, temos que $n_i \notin B$, daí, necessariamente, como $n_1, \ldots, n_l$ é uma sequência construtiva em G, temos que a segunda condição dada acima deveria valer.
Como i é o elemento mínimo de S, temos que $n_{i_1}, \ldots, n_{i_j} \in I$, mas, como I é fechado com respeito a R, daí temos que $n_i \in I$, o que é absurdo, logo, todos os termos desta sequência pertencem a I, ou seja, de modo geral, temos que todos os termos da sequência construtiva em G pertencem a I, portanto, $\underline{G} \subseteq I$. $\square$

4.32 Corolário. *Se G é uma geração, $\underline{G} \subseteq \overline{G}$.*

4.33 Teorema. *Se G é uma geração, $\underline{G} = \overline{G}$.*

O conceito de *Geração Própria* é necessário para iniciar a apresentação do princípio da indução estrutural, que será feita posteriormente.

4.34 Definição. (**Geração Própria**)
Seja $G = \langle U, B, \mathcal{R} \rangle$ uma geração.
G é dita ser uma geração própria se $U = \underline{G}$. Ou seja, uma geração é própria quando o conjunto gerado for igual ao ambiente desta geração.
Se $\mathcal{R}$ é uma coleção de funções, então G também é dito ser uma *Geração Própria Funcional*, caso contrário, G é dito ser uma *Geração Própria Relacional*.

4.35 Exemplo. (Geração Própria Funcional)
Ambiente: Fórmulas em LPC.
Elementos Iniciais: Coleção de letras sentenciais.
Relações Geradoras: $\{\xi_\neg, \xi_\rightarrow, \xi_\wedge, \xi_\vee\}$.

4.36 Exemplo. (Geração Própria Relacional)
Ambiente: Sequentes corretos no cálculo de sequentes para LQC.
Elementos Iniciais: Exemplares de esquemas do cálculo para LQC.
Relações Geradoras: {Tran, Mon, RD, ¬-int, ¬-el, Gen, ∃-el}.

4.37 Escólio. *Toda Geração Própria Funcional é também uma Geração Própria Relacional, mas a recíproca não é verdadeira.*

4.38 Escólio.
Se $\begin{cases} \langle N, B, \mathcal{R}\rangle \text{ é uma geração própria,} \\ f \in \mathcal{R}, \\ f \text{ é uma função,} \end{cases}$
então f preserva Δ em N se, e somente se, para qualquer $j \in \mathbb{N}$ e para quaisquer $n_1, \ldots, n_j \in N$, tais que $\Delta(n_1), \ldots, \Delta(n_j)$ e $\langle n_1, \ldots, n_j\rangle \in \mathcal{D}(f)$, então $\Delta(f(n_1, \ldots, n_j))$.

4.39 Escólio.
Se $\begin{cases} \langle N, B, \mathcal{R}\rangle \text{ é uma geração própria,} \\ R \in \mathcal{R}, \end{cases}$
então R preserva Δ em N se, e somente se, para qualquer $j \in \mathbb{N}$, para quaisquer $n_1, \ldots, n_j \in N$ e para qualquer m, se $\Delta(n_1), \ldots, \Delta(n_j)$ e $\langle n_1, \ldots, n_j\rangle R\, m$, então $\Delta(m)$.

4.40. (**Princípio da Indução Estrutural**)
Sejam $\begin{cases} \langle N, B, \mathcal{R}\rangle \text{ uma geração própria,} \\ \Delta(x) \text{ uma propriedade para elementos, sendo } n \in N. \end{cases}$

Se $\begin{cases} \forall n \in B,\ \Delta(n), \\ \forall R \in \mathcal{R},\ R \text{ preserva } \Delta \text{ em } N, \end{cases}$ então $\forall n \in N,\ \Delta(n)$.

4.41 Escólio.
Se $\langle N, \Theta, s\rangle$ é um Sistema de Peano, então $\langle N, \{\Theta\}, \{s\}\rangle$ é uma geração própria funcional, daí o princípio da indução fraca em Sistemas de Peano é um caso particular do Princípio da Indução Estrutural.

4.42 Exemplo. (Princípio da Indução Estrutural na geração de fórmulas em LPC)
Considere o exemplo 4.35.
Seja $\Delta(P)$ uma propriedade para fórmulas P em LPC.
Se para qualquer letra sentencial p, vale $\Delta(p)$, e $\xi_\neg, \xi_\rightarrow, \xi_\wedge$ e $\xi_\vee$ preservam Δ na coleção de fórmulas em LPC, então, para qualquer fórmula P em LPC, $\Delta(P)$.
Seja $\# \in \{\rightarrow, \wedge, \vee\}$.
Dizer que $\xi_\#$ preserva Δ na coleção de fórmulas em LPC, significa, neste caso, que, para qualquer fórmula P e Q em LPC, $\Delta(P)$ e $\Delta(Q)$ implica que $\Delta(P\#Q)$.
Dizer que $\xi\neg$ preserva Δ na coleção de fórmulas em LPC significa, neste caso, que, para qualquer fórmula P em LPC, $\Delta(P)$ implica que $\Delta(\neg P)$.

4.43 Exemplo. (Princípio da Indução Estrutural na geração de sequentes corretos com respeito ao cálculo de sequentes para LQC)
Considere o exemplo 4.36.
Seja $\Delta(\Gamma \vdash P)$ uma propriedade para sequentes corretos "$\Gamma \vdash P$" em LQC.

Se $\begin{cases} \bullet \text{ para qualquer exemplar "}\Gamma \vdash P\text{" dos esquemas Ref, MP, } \wedge\text{-int, } \wedge\text{-el,} \\ \quad \vee\text{-int, PC, } \forall\text{-el, } \exists\text{-int, vale } \Delta(\Gamma \vdash P), \\ \bullet \text{ Tran, Mon, RD, } \neg\text{-int, } \neg\text{-el, Gen e } \exists\text{-el preservam } \Delta \text{ na coleção de se-} \\ \quad \text{quentes corretos com respeito ao cálculo de sequentes para LQC,} \end{cases}$

então $\Gamma \vdash_{\mathrm{LQC}} P$ implica que $\Delta(\Gamma \vdash_{\mathrm{LQC}} P)$.
Se R é uma regra do cálculo de sequentes para LQC, dizer que R preserva Δ na coleção de sequentes corretos do cálculo para LQC significa dizer que, para cada aplicação $\frac{\langle \Gamma_1, P_1 \rangle, \ldots, \langle \Gamma_n, P_n \rangle}{\langle \varphi, Q \rangle}$ de R,

$$\text{se } \begin{cases} \Gamma \vdash_{\mathrm{LQC}} P_1, \ldots, \Gamma \vdash_{\mathrm{LQC}} P_n, \\ \Delta(\Gamma \vdash_{\mathrm{LQC}} P_1), \ldots, \Delta(\Gamma \vdash_{\mathrm{LQC}} P_n), \end{cases} \qquad \text{então } \Delta(\varphi \vdash_{\mathrm{LQC}} Q).$$

Capítulo 7

Condições Gerais de Correção e Completude de um Cálculo de Sequentes com respeito a uma Semântica de Valorações

Este capítulo apresenta os resultados gerais de correção e completude de um cálculo de sequentes com respeito a uma semântica de valorações. Os resultados para lógicas específicas serão aplicados nas seções §9.3, para LPC, e em §10.3, para LQC.

§1. Correção e Completude

1.1 Notação. Sejam $\mathcal{L}_1$ e $\mathcal{L}_2$ lógicas que possuem as mesmas linguagens, de modo que $\mathcal{L}_1$ é definida por um cálculo de sequentes e $\mathcal{L}_2$ é definida por uma semântica.

As propriedades de *correção* e de *completude* estabelecem uma relação entre o domínio sintático de $\mathcal{L}_1$ e o domínio semântico de $\mathcal{L}_2$.

1.2 Definição.
Dizemos que $\mathcal{L}_1$ é *correto com respeito a* $\mathcal{L}_2$ se, para todo Γ e para todo P, $\Gamma \vdash_{\mathcal{L}_1} P$ implica em $\Gamma \models_{\mathcal{L}_2} P$.

Ou seja, $\mathcal{L}_1$ possui a propriedade da correção sss seu cálculo de sequentes demonstra somente fórmulas que são válidas do ponto de vista da semântica definida em $\mathcal{L}_2$.

A recíproca da propriedade da correção é a propriedade da completude:

1.3 Definição.
Dizemos que $\mathcal{L}_2$ é *completo com respeito a* $\mathcal{L}_1$ se, para todo Γ e para todo P, $\Gamma \models_{\mathcal{L}_2} P$ implica em $\Gamma \vdash_{\mathcal{L}_1} P$.

A correção e a completude são importantes no estudo e apresentação de uma lógica pois relacionam aspectos sintáticos, tal como $\Gamma \vdash P$, a aspectos semânticos, tal como $\Gamma \models P$.

O teorema da correção diz que, se $\Gamma \vdash_{\mathcal{L}_1} P$, então $\Gamma \models_{\mathcal{L}_2} P$, e o teorema da completude diz que se $\Gamma \models_{\mathcal{L}_2} P$, então $\Gamma \vdash_{\mathcal{L}_1} P$, para qualquer coleção de fórmulas Γ e qualquer fórmula P. Se ambos podem ser provados, então o cálculo de sequentes $\mathcal{L}_1$ é correto e completo com respeito à semântica $\mathcal{L}_2$.

O teorema da correção e o teorema da completude podem ser demonstrados entre as vias sintática, semântica e automatização do raciocínio. A semântica normalmente é um ponto de referência, um modelo de verdade para o sistema lógico. Se for possível verificar a correção e a completude, então os mesmos podem ser realizados entre um cálculo de sequentes e a semântica; um sistema de tablôs e a semântica; um sistema de dedução natural e a semântica. Mas também entre semânticas (semântica 1, semântica 2, ...); entre cálculos (cálculo 1, cálculo 2, ...); entre o cálculo e o sistema de tablôs; dentre outros. Neste trabalho, a correção e a completude são verificadas entre cálculo de sequentes e semânticas.

Quando sabemos que o cálculo de sequentes $\mathcal{L}_1$ e a semântica $\mathcal{L}_2$ definem uma mesma lógica, costumamos designar $\mathcal{L}_1$ e $\mathcal{L}_2$ pelo mesmo nome. Usualmente, por abuso de linguagem, usamos o mesmo nome para $\mathcal{L}_1$ e $\mathcal{L}_2$, com a condição de mostrar posteriormente que o cálculo $\mathcal{L}_1$ e a semântica $\mathcal{L}_2$ definem a mesma lógica.

As próximas seções elucidam a essência deste assunto de modo compreensivo, ou seja, a coincidência entre a sintaxe e a semântica de um sistema lógico com o objetivo de flexibilizar a sua argumentação.

§2. Correção do Cálculo de Sequentes de uma Lógica

A correção da lógica $\mathcal{L}_1$ é feita através do Princípio da Indução Estrutural na geração de sequentes corretos com respeito ao cálculo de sequentes para uma lógica, dado inicialmente em 6.4.40.

Um cálculo de sequentes é correto com respeito à semântica da lógica se é possível provar que todas as leis (esquemas e regras) deste cálculo são corretas.

De acordo com a indução sobre sequentes, é necessário demonstrar que a propriedade vale para todos os elementos do conjunto inicial de sequentes e que cada regra geradora, preserva a propriedade, então isso é mostrado esquema por esquema e regra por regra. Dessa forma, provamos o teorema da correção a partir da demonstração de como é gerada a coleção de sequentes corretos com respeito ao cálculo.

Com essas ideias é possível definir o teorema da correção.

2.1 Teorema. ***(da correção de um cálculo de sequentes $\mathcal{L}_1$ com respeito a uma lógica $\mathcal{L}_2$)***
Sejam $\begin{cases} \mathcal{L}_1 \text{ um cálculo de sequentes,} \\ \mathcal{L}_2 \text{ uma lógica,} \end{cases}$ *onde $\mathcal{L}_1$ e $\mathcal{L}_2$ possuem as mesmas linguagens.*
Se cada lei de $\mathcal{L}_1$ é correta com respeito a $\mathcal{L}_2$, então, para qualquer coleção Γ de fórmulas em $\mathcal{L}_1$ e para qualquer fórmula P em $\mathcal{L}_1$, $\Gamma \mathrel{\underset{\mathcal{L}_1}{\vdash}} P$ implica que $\Gamma \mathrel{\underset{\mathcal{L}_2}{\models}} P$.

Prova: É evidente, por indução sobre sequentes corretos em $\mathcal{L}_1$. $\square$

Se alguma aplicação de uma regra é incorreta, então a regra é incorreta. A incorreção de uma regra do cálculo não prova a incorreção deste cálculo porque pode acontecer que no cálculo não sejam derivadas todas as hipóteses dessa aplicação. A incorreção de uma regra apenas acarreta que a correção do cálculo não pode ser demonstrada. Mas não provar a correção do cálculo não acarreta que o cálculo seja incorreto. O que provoca a incorreção de um cálculo é a incorreção do esquema. Se algum exemplar de um esquema é incorreto, então o esquema é incorreto e o cálculo em si é incorreto, pois a regra é uma verdade condicional e o esquema é uma verdade absoluta.

§3. Completude do Cálculo de Sequentes de uma Lógica

A *completude* é a propriedade recíproca da *correção*.

3.1 Pré-Definição.

Sejam $\begin{cases} \mathcal{L} \text{ uma lógica,} \\ L \text{ uma linguagem para } \mathcal{L}, \\ \varphi \text{ uma coleção de fórmulas de } L. \\ P \text{ uma fórmula de } L. \end{cases}$

Consideramos como previamente definido quando φ é uma P-coleção de Henkin em L com respeito à lógica $\mathcal{L}$.

3.2 Teorema. ***(da completude de um cálculo de sequentes $\mathcal{L}_1$ com respeito a uma lógica $\mathcal{L}_2$)***

Sejam $\begin{cases} \mathcal{L}_1 \text{ e } \mathcal{L}_2 \text{ lógicas com as mesmas linguagens,} \\ P \text{ uma fórmula em } \mathcal{L}_1, \\ \Gamma \text{ uma coleção de fórmulas em } \mathcal{L}_1. \\ \mathcal{L}_2 \text{ uma lógica definida por uma semântica de valorações.} \end{cases}$

Considere:

(i) *Para toda linguagem L para $\mathcal{L}_1$ e para toda coleção de fórmulas φ de L, se φ é P-coleção de Henkin em L com respeito à lógica $\mathcal{L}_1$, então $P \notin \varphi$.*

(ii) *Para qualquer linguagem L para a lógica $\mathcal{L}_1$ e para qualquer coleção φ de fórmulas de L, se φ é P-coleção de Henkin em L com respeito à lógica $\mathcal{L}_1$, então existe uma $\mathcal{L}_2$-interpretação para φ tal que para toda fórmula Q de L, $I_V(Q)$ é valor distinguido em $\mathcal{L}_2$ sss $Q \in \varphi$.*

(iii) *Se $\Gamma \not\vdash_{\mathcal{L}_1} P$, então existe uma linguagem L para $\mathcal{L}_1$ e existe uma coleção φ de fórmulas de L tal que*
- *Γ é coleção de fórmulas de L,*
- *P é fórmula de L,*
- *φ é P-coleção de Henkin em L com respeito à lógica $\mathcal{L}_1$,*
- *$\Gamma \subseteq \varphi$.*

Se (i), (ii) e (iii), então $\Gamma \models_{\mathcal{L}_2} P$ implica em $\Gamma \vdash_{\mathcal{L}_1} P$.

Prova:

Assuma as hipóteses e suponha que $\Gamma \not\vdash_{\mathcal{L}_1} P$.

Pela terceira condição das hipóteses assumidas, temos que existe uma linguagem L para $\mathcal{L}_1$ e existe uma coleção φ de fórmulas de L tal que

$\begin{cases} \Gamma \text{ é coleção de fórmulas de } L, \\ P \text{ é fórmula de } L, \\ \varphi \text{ é } P\text{-coleção de Henkin em } L \text{ com respeito à lógica } \mathcal{L}_1, \\ \Gamma \subseteq \varphi. \end{cases}$

Pela primeira condição, temos que φ é P-saturado em L com respeito à lógica $\mathcal{L}_1$, daí $P \notin \varphi$.
Pela segunda condição, existe uma $\mathcal{L}_2$-interpretação I para φ tal que I satisfaz φ.
Temos ainda, pela segunda condição, como P é fórmula de L e $P \notin \varphi$, temos que $I_V(P)$ não é valor distinguido em $\mathcal{L}_2$.
Ou seja, $\varphi \not\models_{\mathcal{L}_2} P$, portanto $\Gamma \not\models_{\mathcal{L}_2} P$.
Logo, $\Gamma \models_{\mathcal{L}_2} P$ implica em $\Gamma \vdash_{\mathcal{L}_1} P$. $\square$

3.3 Definição.
φ é P-saturado em uma linguagem L com respeito a uma lógica $\mathcal{L}$.

§4. Conceitos Adicionais

Os conceitos definidos nesta seção são utilizados, neste trabalho, para a demonstração da completude de LPC e LQC.

4.1 Definição.
φ é *trivial em* $\mathcal{L}$ se, para cada fórmula Q em $\mathcal{L}$, $\varphi \vdash_{\mathcal{L}} Q$.

4.2 Lema. *Se $\mathcal{L}$ possui NC e Tran, então Γ é $\neg$-consistente em $\mathcal{L}$ sss Γ é não-trivial em $\mathcal{L}$.*

Prova da ida: Suponha que Γ é $\neg$-consistente em $\mathcal{L}$.
Se Γ for trivial em $\mathcal{L}$, então existe Q tal que $\Gamma \vdash_{\mathcal{L}} Q$ e $\Gamma \vdash_{\mathcal{L}} \neg Q$, o que é absurdo, logo Γ é trivial em $\mathcal{L}$. $\square$

Prova da volta: Suponha que Γ é não trivial em $\mathcal{L}$, ou seja, Γ não deduz qualquer fórmula em $\mathcal{L}$.
Suponha por absurdo que Γ é $\neg$-consistente em $\mathcal{L}$.
Daí existe uma fórmula Q em $\mathcal{L}$ tal que $\Gamma \vdash_{\mathcal{L}} Q$ e $\Gamma \vdash_{\mathcal{L}} \neg Q$.
Dado uma fórmula P em $\Gamma \vdash_{\mathcal{L}} Q$, por NC, $Q, \neg Q \vdash_{\mathcal{L}} P$, logo, por Tran, $\Gamma \vdash_{\mathcal{L}} P$. $\square$

4.3 Definição.
Se Γ é trivial em $\mathcal{L}$ e $\mathcal{L}$ possui o conectivo monádico da negação e a lei '$P, \neg P \vdash_{\mathcal{L}} Q$', então isso equivale a dizer que Γ também é *inconsistente em* $\mathcal{L}$.

4.4 Definição.
Γ é *inconsistente em* $\mathcal{L}$ se possui o conectivo monádico e se existe uma fórmula P em $\mathcal{L}$ tal que $\begin{cases} \Gamma \vdash_{\mathcal{L}} P, \\ \Gamma \vdash_{\mathcal{L}} \neg P. \end{cases}$

4.5 Lema. $\Gamma \vdash_{\mathcal{L}} P$ *sss* $\Gamma \cup \{\neg P\}$ *é trivial.*

4.6 Definição. (**Coleção P-saturada**)

$$\text{Sejam} \begin{cases} P \text{ uma fórmula de } L, \\ \Gamma \text{ uma coleção de fórmulas de } L, \\ L \text{ é uma linguagem para } \mathcal{L}. \end{cases}$$

Então Γ é *P-saturado em L com respeito a $\mathcal{L}$* se $\Gamma \nvdash_{\mathcal{L}} P$, e para qualquer Γ' em L, se $\Gamma' \nvdash_{\mathcal{L}} P$, então $\Gamma \not\subset \Gamma'$.

De forma similar podemos dizer que, para qualquer Γ' em L, se $\begin{cases} \Gamma' \nvdash_{\mathcal{L}} P, \\ \Gamma \subseteq \Gamma', \end{cases}$ então $\Gamma = \Gamma'$.

Informalmente, temos que Γ é P-saturado se ele for o maior conjunto possível que não deduz P a partir de si mesmo, ou seja, Γ é a coleção maximal de fórmulas em L que não deduz P em $\mathcal{L}$.

Existe a ideia de *maximal não-trivial* utilizada em outros livros-texto (tal como em [6] e [5]), mas optamos por utilizar a ideia de *P-saturado*[1] por ser mais flexível de se aplicar em diversas lógicas.

4.7 Lema.
Seja Γ *uma coleção de fórmulas em* L *tal que* $\Gamma \nvdash_{\mathcal{L}} P$.
Seja Ω *a coleção de todas as coleções de fórmulas em* L, φ, *tal que* $\begin{cases} \varphi \nvdash_{\mathcal{L}} P, \\ \Gamma \subseteq \varphi. \end{cases}$
Ω *satisfaz a hipótese do* Lema de Zorn*, em 2.3.65, ou seja, toda cadeia em* Ω *possui uma cota superior em* Ω.

Prova:
Seja $\mathcal{L}$ uma lógica dotada da regra da monotonicidade, 4.3.3, e da regra da compacidade, 9.2.1.
Seja Σ uma cadeia em Ω[2]. (1)
Spg, considere $\Sigma \neq \varnothing$[3].
Seja Θ um elemento de Σ, ou seja, $\Theta \in \Sigma$.
Então Θ é uma coleção de fórmulas em Σ. (2)
Como $\Sigma \subseteq \Omega$, temos que $\Theta \in \Omega$.
Daí $\Gamma \subseteq \Theta$. (3)
De (2), $\Theta \subseteq \bigcup \Sigma$, donde, de (3), $\Gamma \subseteq \bigcup \Sigma$. (4)
Suponha por absurdo que $\bigcup \Sigma \vdash_{\mathcal{L}} P$.
Então existem fórmulas $Q_1, \ldots, Q_n (n \in \mathbb{N}^*)$ tal que a coleção

[1] Apresentada inicialmente por Newton da Costa, em correspondência enviada ao professor Arthur Buchsbaum.
[2] Que é um subconjunto de Ω.
[3] Se Σ fosse $\varnothing$, o Γ seria sua cota superior.

$\{Q_1, \ldots, Q_n\} \subseteq \bigcup \Sigma$.
Pela RC, $\{Q_1, \ldots, Q_n\} \vdash_{\mathcal{L}} P$. (5)
Para cada $i \in \{1, \ldots, n\}$ existe $\Gamma_i \in \Sigma$ tal que $Q_i \in \Gamma_i$.
$\{Q_1, \ldots, Q_n\} \subseteq \Gamma_1 \cup \ldots \cup \Gamma_n$. (6)
De (5),(6) e pela Mon, $\Gamma_1 \cup \ldots \cup \Gamma_n \vdash_{\mathcal{L}} P$. (7)
Como Σ é uma cadeia, temos que existe $i \in \{1, \ldots, n\}$ e temos que $\Gamma_1 \cup \ldots \cup \Gamma_n = \Gamma_i$. (8)
De (7) e (8), $\Gamma_i \vdash_{\mathcal{L}} P$, o que é absurdo, pois $\Gamma_i \in \Sigma$, $\Sigma \in \Omega$ e pela forma como definimos Ω, os elementos de Ω não podem deduzir P.
Logo, $\bigcup \Sigma \nvdash_{\mathcal{L}} P$. (9)
De (4) e (9), $\bigcup \Sigma \in \Omega$.
Temos obviamente que todo elemento de Σ está contido em $\bigcup \Sigma$.
Portanto, $\bigcup \Sigma$ é cota superior de Σ. $\square$

4.8 Lema.
Se $\Gamma \nvdash_{\mathcal{L}} P$, então existe Γ' tal que $\Gamma \subseteq \Gamma'$ e Γ' é P-saturado em $\mathcal{L}$.

4.9 Lema.
Se Γ é P-saturado em $\mathcal{L}$ e $\mathcal{L}$ é dotada da RD, MP, Tran, NC(i), RDj, TEx, PC, $\wedge$-el, Mon, Ref e $\wedge$-int, então para qualquer Q, $Q \in \Gamma$ ou $(Q \to P) \in \Gamma$.

Prova:
Suponha que Γ é P-saturado em $\mathcal{L}$.
Suponha por absurdo que existe um Q tal que $Q \notin \Gamma$ e $(Q \to P) \in \Gamma$.
Por saturado, $\Gamma \cup \{Q\} \vdash_{\mathcal{L}} P$ e $\Gamma \cup \{Q \to P\} \vdash_{\mathcal{L}} P$.
Por RD, $\Gamma \vdash_{\mathcal{L}} Q \to P$ e $\Gamma \vdash_{\mathcal{L}} (Q \to P) \to P$.
Por Tran, $Q \to P, (Q \to P) \to P \vdash_{\mathcal{L}} P$, portanto $\Gamma \vdash_{\mathcal{L}} P$, o que é absurdo porque Γ é P-saturado.
Portanto, para todo Q, $Q \in \Gamma$ ou $(Q \to P) \in \Gamma$. $\square$

Precisamos mostrar que uma coleção saturada possui certas características concernentes a cada conectivo. Com essas características fica fácil mostrar que existe uma valoração que satisfaz essa coleção saturada, ou seja, que toda coleção saturada é também *satisfatível*.

De forma específica, isso é feito para LPC em 9.3.5, e para LQC em 10.3.5.

4.10 Lema.
Seja Γ *saturado em* L *com respeito a* $\mathcal{L}$*. Então*
(i) Γ *é* $\neg$*-consistente*[4].
(ii) Γ *é* $\neg$*-completo em* L[5].
(iii) $\top \in \Gamma$.
(iv) $\bot \notin \Gamma$.
(v) $(Q \to R) \in \Gamma$*, sss* $Q \notin \Gamma$ *ou* $R \in \Gamma$.
(vi) $(Q \wedge R) \in \Gamma$*, sss* $Q \in \Gamma$ *e* $R \in \Gamma$.
(vii) $(Q \vee R) \in \Gamma$*, sss* $Q \in \Gamma$ *ou* $R \in \Gamma$.

Prova:

- (i): Se Γ não fosse $\neg$-consistente, então existiria uma fórmula Q, tal que $\Gamma \vdash_{\mathcal{L}} Q$ e $\Gamma \vdash_{\mathcal{L}} \neg Q$.
  Mas a partir dessa contradição $\Gamma \vdash_{\mathcal{L}} P$, o que não pode acontecer, uma vez que Γ é P-saturado.
- (ii): Suponha que $\Gamma \nvdash_{\mathcal{L}} Q$ e $\Gamma \nvdash_{\mathcal{L}} \neg Q$.
  Se $\Gamma \nvdash_{\mathcal{L}} Q$, então pelo lema 4.9, $\Gamma \vdash_{\mathcal{L}} Q \to P$. (1)
  E se $\Gamma \nvdash_{\mathcal{L}} \neg Q$, então $\Gamma \vdash_{\mathcal{L}} \neg Q \to P$. (2)
  Seja $\mathcal{L}$ dotada de TEx $(Q \vee \neg Q)$. De (1), (2) e pela PC temos que $\Gamma \vdash_{\mathcal{L}} P$. Assim Γ não seria $\neg$-completo. Então Γ deve ser $\neg$-completo.
- (iii): Se $\Gamma \vdash_{\mathcal{L}} \top$, então $\top \in \Gamma$.
- (iv): Se $\Gamma \vdash_{\mathcal{L}} \bot$, como $\bot \vdash_{\mathcal{L}} P$, daí $\Gamma \vdash_{\mathcal{L}} P$, o que é absurdo, logo $\Gamma \nvdash_{\mathcal{L}} \bot$ e $\bot \notin \Gamma$.
- (v): Suponha que $R \in \Gamma$, então $\Gamma \vdash_{\mathcal{L}} R$. Por Mon, $\Gamma, Q \vdash_{\mathcal{L}} R$ e por RD, $\Gamma \vdash_{\mathcal{L}} Q \to R$.
- (vi): De $Q \in \Gamma$ e $R \in \Gamma$, por Ref temos que $\Gamma \vdash_{\mathcal{L}} Q$ e $\Gamma \vdash_{\mathcal{L}} R$. Por $\wedge$-int, $Q, R \vdash_{\mathcal{L}} Q \wedge R$ e por Tran, $\Gamma \vdash_{\mathcal{L}} Q \wedge R$.
- (vii): Suponha de $Q \in \Gamma$ que $\Gamma \vdash_{\mathcal{L}} Q$, por Ref. Então $Q \vdash_{\mathcal{L}} Q \vee R$ por $\vee$-int. Daí $\Gamma \vdash_{\mathcal{L}} Q \vee R$.

$\square$

4.11 Definição.
Seja $*$ um conectivo monádico em $\mathcal{L}$.
φ é $*$-consistente em $\mathcal{L}$ se não existe uma fórmula Q em $\mathcal{L}$ tal que $\varphi \vdash_{\mathcal{L}}$ e $\varphi \vdash_{\mathcal{L}} *Q$.

[4]É consistente com respeito ao conectivo monádico da negação, ou seja, não deduz simultaneamente, P e $\neg P$.

[5]Seja uma fórmula P qualquer. Γ deduz P ou deduz a sua negação, ou seja, $\neg P$.

4.12 Definição.
Seja $*$ um conectivo monádico em $\mathcal{L}$.
φ é $*$-completo em $\mathcal{L}$ com respeito a $\mathcal{L}$ se para cada fórmula Q de L, $\varphi \vdash_{\mathcal{L}} Q$ ou $\varphi \vdash_{\mathcal{L}} *Q$.

4.13 Lema.
Seja Γ uma coleção de fórmulas em L, onde L é uma linguagem para $\mathcal{L}$ e P é uma fórmula de $\mathcal{L}$.
Se $\Gamma \nvdash_{\mathcal{L}} P$ então existe φ tal que
$\begin{cases} \Gamma \subseteq \varphi, \\ \varphi \text{ é } P\text{-saturado em } L \text{ com respeito a } \mathcal{L}. \end{cases}$

Prova:
Seja L uma linguagem tal que Γ é uma coleção de fórmulas em $\mathcal{L}$.
Seja Ω a coleção de coleções de fórmulas φ em L tal que
$\begin{cases} \Gamma \subseteq \varphi, \\ \varphi \nvdash_{\mathcal{L}} P. \end{cases}$
Pelo lema 4.7, toda coleção em Ω possui uma cota superior em Ω, donde, pelo *Lema de Zorn* em 2.3.65, Ω possui um elemento maximal φ.
Como $\varphi \in \Omega$, temos que $\begin{cases} \Gamma \subseteq \varphi, \\ \varphi \nvdash_{\mathcal{L}} P. \end{cases}$
Suponha que φ' é a coleção de fórmulas em L tal que $\varphi' \nvdash_{\mathcal{L}} P$. (1)
Suponha por absurdo que $\varphi \subseteq \varphi'$. (2)
Por Tran, temos que $\Gamma \subseteq \varphi'$. (3)
De (1) e (3), $\varphi' \in \Omega$. (4)
De (4) e (2), φ não é elemento maximal de Ω, porque elemento maximal não pode ser majorado por outro elemento do conjunto, logo $\neg(\varphi \subset \varphi')$, o que é absurdo.
Ou seja, para qualquer φ' tal que φ' é uma coleção de fórmulas de L e $\varphi' \nvdash_{\mathcal{L}} P$, então $\neg(\varphi \subset \varphi')$.
Logo, φ é P-saturado em L, donde φ é P-saturado. □

4.14 Definição.
Γ é saturado com respeito a uma lógica $\mathcal{L}$ sss:
(i) existe uma linguagem L para $\mathcal{L}$,
(ii) existe uma fórmula P de L tal que Γ é P-saturado em L com respeito a $\mathcal{L}$.

4.15 Lema.
Se $\begin{cases} \varphi \text{ é } P\text{-saturado em } L \text{ com respeito à lógica } \mathcal{L}, \\ Q \text{ é uma fórmula de } L, \end{cases}$
então ou $Q \in \varphi$ ou $\varphi \cup \{Q\} \vdash_{\mathcal{L}} P$.

Prova: Através da Lei da Reflexividade e da Regra da Transitividade Geral. $\square$

4.16 Lema.
Se $\mathcal{L}$ possui Ref, então, para qualquer fórmula Q em $\mathcal{L}$, $Q \in \varphi$ implica que $\varphi \mid\!\frac{\quad}{\mathcal{L}}\, Q$.

4.17 Lema.
Sejam $\begin{cases} \varphi \text{ uma coleção } P\text{-saturada em } L \text{ com respeito à lógica } \mathcal{L}. \\ Q \text{ uma fórmula de } L, \\ \mathcal{L} \text{ uma lógica contendo Ref e TG,} \end{cases}$
então $\varphi \mid\!\frac{\quad}{\mathcal{L}}\, Q$ implica que $Q \in \varphi$.

Prova:
Assuma as hipóteses.
Suponha por absurdo que $Q \notin \varphi$, mas $\varphi \cup \{Q\} \mid\!\frac{\quad}{\mathcal{L}}\, P$, daí $\varphi \mid\!\frac{\quad}{\mathcal{L}}\, P$, o que é absurdo, logo $Q \in \varphi$. $\square$

Considerando 4.16 e 4.17, o seguinte lema é definido:

4.18 Lema.
Sejam $\begin{cases} \varphi \text{ uma coleção } P\text{-saturada em } L \text{ com respeito à lógica } \mathcal{L}. \\ Q \text{ uma fórmula de } L, \\ \mathcal{L} \text{ uma lógica contendo Ref e TG,} \end{cases}$
então $Q \in \varphi$ sss $\varphi \mid\!\frac{\quad}{\mathcal{L}}\, Q$.

Prova:
Se $Q \in \varphi$, então, por Ref, $\varphi \mid\!\frac{\quad}{\mathcal{L}}\, Q$.
Suponha que $\varphi \mid\!\frac{\quad}{\mathcal{L}}\, Q$.
Se $Q \notin \varphi$, então $\varphi \cup \{Q\} \mid\!\frac{\quad}{\mathcal{L}}\, P$, e daí, se $\varphi \mid\!\frac{\quad}{\mathcal{L}}\, Q$, teríamos então, pela Regra da Transitividade Geral, $\varphi \mid\!\frac{\quad}{\mathcal{L}}\, P$, o que é absurdo, logo $\varphi \not\mid\!\frac{\quad}{\mathcal{L}}\, Q$. $\square$

4.19 Lema.
Se $\begin{cases} \varphi \text{ é uma coleção } P\text{-saturada em } L \text{ com respeito à lógica } \mathcal{L}, \\ \mathcal{L} \text{ possui Ref e TG,} \end{cases}$
então $P \notin \varphi$.

Prova:
Assuma as hipóteses.
Suponha por absurdo que $P \in \varphi$.
Se $P \in \varphi$, então pelo lema 4.16, $\varphi \mid\!\frac{\quad}{\mathcal{L}}\, P$, mas como φ é P-saturado em L com respeito a $\mathcal{L}$, então $\varphi \not\mid\!\frac{\quad}{\mathcal{L}}\, P$, logo $P \notin \varphi$. $\square$

4.20 Definição. (**Extensão e Restrição de $\mathcal{L}$-interpretações**[6])
Seja I uma $\mathcal{L}$-interpretação e I' uma extensão de I.
Diminuindo a quantidade de letras sentenciais, constantes, sinais funcionais e sinais predicativos que são interpretadas por I', obtemos I, que é chamada de *restrição de I'*. Também dizemos que I' é uma *extensão* de I.
Se I' é uma extensão de I, então I e I' interpretam as mesmas letras sentenciais, constantes, sinais funcionais e sinais predicativos. Verifica-se que $I(P) = I'(P)$ para toda fórmula fechada P de uma linguagem L de $\mathcal{L}$.

O conceito de coleção saturada é necessário para lidar com conectivos $\neg, \vee, \wedge$, mas não é suficiente para lidar com quantificadores. Por isso será necessário usar uma técnica adicional que é atribuída a Henkin[7].

Abaixo descrevemos a motivação do uso de coleção de Henkin para provar, em §10.3, a completude do cálculo de LQC. Com esse conceito será possível verificar que uma coleção simultaneamente saturada e de Henkin, dita *P-coleção de Henkin*, é satisfatível na semântica de LQC.

Para fazer uma prova da completude é preciso obter uma extensão de Γ que não acarreta P e que seja satisfatível para as fórmulas.

No caso da LPC conseguimos mostrar que a extensão do Γ é uma coleção saturada e é um conjunto satisfatível apenas com os conceitos de coleção saturada apresentados anteriormente.

Ao passo que em LQC pode existir uma coleção saturada que tenha uma fórmula quantificada, mas talvez não seja possível fazer uma interpretação convencível de que para $\exists x P$ ou $\forall x P$ seja atribuído um valor verdadeiro. Por exemplo, em uma coleção saturada de LPC P e Q, então também teremos $P \wedge Q$ se o $I_V(P) = \mathrm{v}$ e $I_V(Q) = \mathrm{v}$, assim, $P \wedge Q$ é verdadeiro. A mesma ideia é válida para $\vee, \neg$, por exemplo. Já para fórmulas quantificadas é diferente. Como se convencer de que $\exists x P$ e $\forall x P$ são verdadeiros? Em virtude do que foi mencionado, a ideia de P-coleção de Henkin em L com respeito a uma lógica $\mathcal{L}$ será útil caso exista uma coleção saturada que possua fórmulas quantificadas.

Então modelamos essas fórmulas como sendo 'termos', como os próprios elementos do universo de discurso. Na prática deveríamos ter $I(P(x|t))$=v, mas pode acontecer que exista uma coleção saturada em uma linguagem dita 'pobre', em que a coleção de termos não ocorra no universo de discurso e ela não deixe de ser saturado por esse motivo.

Devemos forçar o seguinte: quando aparecer uma fórmula da forma $\exists x P$, $\forall x P$, deve aparecer também $I(P(x|t)) = \mathrm{v}$. Se a coleção for saturada

[6] Mais detalhes sobre extensão e restrição de interpretações podem ser obtidas em [8].
[7] Ver artigo em L. Henkin, *The Completeness of the First-Order Functional Calculus*, The Journal of Symbolic Logic, 14, 159-166 (1949).

e essa característica não puder ser conseguida, então não é possível obter uma interpretação que satisfaça esse conjunto com fórmulas quantificadas.

4.21 Definição. (**Coleção de Henkin**)
Seja $\mathcal{L}$ uma lógica cujos quantificadores são '$\forall$' e '$\exists$'.
φ é uma coleção de Henkin em L com respeito a $\mathcal{L}$ se as seguintes condições são satisfeitas:
(i) Se, para todo termo t de L, $P(x|t) \in \varphi$, então $\forall x P \in \varphi$.
(ii) Se $\exists x P \in \varphi$, então existe um termo t de L tal que $P(x|t) \in \varphi$.

Capítulo 8

O Método dos Tablôs

A via da automatização preferencial será, ao longo de todo este trabalho, o *método dos tablôs*.

Neste capítulo uma apresentação geral do método dos tablôs é feita, de uma forma independente da aplicação do mesmo para uma lógica específica, juntamente com provas gerais de correção e completude dos sistemas de tablôs com respeito às semânticas de uma dada lógica.

Para este fim, também é dado um conceito geral de semântica, o qual corresponde a uma ampla classe de lógicas, e através do mesmo é definido satisfabilidade e relação de consequência.

§1. Caracterização do Método dos Tablôs

O método dos tablôs, concebido por [25], aprimorado e propagado principalmente por [26] e [24], é um método de prova por refutação, no qual um teorema é provado pelo insucesso na tentativa de construção sistemática de um modelo para a sua negação. De acordo com [47], o método tem fundamento semântico: ao tentar provar que P é um teorema lógico (ou que P é teorema de uma teoria Γ), o que o método em sua concepção original faz, de fato, é verificar a impossibilidade da satisfação de $\neg P$ (ou da satisfação simultânea de Γ e $\neg P$).

A adaptabilidade e a flexibilidade do método dos tablôs o torna extremamente atraente ao projeto de automatização do raciocínio, o qual consiste na sua modelagem lógica, seguida do desenvolvimento de métodos de prova para a lógica empregada e da sua efetiva implementação na forma de um raciocinador automático[1].

[1] Um raciocinador automático é um programa que pode, munido de uma base de dados con-

1.1 Convenção. Tableau para uso no singular e Tableaux para uso no plural são palavras de origem francesa que significam "árvore" ou "árvores", respectivamente. No escopo da lógica referem-se à árvore(s) de fórmula(s). Em português, usaremos a palavra Tablô para uso no singular e Tablôs para uso no plural.

Com a finalidade de generalizar a potencialidade do método dos tablôs, procuramos abstrair suas características essenciais em uma generalização cujas instâncias podem ser aplicadas a uma grande variedade de lógicas.

§2. O Método da Confutação Generalizado

Um sistema de tablôs é um método de prova que consiste na geração de uma árvore de fórmulas em uma dada lógica (tablô), originada de um *tablô inicial*. *A árvore inicial* ou *tablô inicial* é gerada a partir de uma dada fórmula de uma lógica, da qual se pretende saber algo, tal como a sua validade ou a sua satisfabilidade, dependendo do sistema de tablôs.

Esta árvore é a primeira de uma sequência de árvores, onde cada árvore não inicial sucede a anterior, aplicando uma *regra de expansão* para um nó não usado ou não marcado. Cada ramo desta árvore em crescimento é aberto ou fechado, de acordo com um *critério de fechamento* previamente definido. Ramos fechados não crescem mais, enquanto ramos abertos não exauridos são expandidos através de aplicações de regras para seus nós ainda não utilizados. O objetivo do procedimento é *fechar* todos os ramos, provando assim a condição desejada com respeito à fórmula dada (geralmente a sua validade ou a sua satisfabilidade). Semanticamente, o fechamento de todos os ramos do tablô estabelece a validade ou insatisfabilidade da fórmula na sua raiz, dependendo do sistema de tablôs, caso contrário, a existência de um tablô com um ramo exaurido (aberto) implica a invalidade ou a satisfabilidade desta fórmula .

Neste trabalho são considerados sistemas de tablôs para os quais o fechamento de todos os ramos implica a insatisfabilidade da fórmula na raiz, caso contrário, essa fórmula é satisfatível.

Enquanto o método dos tablôs por prova direta investiga a validade da fórmula geradora do tablô inicial, conforme apresentado em [48], o método dos tablôs por confutação investiga a satisfabilidade do tablô inicial, o qual corresponde semanticamente, na Lógica Clássica, à negação da possível tese.

Duas lógicas $\mathcal{L}$ and $\mathcal{L}'$ são consideradas, onde L and L' são respectivamente a linguagem *inicial* e de *trabalho* de um sistema de tablôs genérico,

---

tendo "conhecimento" codificado em uma linguagem lógica (base de conhecimento) e de acordo com a lógica considerada, responder a perguntas efetuando raciocínio sobre este conhecimento.

o qual será definido a seguir. A primeira lógica é aquela para a qual o sistema de tablôs é construído e a segunda é auxiliar, ou seja, é definida para permitir, no caso do método dos tablôs, o estudo da primeira lógica. O algoritmo sempre trabalha com árvores de fórmulas de L', isto é, fórmulas da linguagem da lógica auxiliar.

Embora em muitos casos a linguagem inicial e a linguagem de trabalho de um sistema de tablôs sejam as mesmas, como em alguns sistemas da Lógica Clássica, há muitas situações que originam diferenças entre estas linguagens:

- Existem lógicas cuja linguagem própria não permite uma análise semântica, tal como a Lógica Positiva Clássica; neste caso, a linguagem inicial pode ser a da Lógica Positiva Clássica, mas a linguagem de trabalho deve ser a da Lógica Clássica, cujo alfabeto é obtido acrescentando o conectivo de negação ao alfabeto da Lógica Positiva Clássica.
- Em alguns casos é conveniente associar na linguagem de trabalho cada fórmula da linguagem inicial com o seu valor veritativo, assim como faz [24].
- Para algumas Lógicas Modais um mundo pode ser associado a uma constante especial na linguagem de trabalho. Neste caso, se alguém quiser dizer que uma dada fórmula P da linguagem inicial é verdadeira em um mundo w, isso pode ser expressado na linguagem de trabalho como algo do tipo '$w(P)$' .

2.1 Definições.
Um *tablô em uma linguagem* L' é uma árvore na qual cada nó contém uma fórmula de L' e um dos valores booleanos v (verdadeiro) ou f (falso). Um *ramo em uma linguagem* L' é uma sequência de nós satisfazendo a mesma condição anterior. Um nó é dito *marcado* se o seu valor booleano é v.

Para nós, um nó marcado é simplesmente aquele que possui fisicamente uma marca. Do ponto de vista algorítmico, é mais simples trabalhar com um nó marcado do que verificar se ele foi usado[2].

2.2 Notação. Usaremos, graficamente, o sinal de *ticagem* "$\surd$" significando o valor 1 da marca, e sua ausência como o valor 0 da marca.

2.3 Notação.
- Dado um nó η de um tablô em L', $\mathrm{form}(\eta)$ é uma fórmula de η.
- Dado um ramo ρ em L', $\mathrm{coll}(\rho)$ é a coleção de fórmulas em ρ.

2.4 Definição. Um tablô é dito *finito* se ele possui um número finito de nós, caso contrário ele é dito *infinito*.

[2]Veja definição de nó usado em [2].

2.5 Definição.
Uma *função de inicialização de* L *em* L' é uma função que associa cada fórmula de L a um tablô em L'.

No método da confutação o tableau inicial tem, em muitos casos, apenas um nó, o qual contém a negação da suposta tese.

2.6 Definição.
Um *critério de fechamento* em L' é uma função que associa cada coleção Γ de fórmulas em L' a um dos valores booleanos v ou f.

2.7 Definição.
Uma *regra* em L' é uma função que associa cada fórmula em L' e cada coleção finita de fórmulas em L' tendo essa fórmula uma coleção finita de tablôs finitos em L'.

2.8 Definição.
Um *sistema de tablôs* é uma quíntupla ordenada $\mathcal{S} = \langle L, L', \mathcal{I}, C, \mathcal{R}\rangle$, onde L e L' são linguagens formais (para duas lógicas), $\mathcal{I}$ é uma função de inicialização para L, C é um critério de fechamento em L' e $\mathcal{R}$ é uma coleção de regras em L'. L é a *linguagem inicial de* $\mathcal{S}$, L' é a *linguagem de trabalho em* $\mathcal{S}$, $\mathcal{I}$ é a *função de inicialização de* $\mathcal{S}$, C é o *critério de fechamento de* $\mathcal{S}$ e $\mathcal{R}$ é a *coleção de regras de* $\mathcal{S}$. Cada elemento de $\mathcal{R}$ é dito uma *regra de* $\mathcal{S}$.

2.9 Notação. Daqui em diante, $\mathcal{S} = \langle L, L', \mathcal{I}, C, \mathcal{R}\rangle$ é um sistema de tablôs.

2.10 Definição.
- Γ é uma *coleção de fórmulas em* $\mathcal{S}$ se cada $P \in \Gamma$ é uma fórmula em L'.
- Um *ramo em* $\mathcal{S}$ é um ramo cuja coleção de fórmulas está em $\mathcal{S}$.
- Um *tablô em* $\mathcal{S}$ é um tablô em L'.

2.11 Definição.
- Uma *coleção* Γ *de fórmulas em* L' é dita *fechada em* $\mathcal{S}$ se $C(\Gamma) = \mathrm{v}$, caso contrário ela é dita *aberta em* $\mathcal{S}$.
- Um *ramo em* L' é dito *fechado em* $\mathcal{S}$ se a sua coleção de fórmulas é fechada em $\mathcal{S}$, caso contrário ele é dito *aberto em* $\mathcal{S}$.
- Um *tablô em* L' é dito *fechado em* $\mathcal{S}$ se todos os seus ramos são fechados em $\mathcal{S}$, caso contrário ele é dito *aberto em* $\mathcal{S}$.

2.12 Definição.

- Um ramo em $\mathcal{S}$ é dito *$\mathcal{L}'$-satisfatível* se a sua coleção de fórmulas é $\mathcal{L}'$-satisfatível.
- Um tablô em $\mathcal{S}$ é dito *$\mathcal{L}'$-satisfatível* se este possuir pelo menos um ramo $\mathcal{L}'$-satisfatível.

2.13 Definição.

Seja r e P uma regra em L' e uma fórmula em L', respectivamente.

- r é dita ser *aplicável à fórmula P* se, para qualquer coleção de fórmulas Γ em L' tal que $P \in \Gamma$, $\langle P, \Gamma\rangle$ pertence ao domínio de r.
- P é dito ser *fórmula excluída em $\mathcal{S}$* se não houver nenhuma regra em $\mathcal{S}$ aplicável a P.

2.14 Definição.

Seja Γ uma coleção de fórmulas em L'. Γ é dita *exaurida em $\mathcal{S}$* se as seguintes condições são satisfeitas:

- Γ é aberta em $\mathcal{S}$.
- Para cada $P \in \Gamma$ e para cada regra r de $\mathcal{S}$, se r é aplicável a P, então existe um ramo σ em $r(P, \Gamma)$ tal que $\mathrm{coll}(\sigma) \subseteq \Gamma$.

Um ramo em L' é dito *exaurido em $\mathcal{S}$* se a sua coleção de fórmulas é exaurida em $\mathcal{S}$.

Um tablô em L' é dito *exaurido em $\mathcal{S}$* se ele possuir pelo menos um ramo exaurido em $\mathcal{S}$.

2.15 Escólio.

- *Um ramo exaurido em $\mathcal{S}$ é um ramo aberto em $\mathcal{S}$.*
- *Um tablô exaurido em $\mathcal{S}$ é um tablô aberto em $\mathcal{S}$.*

2.16 Definição.

- Para um dado $n \in \mathbb{N}$, um *n-segmento* é o conjunto dos números naturais j tal que $0 \leq j \leq n$.
- Um *segmento* é um n-segmento, para algum $n \in \mathbb{N}$.

2.17 Definição.

- Dada uma fórmula P em L, $\mathcal{I}(P)$ é dito o *tablô inicial para P em $\mathcal{S}$*.
- Um *tablô inicial em $\mathcal{S}$* é $\mathcal{I}(P)$, para alguma fórmula P em L.

2.18 Definição.

Um tablô T' em L' é dito *um sucessor do tablô T em L'* se existe um nó η não marcado em T e uma regra r em $\mathcal{S}$ aplicável a $\mathrm{form}(\eta)$ tal que T' é obtido de T pela marcação de η e adicionando $r(\mathrm{form}(\eta), \mathrm{coll}(\rho))$ em cada ramo aberto ρ de T onde η ocorre. Neste caso, T é dito um *predecessor* de T' *em $\mathcal{S}$*.

2.19 Definição.
Uma *sequência de desenvolvimento em $\mathcal{S}$* é dita uma sequência $(T_j)_{j\in J}$ de tablôs em $\mathcal{S}$ tal que:

- J é um segmento ou $J = \mathbb{N}$.
- Para cada $j \in J$, se $j+1 \in J$, então T_{j+1} é um sucessor de T_j em $\mathcal{S}$.

2.20 Definição.
Seja P uma fórmula em L. $(T_j)_{j\in J}$ é dito ser uma *sequência de desenvolvimento para P em $\mathcal{S}$* se esta é uma sequência de desenvolvimento em $\mathcal{S}$ e T_0 é o tablô inicial para P em $\mathcal{S}$.

2.21 Definição.
Uma *sequência de desenvolvimento em $\mathcal{S}$* é chamada *completa em $\mathcal{S}$* se esta satisfaz uma das seguintes condições:

- É finita e termina com um tablô fechado em $\mathcal{S}$ ou com um tablô exaurido em $\mathcal{S}$.
- É infinita e tende a um tablô exaurido infinito em $\mathcal{S}$.

2.22 Definição.

- Um tablô T' em L' é dito ser um *desenvolvimento em $\mathcal{S}$ de um tablô T em L'* se existe uma sequência de desenvolvimento $(T_j)_{j\in J}$ e dois números j e k pertencentes a J tais que $j \leq k$, $T = T_j$ e $T' = T_k$.
- Seja P uma fórmula em $\mathcal{L}$. Um tablô T em $\mathcal{S}$ é dito ser um *tablô para P em $\mathcal{S}$* se T é um desenvolvimento do tablô inicial para P em $\mathcal{S}$.

2.23 Definição.
$\mathcal{S}$ é dito *progressivo em $\mathcal{L}'$* se, para quaisquer tablôs T e T' em $\mathcal{S}$ tal que T' é sucessor de T em $\mathcal{S}$, se T é $\mathcal{L}'$-satisfatível, então T' é $\mathcal{L}'$-satisfatível.

2.24 Definição.
Seja r uma regra em L'. r é dita *correta em $\mathcal{L}'$* se, para qualquer fórmula P em L' tal que r é aplicável a P e para qualquer coleção finita de fórmulas ϑ em L' tal que $P \in \vartheta$, se ϑ é $\mathcal{L}'$-satisfatível, então existe um ramo ρ de $r(P,\vartheta)$, tal que $\vartheta \cup \mathrm{coll}(\rho)$ é $\mathcal{L}'$-satisfatível.

2.25 Lema. *Se toda regra de $\mathcal{S}$ é correta em $\mathcal{L}'$, então $\mathcal{S}$ é $\mathcal{L}'$-progressivo.*

Prova:
Sejam T e T' tablôs em $\mathcal{S}$ tal que T' é sucessor de T em $\mathcal{S}$ e T é $\mathcal{L}'$-satisfatível. Então existe um nó η não marcado em T e uma regra r em $\mathcal{S}$ aplicável a $\mathrm{form}(\eta)$ tal que T' é obtido de T pela marcação de η e adicionando $r(\mathrm{form}(\eta), \mathrm{coll}(\rho))$ em cada ramo aberto ρ de T onde η ocorre. Sendo T $\mathcal{L}'$-satisfatível, existe um ramo ρ de T que é $\mathcal{L}'$-satisfatível. Se η não está em ρ, então ρ é um ramo de T', daí T' é $\mathcal{L}'$-satisfatível. Considere

agora que η está em ρ. Como r é correto em $\mathcal{L}'$ e $\mathrm{coll}(\rho)$ é $\mathcal{L}'$-satisfatível, temos que existe um ramo σ de $r(\mathrm{form}(\eta), \mathrm{coll}(\rho))$ tal que $\mathrm{coll}(\rho) \cup \mathrm{coll}(\sigma)$ é $\mathcal{L}'$-satisfatível. Como $\mathrm{coll}(\rho) \cup \mathrm{coll}(\sigma)$ é a coleção de fórmulas de um ramo ρ' de T' e este ramo é $\mathcal{L}'$-satisfatível, temos que T' é $\mathcal{L}'$-satisfatível. $\square$

2.26 Lema. *Para cada fórmula P de L, existe uma sequência de desenvolvimento completa para P em $\mathcal{S}$.*

2.27 Definição.

- $\mathcal{S}$ é dito *correto com respeito a $\mathcal{L}$* se, para qualquer fórmula P em $\mathcal{L}$, a existência de um tablô fechado para P em $\mathcal{S}$ implica na $\mathcal{L}$-validade de P.
- $\mathcal{S}$ é dito *completo com respeito a $\mathcal{L}$* se, para qualquer fórmula P em $\mathcal{L}$, a $\mathcal{L}$-validade de P implica na existência de um tablô fechado para P em $\mathcal{S}$.

A próxima seção apresenta as condições gerais de correção e completude que um sistema de tablôs deve cumprir a fim de ser correto e completo com respeito a uma dada lógica, portanto essas considerações não se limitam aqui a uma lógica particular.

§3. Condições Gerais de Correção e Completude para Sistemas de Tablôs

3.1 Definição.

- Um ramo em $\mathcal{S}$ é dito *$\mathcal{L}'$-satisfatível* se a sua coleção de fórmulas é $\mathcal{L}'$-satisfatível, caso contrário este ramo é dito *$\mathcal{L}'$-insatisfatível*.
- Um tablô em S é dito *$\mathcal{L}'$-satisfatível* se ele possuir pelo menos um ramo $\mathcal{L}'$-satisfatível, caso contrário este tablô é dito $\mathcal{L}'$-insatisfatível.

3.2 Teorema. ***(das condições de correção)***

Se $\begin{cases} * & \textit{para qualquer fórmula } P \textit{ em } \mathcal{L}, \textit{ a } \mathcal{L}'\textit{-insatisfabilidade do tablô} \\ & \textit{inicial para } P \textit{ em } \mathcal{S} \textit{ implica na } \mathcal{L}\textit{-validade de } P, \\ * & \textit{toda regra de } S \textit{ é correta em } \mathcal{L}', \\ * & \textit{toda coleção fechada de fórmulas em } \mathcal{S} \textit{ é } \mathcal{L}'\textit{-insatisfatível,} \end{cases}$

então $\mathcal{S}$ é correto com respeito a $\mathcal{L}$.

Prova:

Sejam P e T respectivamente uma fórmula em $\mathcal{L}$ e um tablô fechado para P em $\mathcal{S}$. Então existe uma sequência de desenvolvimento $(T_j)_{j \in \{0,\ldots,n\}}$ tal que T_0 é o tablô inicial para P em S e T_n é um tablô fechado em S tal que $T = T_n$. Suponha por absurdo que T_0 é $\mathcal{L}'$-satisfatível. Suponha também, por hipótese de indução, que, para um dado número natural $i < n$, T_i é $\mathcal{L}'$-satisfatível. Pela segunda condição de correção e pelo lema 2.25, temos que T_{i+1} também é $\mathcal{L}'$-satisfatível. Logo, para todo $i \in \{0, \ldots, n\}$, temos que T_i é $\mathcal{L}'$-satisfatível, e daí, em particular, T_n é $\mathcal{L}'$-satisfatível, ou seja, T é $\mathcal{L}'$-satisfatível, donde T possui um ramo $\mathcal{L}'$-satisfatível. Sendo ϕ a coleção de fórmulas deste ramo, temos que ϕ é $\mathcal{L}'$-satisfatível. Por outro lado, como T é fechado em $\mathcal{S}$, temos que ρ é fechado em $\mathcal{S}$, e daí, pela terceira condição de correção, temos que ϕ é $\mathcal{L}'$-insatisfatível, o que é absurdo, logo T_0 é $\mathcal{L}'$-insatisfatível, e portanto, pela primeira condição de correção, concluímos que P é uma fórmula $\mathcal{L}$-válida. □

3.3 Teorema. ***(das condições de completude)***

Se $\begin{cases} * \text{ para qualquer fórmula } P \text{ em } \mathcal{L}, \text{ a } \mathcal{L}\text{-validade de } P \text{ implica na} \\ \quad \mathcal{L}'\text{-insatisfabilidade do tablô inicial para } P \text{ em } \mathcal{S}, \\ * \text{ toda coleção exaurida de fórmulas em } \mathcal{S} \text{ é } \mathcal{L}'\text{-satisfatível,} \end{cases}$

então $\mathcal{S}$ é completo com respeito a $\mathcal{L}$.

Prova:

Seja P uma fórmula $\mathcal{L}$-válida. Pelo lema 2.26, existe uma sequência $(T_j)_{j\in J}$ de desenvolvimento completa para P em $\mathcal{S}$. Suponha por absurdo que esta sequência não termina com um tablô fechado para P. Duas situações daí podem ocorrer:

(i) $(T_j)_{j\in J}$ termina com um tablô exaurido em $\mathcal{S}$.

(ii) $J = \mathbb{N}$ e $(T_j)_{j\in J}$ tende para um tablô infinito exaurido em $\mathcal{S}$.

Em ambos os casos, temos que T_0 está contido em um tablô exaurido em $\mathcal{S}$. Seja τ o ramo deste tablô exaurido em $\mathcal{S}$ e seja σ o ramo deste tablô inicial contido em τ. Pela segunda condição de completude, temos que τ é $\mathcal{L}'$-satisfatível, logo σ também é $\mathcal{L}'$-satisfatível, donde, pela primeira condição de completude, temos que P não é uma fórmula $\mathcal{L}$-válida, o que é absurdo, e daí, necessariamente, temos que esta sequência termina com um tablô fechado para P. □

Capítulo 9

Lógica Proposicional Clássica com Indução

Este capítulo apresenta resultados da Lógica Proposicional Clássica relacionados a conceitos de indução e teoria dos conjuntos, apresentados nos capítulo 6 e 2, respectivamente.

Neste capítulo nós falaremos somente da Lógica Proposicional Clássica, assim, para dizer que P é consequência de Γ em LPC, notaremos isto por $\Gamma \vdash P$.

§1. Conceitos Gerais

A definição a seguir estabelece uma aplicação do Princípio da Indução Estrutural Funcional na demonstração de propriedades das fórmulas de LPC. Utilizando a base e o passo de indução conseguimos provar que para toda fórmula P, a propriedade $\Delta(P)$ é verdadeira.

1.1. (**Indução sobre fórmulas em** LPC)
É um caso particular do Princípio da Indução Estrutural Funcional.
Seja $\Delta(P)$ uma propriedade sobre fórmulas em LPC.

$$\text{Se } \begin{cases} \text{para cada letra sentencial } p,\ \Delta(p), \\ \text{para cada fórmula } P \text{ em LPC, } \Delta(P) \text{ implica em } \Delta(\neg P), \\ \text{para cada fórmula } P \text{ e } Q \text{ em LPC, } \Delta(P) \text{ e } \Delta(Q) \text{ implica em } \Delta(P \to Q), \\ \text{para qualquer fórmula } P \text{ e } Q \text{ em LPC, } \Delta(P) \text{ e } \Delta(Q) \text{ implica em } \Delta(P \wedge Q), \\ \text{para qualquer fórmula } P \text{ e } Q \text{ em LPC, } \Delta(P) \text{ e } \Delta(Q) \text{ implica em } \Delta(P \vee Q), \end{cases}$$

então, para qualquer fórmula P em LPC, $\Delta(P)$.

1.2 Exemplo.
Seja $\Delta(Q)$ a propriedade "$P_1 \leftrightarrow P_2 \vdash_{\mathrm{LPC}} Q(S\|P_1) \leftrightarrow Q(S\|P_2)$".
Se $S = Q$, então $\begin{cases} Q(S\|P_1) = P_1, \\ Q(S\|P_2) = P_2, \end{cases}$ daí vale $\Delta(Q)$, em virtude de Ref.
Se S não ocorre em Q, então $\begin{cases} Q(S\|P_1) = Q, \\ Q(S\|P_2) = Q, \end{cases}$ daí, por RE, temos que $\vdash Q \leftrightarrow Q$, ou seja, $\vdash Q(S\|P_1) \leftrightarrow Q(S\|P_2)$, donde, por Mon, $P_1 \leftrightarrow P_2 \vdash Q(S\|P_1) \leftrightarrow Q(S\|P_2)$, logo vale $\Delta(Q)$.
Podemos daí, sem perder a generalidade, dizer que $S \neq Q$ e S ocorre em Q.

- O caso em que Q é letra sentencial já foi considerado, pois, neste caso, $S = Q$ ou S não ocorre em Q.
- Caso Q é $\neg R$.
  Por hi, $P_1 \leftrightarrow P_2 \vdash R(S\|P_1) \leftrightarrow R(S\|P_2)$.
  Por LSC, $R(S\|P_1) \leftrightarrow R(S\|P_2) \vdash Q(S\|P_1) \leftrightarrow Q(S\|P_2)$ e daí, por Tran, $P_1 \leftrightarrow P_2 \vdash Q(S\|P_1) \leftrightarrow Q(S\|P_2)$.
- Caso Q é $R_1 \# R_2$, onde $\# \in \{\rightarrow, \wedge, \vee\}$.
  Por hi, suponha que $\begin{cases} P_1 \leftrightarrow P_2 \vdash R_1(S\|P_1) \leftrightarrow R_1(S\|P_2), \text{(1)} \\ P_1 \leftrightarrow P_2 \vdash R_2(S\|P_1) \leftrightarrow R_2(S\|P_2). \text{(2)} \end{cases}$
  Por LSC (viii, ix e x), temos que
  $R_1(S\|P_1) \leftrightarrow R_1(S\|P_2), R_2(S\|P_1) \leftrightarrow R_2(S\|P_2) \vdash$
  $(R_1(S\|P_1)\#R_2(S\|P_1)) \leftrightarrow R_1(S\|P_2)\#R_2(S\|P_2)$, ou seja,
  $R_1(S\|P_1) \leftrightarrow R_1(S\|P_2), R_2(S\|P_1) \leftrightarrow R_2(S\|P_2) \vdash Q(S\|P_1) \leftrightarrow Q(S\|P_2)$.
  (3)
  De (1), (2), (3) e Tran, temos que, $P_1 \leftrightarrow P_2 \vdash Q(S\|P_1)\#Q(S\|P_2)$.
  Portanto, aplicando indução sobre fórmulas em LPC, temos que, para toda fórmula Q em LPC, vale $P_1 \leftrightarrow P_2 \vdash Q(S\|P_1) \leftrightarrow Q(S\|P_2)$.

O exemplo a seguir estabelece uma aplicação do Princípio da Indução Estrutural Relacional na demonstração de propriedades sobre sequentes corretos em LPC.

1.3 Exemplo. As seguintes regras derivadas de LPC são válidas:

(i) Se $\Gamma \vdash P$, então existe Γ' finito tal que $\Gamma' \subseteq \Gamma$ e $\Gamma' \vdash P$ (lei da compacidade).
  Seja $\Delta(\Gamma \vdash P)$ a propriedade 'existe Γ' finito tal que $\Gamma' \subseteq P$ e $\Gamma' \vdash P$'.
  - No esquema da reflexividade.
    Considere que $P \in \Gamma$.
    Temos que '$\Gamma \vdash P$' é um exemplar de Ref.

$\{P\}$ é finito, $\{P\} \subseteq \Gamma$ e $\{P\} \vdash P$ é um exemplar de Ref, donde $\{P\} \vdash_{\mathbf{LPC}} P$, logo $\Delta(\Gamma \vdash P)$.

- No esquema modus ponens.
  Temos que '$P, P \to Q \vdash Q$' é um exemplar de MP.
  $\{P, P \to Q\}$ é finito e $\{P, P \to Q\} \subseteq \{P, P \to Q\}$, daí não há mais o que fazer, ou seja, $\Delta(P, P \to Q \vdash Q)$[1].
- Na regra da transitividade.
  Temos que $\frac{\Gamma \vdash P_1, \ldots, \Gamma \vdash P_n, \{P_1, \ldots, P_n\} \vdash Q}{\Gamma \vdash Q}$ é uma aplicação de Tran.
  Considere que $\begin{cases} \Gamma \vdash P_1 \\ \vdots \\ \{P_1, \ldots, P_n\} \vdash Q \end{cases}$, ou seja, considere que os sequentes são corretos.
  A partir daí a propriedade é válida.
  Suponha por HI, que $\begin{cases} \Delta(\Gamma \vdash P_1), \\ \vdots \\ \Delta(\Gamma \vdash P_n), \\ \Delta(P_1, \ldots, P_n) \vdash Q, \end{cases}$ ou seja,
  $\begin{cases} \text{existe } \Gamma_1 \text{ finito, } \Gamma_1 \leq \Gamma \text{ tal que } \Gamma_1 \vdash P_1, \\ \vdots \\ \text{existe } \Gamma_n \text{ finito, } \Gamma_n \subseteq \Gamma \text{ tal que } \Gamma_n \vdash P_n. \end{cases}$
  Seja $\varphi = \Gamma_1 \cup \ldots \cup \Gamma_n$.
  Daí $\begin{cases} \Gamma_1 \subseteq \varphi, \\ \vdots \\ \Gamma_n \subseteq \varphi, \end{cases}$ logo, pela Mon $\begin{cases} \varphi \vdash P_1 \\ \vdots \\ \varphi \vdash P_n \end{cases}$, e daí com $\{P_1, \ldots, P_n\} \vdash Q$, donde, por Tran, $\varphi \vdash Q$.
  Logo, como φ é finito e $\varphi = \Gamma$, temos que $\Delta(\Gamma \vdash Q)$.
- Na regra da Monotonicidade
  Considere que $\Gamma \subseteq \Gamma''$.
  Daí $\frac{\Gamma \vdash P}{\Gamma'' \vdash P}$ é uma aplicação da Mon.
  Considere que $\Gamma \vdash P$ e por HI, $\Delta(\Gamma \vdash P)$, ou seja, existe Γ' finito

[1] O raciocínio é análogo nos exemplares dos seguintes esquemas, pois estes exemplares possuem um número finito de premissas: Esquema da Reflexividade, Modus Ponens, $\wedge$-int, $\wedge$-el, Prova por Casos.

tal que $\Gamma' \subseteq \Gamma$ e $\Gamma' \vdash P$.
Temos que $\Gamma'' \vdash P$.
Caso $\begin{cases} \Gamma' \subseteq \Gamma'', \\ \Gamma' \vdash P, \\ \Gamma' \text{ é finito,} \end{cases}$ temos que $\Delta(\Gamma'' \vdash P)$.

- Na regra da Dedução

  Temos que $\dfrac{\Gamma \cup \{P\} \vdash Q}{\Gamma \vdash P \to Q}$ é uma aplicação da RD.

  Considere $\Gamma \cup \{P\} \vdash Q$ e, por HI, $\Delta(\Gamma \cup \{P\} \vdash Q)$, ou seja, existe Γ' finito tal que $\Gamma' \subseteq \Gamma \cup \{P\}$ e $\Gamma' \vdash Q$. (1)
  Temos que $\Gamma \vdash P \to Q$.
  De (1) temos que $(\Gamma' - \{P\}) \cup \{P\} \vdash Q$, daí, por RD, temos que $\Gamma' - \{P\} \vdash P \to Q$.
  Como $\begin{cases} \Gamma' - \{P\} \text{ é finito,} \\ \Gamma' - \{P\} \subseteq \Gamma, \end{cases}$ temos que $\Delta(\Gamma \vdash P \to Q)$.

- Na regra do $\neg$-introdução

  Temos que $\dfrac{\Gamma \cup \{P\} \vdash Q \quad \Gamma \cup \{P\} \vdash \neg Q}{\Gamma \vdash \neg P}$ é uma aplicação de $\neg$-int.

  Considere que $\begin{cases} \Gamma \cup \{P\} \vdash Q, \\ \Gamma \cup \{P\} \vdash \neg Q \end{cases}$ e, por HI, $\begin{cases} \Delta(\Gamma \cup \{P\} \vdash Q), \\ \Delta(\Gamma \cup \{P\} \vdash \neg Q). \end{cases}$
  Daí $\begin{cases} \Gamma \vdash \neg P, \\ \text{existe } \Gamma_1 \subseteq \Gamma \cup \{P\} \text{ tal que } \Gamma_1 \text{ é finito e } \Gamma_1 \vdash Q, \\ \text{existe } \Gamma_2 \subseteq \Gamma \cup \{P\} \text{ tal que } \Gamma_2 \text{ é finito e } \Gamma_2 \vdash \neg Q. \end{cases}$
  Seja $\varphi = \Gamma_1 \cup \Gamma_2$.
  Daí $\begin{cases} \varphi \vdash Q, \\ \varphi \vdash \neg Q, \end{cases}$ donde $\begin{cases} (\varphi - \{P\}) \cup \{P\} \vdash Q, \\ (\varphi - \{P\}) \cup \{P\} \vdash \neg Q, \end{cases}$ logo, por $\neg$-int, $\varphi - \{P\} \vdash \neg P$. (1)
  Temos que φ é finito, daí $\varphi - \{P\}$ é finito. (2)
  Como $\varphi \subseteq \Gamma \cup \{P\}$, temos que $\varphi - \{P\} \subseteq \Gamma$. (3)
  De (1), (2) e (3), $\Delta(\Gamma \vdash \neg P)$.

- Na regra do $\neg$-eliminação

  Temos que $\dfrac{\Gamma \cup \{\neg P\} \vdash Q \quad \Gamma \cup \{\neg P\} \vdash \neg Q}{\Gamma \vdash P}$ é uma aplicação de $\neg$-el.

Considere que $\begin{cases} \Gamma \cup \{\neg P\} \vdash Q, \\ \Gamma \cup \{\neg P\} \vdash \neg Q \end{cases}$ e, por HI, $\begin{cases} \Delta(\Gamma \cup \{\neg P\} \vdash Q), \\ \Delta(\Gamma \cup \{\neg P\} \vdash \neg Q). \end{cases}$

Daí $\begin{cases} \Gamma \vdash P, \\ \text{existe } \Gamma_1 \subseteq \Gamma \cup \{\neg P\} \text{ tal que } \Gamma_1 \text{ é finito e } \Gamma_1 \vdash Q, \\ \text{existe } \Gamma_2 \subseteq \Gamma \cup \{\neg P\} \text{ tal que } \Gamma_2 \text{ é finito e } \Gamma_2 \vdash \neg Q. \end{cases}$

Seja $\varphi = \Gamma_1 \cup \Gamma_2$.

Daí $\begin{cases} \varphi \vdash Q, \\ \varphi \vdash \neg Q, \end{cases}$ donde $\begin{cases} (\varphi - \{\neg P\}) \cup \{\neg P\} \vdash Q, \\ (\varphi - \{\neg P\}) \cup \{\neg P\} \vdash \neg Q, \end{cases}$ logo, por $\neg$-el, $\varphi - \{\neg P\} \vdash P$. (1)

Temos que φ é finito, daí $\varphi - \{\neg P\}$ é finito. (2)

Como $\varphi \subseteq \Gamma \cup \{\neg P\}$, temos que $\varphi - \{\neg P\} \subseteq \Gamma$. (3)

De (1), (2) e (3), $\Delta(\Gamma \vdash P)$.

- Na regra do $\top$-int
  $\Gamma \vdash \top$. Observe que $\emptyset \vdash \top$.
- O resultado é análogo para as regras do $\wedge$-el, $\vee$-int, Prova por Casos, $\bot$-el,$\forall$-el e $\exists$-int.

(ii) Se $\begin{cases} \text{para todo } P \in \varphi, \Gamma \vdash P, \\ \varphi \vdash Q, \end{cases}$ então $\Gamma \vdash Q$ (regra da transitividade geral).

Prova:

Assuma a hipótese.

De $\varphi \vdash Q$, temos pela lei da compacidade, que existe Γ' finito, tal que $\Gamma' \subseteq \varphi$, e $\Gamma' \vdash Q$. (1)

Daí $\begin{cases} \text{para todo } P \in \Gamma', \Gamma \vdash P \text{ (2)} \\ \Gamma' \vdash Q \text{ (3)} \end{cases}$ então $\Gamma' = \phi$ ou existem $P_1, \ldots, P_n$ tal que $\Gamma' = \{P_1, \ldots, P_n\}$.

Se $\Gamma' = \phi$, caso $\Gamma' \subseteq \Gamma$, temos de (1) que $\Gamma \vdash Q$.

Se $\Gamma' = \{P_1, \ldots, P_n\}$, então, de (2) e (3), $\begin{cases} \Gamma \vdash P_1 \\ \vdots \\ \Gamma \vdash P_n \\ \{P_1, \ldots, P_n\} \vdash Q, \end{cases}$ daí pela Tran $\Gamma \vdash Q$. $\square$

(iii) Se $\left\{\begin{array}{l}\Gamma \vdash P_1 \\ \vdots \\ \Gamma \vdash P_n\end{array}\right\}$ (1) e $\varphi, P_1, \ldots, P_n \vdash Q$ (2), então $\Gamma, \varphi \vdash Q$ (regra do corte).

Prova:
Assuma a hipótese.
De (1) e Mon $\left\{\begin{array}{l}\Gamma, \varphi \vdash P_1 \\ \vdots \\ \Gamma, \varphi \vdash P_n\end{array}\right.$ e para todo $P \in \varphi, \Gamma, \varphi \vdash P$ (3).
De (3), (2) e lei da transitividade geral $\Gamma, \varphi \vdash Q$. $\square$

(iv) Se $\left\{\begin{array}{l}\text{para todo } P \in \Delta, \Gamma \vdash P \text{ (1) ,} \\ \varphi, \Delta \vdash Q \text{ (2),}\end{array}\right.$ então $\Gamma, \varphi \vdash Q$ (regra do corte geral).

Prova:
Assuma a hipótese.
De (2) e lei da compacidade, temos que existe Γ_0 finito,
$\Gamma_0 \subseteq \varphi \cup \Delta$ tal que $\Gamma_0 \vdash Q$. (3)
De (3) , $(\Gamma_0 \cap \varphi) \cup (\Gamma_0 \cap \Delta) \vdash Q$.
Se $\Gamma_0 \cap \Delta = \phi$, então $\Gamma_0 \cap \varphi \vdash Q$, daí, por Mon, $\square$

(v) Se P, P' e Q, Q' são pares de fórmulas contraditórias, então $\left\{\begin{array}{l}\Gamma, P \vdash Q \\ \Gamma, P \vdash Q'\end{array}\right.$, implicam que $\Gamma \vdash P'$.

Se existe uma aplicação da regra, e tanto as hipóteses quanto as propriedades para as hipóteses são válidas, consequentemente a conclusão é correta e existe a possibilidade de prová-la.

Em algumas provas e definições por indução, em LPC, é relevante considerar o número de ocorrências de conectivos em uma fórmula. Para isso, considere a seguinte definição:

1.4 Definição. O *grau de uma fórmula* P em LPC, notado por $\mathrm{gr}(P)$, é o número de ocorrências de conectivos em P.

§2. Cálculo de Sequentes

2.1. **Lei da Compacidade**

Se $\Gamma \mid\!\!-\, P$, então existe $\Gamma' \subseteq \Gamma$ tal que $\begin{cases} \Gamma' \text{ é finito,} \\ \Gamma' \mid\!\!-\, P\ . \end{cases}$

Prova:

Seja $\Delta(\Gamma \mid\!\!-\, P)$ a propriedade 'Se $\Gamma \mid\!\!-\, P$, então existe $\Gamma' \subseteq \Gamma$ tal que $\begin{cases} \Gamma' \text{ é finito,} \\ \Gamma' \mid\!\!-\, P\text{'}\ . \end{cases}$ $\square$

O *Esquema da Substituição para Equivalência*, dado logo abaixo, é uma generalização do Lema da Substituição para Conectivos. Antes de sua formulação, precisamos de um conceito sintático, a substituição de uma fórmula por uma fórmula em uma fórmula.

2.2 Definição.

As cláusulas abaixo especificam substituição de fórmulas por fórmulas:

- A *substituição de S por P em Q*, notada por $Q(S\|P)$, é a fórmula obtida de Q substituindo todas as ocorrências de S por P.
- A *substituição de S por P em Γ*, notada por $\Gamma(S\|P)$, é a coleção de fórmulas obtida de Γ substituindo todas as ocorrências de S por P.

2.3. **Esquema da Substituição da Equivalência**

$P_1 \leftrightarrow P_2 \mid\!\!-\, Q(S\|P_1) \leftrightarrow Q(S\|P_2)$.

2.4 Corolário.

Se $\begin{cases} \Gamma \mid\!\!-\, Q(S\|P_1), \\ \Gamma \mid\!\!-\, P_1 \leftrightarrow P_2, \end{cases}$ *então* $\Gamma \mid\!\!-\, Q(S\|P_2)$.

2.5 Corolário.

Se $\begin{cases} \Gamma \mid\!\!-\, R, \\ \Gamma \mid\!\!-\, P_1 \leftrightarrow P_2, \\ R' \textit{ é obtido de } R \textit{ substituindo algumas ocorrências de } P_1 \textit{ por } P_2, \end{cases}$

então $\Gamma \mid\!\!-\, R'$.

2.6 Corolário.

Se $\begin{cases} \Gamma \mid\!\!-\, Q(S\|P_2), \\ \Gamma \mid\!\!-\, P_1 \leftrightarrow P_2, \end{cases}$ *então* $\Gamma \mid\!\!-\, Q(S\|P_1$.

2.7. **Lei da Substituição da Equivalência em** LPC

$P_1 \leftrightarrow P_2 \mid\!\!-_{\mathrm{LPC}}\, Q(S\|P_1) \leftrightarrow Q(S\|P_2)$.

Prova:

Seja $\Delta(Q)$ a propriedade '$P_1 \leftrightarrow P_2 \mid\!\!-_{\mathrm{LPC}}\, Q(S\|P_1) \leftrightarrow Q(S\|P_2)$'.

- Caso S não ocorre em Q
  Então $Q(S\|P_1) = Q(S\|P_2) = Q$.
  Daí, como $\vdash_{\text{LPC}} Q \leftrightarrow Q$, daí $\vdash_{\text{LPC}} Q(S\|P_1) \leftrightarrow Q(S\|P_2)$, donde, por Mon, $P_1 \leftrightarrow P_2 \vdash_{\text{LPC}} Q(S\|P_1) \leftrightarrow Q(S\|P_2)$.
- Caso $S = Q$
  Então $\begin{cases} Q(S\|P_1) = P_1, \\ Q(S\|P_2) = P_2, \end{cases}$ e daí, com $P_1 \leftrightarrow P_2 \vdash_{\text{LPC}} P_1 \leftrightarrow P_2$, temos que $P_1 \leftrightarrow P_2 \vdash_{\text{LPC}} Q(S\|P_1) \leftrightarrow Q(S\|P_2)$.
  Podemos daí considerar, spg, que S ocorre em Q e $S \neq Q$.
- Caso Q é uma letra sentencial, temos que S não ocorre em Q ou $S = Q$, o que já foi considerado anteriormente.
- Caso Q é $\neg R$
  Então $\begin{cases} Q(S\|P_1) = \neg R(S\|P_1), \\ Q(S\|P_2) = \neg R(S\|P_2). \end{cases}$
  Por HI, $P_1 \leftrightarrow P_2 \vdash_{\text{LPC}} R(S\|P_1) \leftrightarrow R(S\|P_2$.
  Pelo LSC, $R(S\|P_1) \leftrightarrow R(S\|P_2) \vdash_{\text{LPC}} \neg R(S\|P_1) \leftrightarrow \neg R(S\|P_2)$, donde $P_1 \leftrightarrow P_2 \vdash_{\text{LPC}} (\neg R)(S\|P_1) \leftrightarrow (\neg R)(S\|P_2)$.
- Caso Q é $R_1\#R_2$
  Então $\begin{cases} Q(S\|P_1) = R_1(S\|P_1)\#R_2(S\|P_2), \\ Q(S\|P_2) = R_1(S\|P_2)\#R_2(S\|P_2). \end{cases}$
  Por HI, $P_1 \leftrightarrow P_2 \vdash_{\text{LPC}} R_1(S\|P_1) \leftrightarrow R_1(S\|P_2)$ (1).
  Por HI, $P_1 \leftrightarrow P_2 \vdash_{\text{LPC}} R_2(S\|P_1) \leftrightarrow R_2(S\|P_2)$ (2).
  De LSC, temos que $R_1(S\|P_1) \leftrightarrow R_1(S\|P_2), R_2(S\|P_1) \leftrightarrow R_2(S\|P_2) \vdash_{\text{LPC}} R_1(S\|P_1)\#R_2(S\|P_1) \leftrightarrow R_1(S\|P_2)\#R_2(S\|P_2)$, ou seja, $R_1(S\|P_1) \leftrightarrow R_1(S\|P_2), R_2(S\|P_1) \leftrightarrow R_2(S\|P_2) \vdash_{\text{LPC}} (R_1\#R_2)(S\|P_1) \leftrightarrow (R_1\#R_2)(S\|P_2)$. (3)
  De (1), (2), (3) e TRAN, $P_1 \leftrightarrow P_2 \vdash_{\text{LPC}} (R_1\#R_2)(S\|P_1) \leftrightarrow (R_1\#R_2)(S\|P_2)$.

$\square$

2.8. (**Lei da Substituição Uniforme**[2])
$\Gamma \vdash P$ implica que $\Gamma(p\|S) \vdash P(p\|S)$.

Os conceitos de *Dual* e *Conjugado* são exemplos de aplicação da indução sobre fórmulas.
Em [29], o autor define *Fórmula Dual* ou *Fórmulas Duais* como sendo uma certa fórmula P constituída apenas com átomos $(p, q, r, \ldots)$ e suas negações $(\neg p, \neg q, \neg r, \ldots)$, usando exclusivamente os conectivos $\wedge$ e $\vee$. Sendo assim, ele define *dual de* P a fórmula obtida através de P, substituindo $\wedge$ por $\vee$.

[2] A Lei da Substituição Uniforme também se prova via indução sobre sequentes.

Para a fórmula $P : p \vee (\neg q \wedge r)$, sua dual será $p \wedge (\neg q \vee r)$.

De forma geral nós apresentamos a seguinte definição para *dual*:

2.9 Definição. (Dual em LPC)
Para cada fórmula P em LPC, definimos a *fórmula dual de* P, P_d, pelas cláusulas abaixo:
- $p(t_1, \ldots, t_n)_d = p(t_1, \ldots, t_n)$
- $(\neg P)_d = \neg(P_d)$
- $(P \to Q)_d = \neg(P_d) \wedge Q_d$
- $(P \wedge Q)_d = P_d \vee Q_d$
- $(P \vee Q)_d = P_d \wedge Q_d$
- $(\top_d) = \bot$
- $(\bot_d) = \top$
- $(P \leftrightarrow Q)_d = P \mathbin{\bar{\vee}} Q$
- Γ_d é a coleção $\{P_d | P \in \Gamma\}$.

2.10 Definição. (Conjugado em LPC)
Dada uma fórmula P em LPC, definimos a *fórmula conjugada de* P, P_{cj}, substituindo cada letra sentencial em P juntamente com todos os sinais de negação que a precedem pela sua negação, se o número de sinais de negação for par, e pela própria letra sentencial, se o número de sinais de negação for ímpar.

O *conjugado de uma fórmula* é a negação dessa fórmula.
Para cada fórmula em LPC, definimos P_{cj} da seguinte forma: $p(t_1, \ldots, t_n)_{cj} = \neg p(t_1, \ldots, t_n)$ onde:
- $(\neg P)_{cj} = \neg(P_{cj})$
- $(P \# Q)_{cj} = P_{cj} \# Q_{cj}$, onde $\# \in \{\to, \wedge, \vee\}$
- Γ_{cj} é a coleção $\{P_{cj} | P \in \Gamma\}$.

Dizemos que a negação de uma fórmula equivale ao dual de sua conjugada, ou ao conjugado de sua dual. Pois, dada uma fórmula, o dual de sua conjugada e o conjugado de sua dual são iguais. Esta seria a *primeira Lei da Dualidade* apresentada abaixo.

2.11. **Lei da Dualidade**
$\vdash \neg P \leftrightarrow (P_{cj})_d$

2.12 Exemplo.

- $\vdash \neg(p \wedge q) \leftrightarrow \neg p \vee \neg q$.
- $\vdash \neg(p \to q) \leftrightarrow \neg\neg p \wedge \neg q$.

A *segunda Lei da Dualidade* diz que se duas fórmulas são equivalentes, então os seus duais também são equivalentes.

Prova da Lei da Dualidade:

- Caso P é $p(t_1, \ldots, t_n)$
  $p(t_1, \ldots, t_n)_{cj} = \neg p(t_1, \ldots, t_n)$
  $(p(t_1, \ldots, t_n)_{cj})_d = \neg p(t_1, \ldots, t_n)$
  $\neg p = \neg p(t_1, \ldots, t_n)$
  Logo, vale $\Delta(P)$.
- Caso P é $\neg Q$
  Suponha, por HI, que vale $\Delta(Q)$, ou seja, $\vdash \neg Q \leftrightarrow (Q_{cj})_d$. (1)
  $\neg P = \neg\neg Q$. (2)
  $P_{cj} = \neg Q_{cj}$.
  $(P_{cj})_d = \neg(Q_{cj})_d$. (3)
  De (1), $\vdash \neg\neg Q \leftrightarrow \neg(Q_{cj})_d$. (4)
  De (4), (2) e (3), $\vdash \neg P \leftrightarrow (P_{cj})_d$.
  Logo, para este caso, vale $\Delta(P)$.
- Caso P é $(Q \to R)$
  Suponha, por HI, que vale $\Delta(Q)$ e $\Delta(R)$, ou seja,
  $\begin{cases} \vdash \neg Q \leftrightarrow (Q_{cj})_d \text{ (1)} \\ \vdash \neg R \leftrightarrow (R_{cj})_d \text{ (2)} \end{cases}$
  Temos que $\vdash \neg P \leftrightarrow Q \wedge \neg R$. (3)
  De (1), $\vdash Q \leftrightarrow \neg(Q_{cj})_d$. (4)
  De (3), (4) e (2), $\vdash \neg P \leftrightarrow \neg(Q_{cj})_d \wedge (R_{cj})_d$. (5)
  $((Q \to R)_{cj})_d = (Q_{cj} \to R_{cj})_d = \neg(Q_{cj})_d \wedge (R_{cj})_d$. (6)
  De (5) e (6) $\vdash \neg P \leftrightarrow ((Q \to R)_{cj})_d$, ou seja $\vdash \neg P \leftrightarrow (P_{cj})_d$, logo vale $\Delta(P)$ para este caso.
- Caso P é $(Q \wedge R)$
  Suponha, por HI, que vale $\Delta(Q)$ e $\Delta(R)$, ou seja, temos que
  $\begin{cases} \vdash \neg Q \leftrightarrow (Q_{cj})_d \text{ (1)} \\ \vdash \neg R \leftrightarrow (R_{cj})_d \text{ (2)} \end{cases}$
  Temos que $\neg P \leftrightarrow \neg Q \vee \neg R$. (3)
  De (3), (2), (1), $\vdash \neg P \leftrightarrow (Q_{cj})_d \vee (R_{cj})_d$. (4)
  Mas $(Q_{cj})_d \vee (R_{cj})_d$ é igual a $((Q \vee R)_{cj})_d$ que é igual a $(Q_{cj} \vee R_{cj})_d$. (5)
  Mas $(Q_{cj} \vee R_{cj})_d = (Q_{cj} \wedge R_{cj})_d = ((Q \wedge R)_{cj})_d$. (6)
  De (3) e (6), $\vdash \neg P \leftrightarrow ((Q \wedge R)_{cj})_d$.
  Ou seja, $\vdash \neg P \leftrightarrow (P_{cj})_d$. Logo vale $\Delta(P)$.
- Caso P é $(Q \vee R)$
  Suponha, por HI, que vale $\Delta(Q)$ e $\Delta(R)$, ou seja, temos que

$\begin{cases} \vdash \neg Q \leftrightarrow (Q_{cj})_d \text{ (1)} \\ \vdash \neg R \leftrightarrow (R_{cj})_d \text{ (2)} \end{cases}$

Temos que $\neg P \leftrightarrow \neg Q \wedge \neg R$. (3)
De (3), (2), (1), $\vdash \neg P \leftrightarrow (Q_{cj})_d \wedge (R_{cj})_d$. (4)
Mas $(Q_{cj})_d \wedge (R_{cj})_d$ é igual a $((Q \wedge R)_{cj})_d$ que é igual a $(Q_{cj} \wedge R_{cj})_d$. (5)
Mas $(Q_{cj} \wedge R_{cj})_d = (Q_{cj} \vee R_{cj})_d = ((Q \vee R)_{cj})_d$. (6)
De (3) e (6), $\vdash \neg P \leftrightarrow ((Q \vee R)_{cj})_d$.
Ou seja, $\vdash \neg P \leftrightarrow (P_{cj})_d$. Logo vale $\Delta(P)$.

- Caso P é $\forall x Q$
  Por HI, $\Delta(Q)$, ou seja, $\vdash \neg Q \leftrightarrow (Q_{cj})_d$. (1)
  Temos que $\neg P = \neg \forall x Q$. (2)
  De (1) e (2) temos que $P_{cj} = \forall x Q_{cj}$.
  Mas $P_{cj} = \forall x Q_{cj} = (P_{cj})_d = \exists x (Q_{cj})_d$. (3)
  De (3) e (1), $\vdash \forall x(\neg Q \leftrightarrow (Q_{cj})_d)$. (4)
  De (4) e DFDQ(v), $\vdash \exists x \neg Q \leftrightarrow \exists x (Q_{cj})_d$. (5)
  De (5) e NU, $\vdash \neg \forall x Q \leftrightarrow \exists x (Q_{cj})_d$.
  Temos que $\vdash \neg P \leftrightarrow (P_{cj})_d$. Logo vale $\Delta(P)$.
- Caso P é $\forall x Q$
  Por HI, $\Delta(Q)$, ou seja, $\vdash \neg Q \leftrightarrow (Q_{cj})_d$. (1)
  Temos que $\neg P = \neg \forall x Q$. (2)
  De (1) e (2) temos que $P_{cj} = \forall x Q_{cj}$.
  Mas $P_{cj} = \forall x Q_{cj} = (P_{cj})_d = \exists x (Q_{cj})_d$. (3)
  De (3) e (1), $\vdash \forall x(\neg Q \leftrightarrow (Q_{cj})_d)$. (4)
  De (4) e DFDQ(v), $\vdash \exists x \neg Q \leftrightarrow \exists x (Q_{cj})_d$. (5)
  De (5) e NU, $\vdash \neg \forall x Q \leftrightarrow \exists x (Q_{cj})_d$.
  Temos que $\vdash \neg P \leftrightarrow (P_{cj})_d$. Logo vale $\Delta(P)$.

De forma geral, primeiro temos que provar que a propriedade, no caso é a Lei da Dualidade, vale para todas as fórmulas atômicas. Para uma negação temos que mostrar que se a propriedade vale para a fórmula negada então vale para a negação. Para a implicação temos que supor que se a propriedade vale para o antecedente e para o consequente da implicação, então a propriedade vale também para a implicação. Para uma conjunção deve-se mostrar que se a propriedade vale para o primeiro e segundo conjuntores, então ela vale para a conjunção. E assim por diante. $\square$

Apresentamos a motivação semântica da dualidade através da tabela veritativa da negação da fórmula P e a sua forma dual.

Tabela 9.1: *Tabela Veritativa da* Negação de P

P	**¬P**
v	f
f	v

Tabela 9.2: *Tabela Veritativa da forma dual da* Negação de P^*

¬P	$(\neg\mathbf{P})_d$
f	v
v	f

2.13 Exemplo. O seguinte caso também exemplifica os conceitos apresentados:

- $(P \wedge Q)_d = P \vee Q$
- $(P \wedge Q)_{cj} = \neg P \wedge \neg Q$
- $\neg((P \wedge Q)_{cj}) = \neg(\neg P \wedge \neg Q)$

Também observe este caso:

- $(\forall x P(x, y))_d = \exists x P(x, y)$
- $(\forall x P(x, y))_{cj} = \forall x \neg P(x, y)$
- $\neg((\forall x P(x, y))_{cj}) = \neg \forall x \neg P(x, y)$

As proposições abaixo generalizam os esquemas do $\wedge$-introdução, do $\wedge$-eliminação, do $\vee$-introdução e da prova por casos:

(i) $P_1, \ldots, P_n \vdash P_1 \wedge \ldots \wedge P_n$.
(ii) $P_1 \wedge \ldots \wedge P_n \vdash P_i$.
(iii) $P_i \vdash P_1 \vee \ldots \vee P_n$.
(iv) $P_1 \vee \ldots \vee P_n, P_1 \rightarrow R, \ldots, P_n \rightarrow R \vdash R$.

As proposições abaixo generalizam as leis do silogismo hipotético e da transitividade da equivalência:

(i) $P_1 \rightarrow P_2, \ldots, P_{n-1} \rightarrow P_n \vdash P_1 \rightarrow P_n$.
(ii) $P_1 \leftrightarrow P_2, \ldots, P_{n-1} \leftrightarrow P_n \vdash P_1 \leftrightarrow P_n$.

As proposições abaixo generalizam as leis do dilema construtivo e do dilema destrutivo[3]:

(i) $P_1 \vee \ldots \vee P_n, P_1 \rightarrow Q_1, \ldots, P_n \rightarrow Q_n \vdash Q_1 \vee \ldots \vee Q_n$.
(ii) $\neg Q_1 \vee \ldots \vee \neg Q_n, P_1 \rightarrow Q_1, \ldots, P_n \rightarrow Q_n \vdash \neg P_1 \vee \ldots \vee \neg P_n$.

A proposição abaixo generaliza a lei da divisão e conquista[4]

[3]Estas leis foram formuladas respectivamente em 4.3.55 e 4.3.56.
[4]Esta lei foi formulada em 4.3.57.

$R \to P_1 \vee \ldots \vee P_n, R \wedge P_1 \leftrightarrow P_1', \ldots, R \wedge P_n \leftrightarrow P_n', P_1' \vee \ldots \vee P_n' \leftrightarrow S \vdash R \leftrightarrow S.$

As proposições abaixo generalizam as leis de De Morgan[5]:

(i) $\vdash \neg(P_1 \vee \ldots \vee P_n) \leftrightarrow \neg P_1 \wedge \ldots \wedge \neg P_n.$
(ii) $\vdash \neg(P_1 \wedge \ldots \wedge P_n) \leftrightarrow \neg P_1 \vee \ldots \vee \neg P_n.$

As proposições abaixo generalizam a lei da Distributividade[6]:

(i) $P \wedge (Q_1 \vee \ldots \vee Q_n) \leftrightarrow (P \wedge Q_1) \vee \ldots \vee (P \wedge Q_n).$
(ii) $P \vee (Q_1 \wedge \ldots \wedge Q_n) \leftrightarrow (P \vee Q_1) \wedge \ldots \wedge (P \vee Q_n).$

As três proposições seguintes são equivalentes:

(i) $\Gamma, P_1, \ldots, P_n \vdash Q.$
(ii) $\Gamma \vdash P_1 \wedge \ldots \wedge P_n \to Q.$
(iii) $\Gamma \vdash P_1 \to \ldots \to P_n \to Q.$

A proposição abaixo generaliza a lei de Importação/Exportação[7]:

$\vdash (P_1 \to \ldots \to P_n \to Q) \leftrightarrow (P_1 \wedge \ldots \wedge P_n \to Q).$

2.14 Definição.
Definimos *positividade* e *negatividade de uma dada fórmula em outra fórmula em* LPC, de S em relação a Q, pelas cláusulas abaixo:

- Se S não ocorre em p, então S é positivo em p e S é negativo em p.
- Q é positivo em Q.
- S é positivo em Q, se e somente se S é negativo em $\neg Q$.
- S é negativo em Q, se e somente se S é positivo em $\neg Q$.
- S é positivo em $Q_1 \# Q_2$, se e somente se S é positivo em Q_1 e S é positivo em Q_2.
- S é negativo em $Q_1 \# Q_2$, se e somente se S é negativo em Q_1 e S é negativo em Q_2.
- S é positivo em $Q_1 \to Q_2$, se e somente se S é negativo em Q_1 e S é positivo em Q_2.
- S é negativo em $Q_1 \to Q_2$, se se somente se S é positivo em Q_1 e S é negativo em Q_2.

P é dito ser *estritamente positivo em* Q se P não ocorre em Q ou se P é positivo em Q e P não é negativo em Q.
P é dito ser *estritamente negativo em* Q se P não ocorre em Q ou se P é negativo em Q e P não é positivo em Q[8].

Considerando que S é estritamente positivo em Q, mostra-se que:

[5] As leis de De Morgan foram formuladas em 4.3.34.
[6] A lei da Distributividade foi formulada em 4.3.58.
[7] A lei de Importação/Exportação foi formulada em 4.3.59.
[8] Assim, se P não ocorrer em Q, então P é tanto estritamente positivo como estritamente negativo em Q; esta é a única situação em que estas duas propriedades podem concorrer.

(1) $\vdash Q(S\|\bot) \to Q(S\|P)$.
(2) $\vdash Q(S\|P) \to Q(S\|\top)$.
(3) $((P_1 \to P_2) \to Q(S\|P_1)) \to ((P_1 \to P_2) \to Q(S\|P_2))$.
(4) $(P_1 \to P_2) \wedge Q(S\|P_1) \to (P_1 \to P_2) \wedge Q(S\|P_2)$.

Considerando que S é estritamente negativo em Q, mostra-se que:

(1) $\vdash Q(S\|\ P) \to Q(S\|\bot)$.
(2) $\vdash Q(S\|\top) \to Q(S\|P)$.
(3) $((P_1 \to P_2) \to Q(S\|P_2)) \to ((P_1 \to P_2) \to Q(S\|P_1))$.
(4) $(P_1 \to P_2) \wedge Q(S\|P_2) \to (P_1 \to P_2) \wedge Q(S\|P_1)$.

§3. Correção e Completude do cálculo de sequentes de LPC com respeito à semântica de LPC

3.1 Correção do cálculo de sequentes de LPC com respeito à semântica de LPC

3.1 Teorema. ***(Correção)*** *Se $\Gamma \vdash_{\mathrm{LPC}} P$, então $\Gamma \models_{\mathrm{LPC}} P$.*

Vamos utilizar a indução sobre sequentes para provar a propriedade descrita pelo teorema acima. Desta forma, de acordo com o princípio 6.4.40, demonstramos que a propriedade vale para todos os elementos do conjunto inicial e que cada função/relação geradora, preserva a propriedade.

Para o caso proposicional as leis estruturais e as leis de introdução e eliminação de conectivos devem ser corretas. Para isso é necessário provar que todo esquema é correto e toda regra é correta.

Esquemas

De acordo com a definição apresentada em 3.2.12, podemos prosseguir com a demonstração.

(i) Queremos mostrar que o esquema da reflexividade é correto.
Consideramos um exemplar de Ref, que é um sequente da forma '$\Gamma \vdash P$ tal que $P \in \Gamma$'.
Dados P e Γ tal que $P \in \Gamma$, queremos mostrar que $\Gamma \models_{\mathrm{LPC}} P$.
Seja I uma LPC-interpretação para $\Gamma \cup \{P\}$.
Se I satisfaz Γ, então I satisfaz todas as fórmulas de Γ, mas como $P \in \Gamma$, temos que I também satisfaz P. Então Ref é correto.

(ii) Queremos mostrar que o esquema de modus ponens é correto.
Um exemplar arbitrário de MP é um sequente da forma '$P, P \to Q \vdash Q$'.

Queremos mostrar que $P, P \to Q \models_{\text{LPC}} Q$.
Seja I uma interpretação para este sequente tal que
$\begin{cases} I_V(P) = \text{v}, (1) \\ I_V(P \to Q) = \text{v}. (2) \end{cases}$
De (2), temos que $I_V(P) = \text{f}$ ou $I_V(Q) = \text{v}$, donde, de (1), temos que $I_V(Q) = \text{v}$.

(iii) Queremos mostrar que $\top$-introdução é correto.
Um exemplar de $\top$-introdução é da forma '$\Gamma \vdash \top$'.
Como, para qualquer LPC-interpretação, $I_V(\top) = \text{v}$, então qualquer interpretação para $\Gamma \cup \{\top\}$ que satisfaz Γ também satisfaz $\top$.
Logo, $\Gamma \models_{\text{LPC}} \top$.

(iv) Queremos mostrar que $\bot$-eliminação é correto.
Um exemplar de $\bot$-eliminação é da forma '$\Gamma \vdash \bot$'.
Como, para qualquer LPC-interpretação, $I_V(\bot) = \text{f}$, ou seja, não existe uma interpretação que satisfaz o $\bot$, então é vacuamente verdadeiro que qualquer exemplar de $\bot$ é correto.

Um raciocínio análogo pode ser aplicado aos demais esquemas: $\wedge$-int, $\wedge$-el, $\vee$-int e PC.

Regras

De forma similar, escolhe-se uma aplicação de cada regra e mostra-se que a propriedade é correta para todas as aplicações dessa regra.

Assuma a hipótese, se $\Gamma \vdash_{\text{LPC}} P$, então $\Gamma \models_{\text{LPC}} P$.

(i) Queremos mostrar que a regra da transitividade é correta.
Uma aplicação de Tran é da forma $\dfrac{\Gamma_1 \vdash P_1, \ldots, \Gamma_n \vdash P_n,\ \{P_1, \ldots, P_n\} \vdash Q}{\Gamma \vdash Q}$, onde $n \geq 1$.
Suponha, por HI, que $\begin{cases} \Gamma \models_{\text{LPC}} P_1, (1) \\ \vdots \\ \Gamma \models_{\text{LPC}} P_n, \\ \{P_1, \ldots, P_n\} \models_{\text{LPC}} Q. (2) \end{cases}$
Seja I uma LPC-interpretação para $\Gamma \cup \{Q\}$ que satisfaz Γ.
Mas as letras sentenciais em $\{P_1, \ldots, P_n\}$ podem não figurar na coleção $\Gamma \cup \{Q\}$ e assim, não serem interpretadas por I. Seja I' uma extensão de I tal que, para cada letra sentencial p figurando em $\{P_1, \ldots, P_n\}$ e não figurando em $\Gamma \cup \{Q\}$, $I'(p) = \text{v}$.
Temos que I' é uma interpretação para $\Gamma \cup \{P_1, \ldots, P_n, Q\}$.

Pelo lema 4.2.9, I' satisfaz Γ, donde, por (1), $\begin{cases} I'_V(P_1) = \text{v}, \\ \vdots \\ I'_V(P_n) = \text{v}, \end{cases}$ logo I' satisfaz $\{P_1, \ldots, P_n\}$, donde, de (2), $I'_V(Q) = \text{v}$, mas como I e I' coincidem em Q e pelo lema 4.2.9, $I_V(Q) = \text{v}$, ou seja, $\Gamma \models_{\overline{\text{LPC}}} Q$.
Como consideramos uma aplicação arbritrária, através de Gen, a regra da transitividade é correta.

(ii) Queremos mostrar que a regra da dedução é correta.
Uma aplicação da RD é da forma $\dfrac{\Gamma, P \vdash Q}{\Gamma \vdash P \to Q}$.
Suponha, por HI, que, $\Gamma, P \models_{\overline{\text{LPC}}} Q$. (1)
Seja I uma LPC-interpretação para $\Gamma \cup \{P \to Q\}$ que satisfaz Γ.
Se $I_V(P) = \text{v}$, então I satisfaz $\Gamma \cup \{P\}$, donde, de (1), $I_V(Q) = \text{v}$.
Pela IM mostramos que $I_V(P) = \text{f}$ ou $I_V(Q) = \text{v}$, ou seja, $I_V(P \to Q) = \text{v}$, logo $\Gamma \models_{\overline{\text{LPC}}} P \to Q$.

(iii) Queremos mostrar que $\neg$-int é correto.
Uma aplicação de $\neg$-int é da forma $\dfrac{\Gamma, P \vdash Q \quad \Gamma, P \vdash \neg Q}{\Gamma \vdash \neg P}$.
Suponha, por HI, que $\left.\begin{cases} \Gamma, P \models_{\overline{\text{LPC}}} Q, \\ \Gamma, P \models_{\overline{\text{LPC}}} \neg Q. \end{cases}\right\}$ (1)
Seja I uma LPC-interpretação para $\Gamma \cup \{\neg P\}$ que satisfaz Γ.
Seja I' uma extensão de I tal que, para cada letra sentencial p que figure em Q, mas não figure em $\Gamma \cup \{P\}$, $I'_V(p) = \text{v}$.
Pela lema 4.2.9, I e I' coincidem em Γ, então I' satisfaz Γ.
Se $I'_V(P) = \text{v}$, então, de (1) e $\begin{cases} I_V(Q) = \text{v} \\ I_V(Q) = \text{f} \end{cases}$, o que é absurdo, logo $I'_V(P) = \text{f}$, donde $I'_V(\neg P) = \text{v}$, daí, pelo lema 4.2.9, $I_V(\neg P) = \text{v}$, logo $\Gamma \models_{\overline{\text{LPC}}} \neg P$.

Como todo exemplar de esquema é correto e toda aplicação de regra é correta, então todo sequente correto no cálculo será correto semanticamente, ou seja, todo sequente correto no cálculo é um sequente correto na semântica de LPC.

3.2 Teorema. ***(Contraposição do Teorema da Correção)***
Se $\Gamma \not\models_{\overline{\textbf{LPC}}} P$, *então* $\Gamma \not\vdash_{\textbf{LPC}} P$.

3.3 Exemplo.
Mostre que '$\vdash (P \to Q) \leftrightarrow (Q \to P)$' não é um esquema correto no cálculo de sequentes para LPC.

Seja I uma LPC-interpretação tal que $I(p) = \text{f}$ e $I(q) = \text{v}$. Daí temos que $I_V(p \to Q) = \text{v}$ e $I_V(q \to p) = \text{f}$. Logo, $I_V((p \to q) \leftrightarrow (q \to p)) = \text{f}$. Dessa forma '$\models (p \to q) \leftrightarrow (q \to p)$' não é correto em LPC, logo $\not\models (p \to q) \leftrightarrow (q \to p)$, donde, pelo teorema da correção, $\not\vdash (p \to q) \leftrightarrow (q \to p)$, portanto o esquema não é correto em LPC.

3.2 Completude do cálculo de sequentes de LPC com respeito à semântica de LPC

Mostraremos aqui que o cálculo de LPC é completo com respeito à semântica de LPC.

3.4 Definição. φ é P-coleção de Henkin em L com respeito a LPC se φ é P-saturado em L com respeito a LPC.

3.5 Lema.
Se φ é P-coleção de Henkin em L com respeito a LPC*, então valem as seguintes condições:*

(i) *φ é $\neg$-consistente em* LPC.
(ii) *φ é $\neg$-completo em L com respeito a* LPC.
(iii) $\top \in \varphi$.
(iv) $\bot \notin \varphi$.
(v) *$(Q \to R) \in \varphi$, sss $Q \notin \varphi$ ou $R \in \varphi$.*
(vi) *$(Q \wedge R) \in \varphi$, sss $Q \in \varphi$ e $R \in \varphi$.*
(vii) *$(Q \vee R) \in \varphi$, sss $Q \in \varphi$ ou $R \in \varphi$.*

Prova:

- (i): Se Γ não fosse $\neg$-consistente, então existiria uma fórmula Q, tal que $\Gamma \vdash_{\mathcal{L}} Q$ e $\Gamma \vdash_{\mathcal{L}} \neg Q$.
  Mas a partir dessa contradição $\Gamma \vdash_{\mathcal{L}} P$, o que não pode acontecer, uma vez que Γ é P-saturado.
- (ii): Suponha que $\Gamma \not\vdash_{\mathcal{L}} Q$ e $\Gamma \not\vdash_{\mathcal{L}} \neg Q$.
  Se $\Gamma \not\vdash_{\mathcal{L}} Q$, então pelo lema 7.4.9, $\Gamma \vdash_{\mathcal{L}} Q \to P$. (1)
  E se $\Gamma \not\vdash_{\mathcal{L}} \neg Q$, então $\Gamma \vdash_{\mathcal{L}} \neg Q \to P$. (2)
  Seja $\mathcal{L}$ dotada de TEx $(Q \vee \neg Q)$. De (1), (2) e pela PC temos que $\Gamma \vdash_{\mathcal{L}} P$. Assim Γ não seria $\neg$-completo. Então Γ deve ser $\neg$-completo.
- (iii): Se $\Gamma \vdash_{\mathcal{L}} \top$, então $\top \in \Gamma$.
- (iv): Se $\Gamma \vdash_{\mathcal{L}} \bot$, como $\bot \vdash_{\mathcal{L}} P$, daí $\Gamma \vdash_{\mathcal{L}} P$, o que é absurdo, logo $\Gamma \not\vdash_{\mathcal{L}} \bot$ e $\bot \notin \Gamma$.

- (v): Suponha que $R \in \Gamma$, então $\Gamma \,|\!\!\frac{\;\;}{\mathcal{L}}\, R$. Por Mon, $\Gamma, Q \,|\!\!\frac{\;\;}{\mathcal{L}}\, R$ e por RD, $\Gamma \,|\!\!\frac{\;\;}{\mathcal{L}}\, Q \to R$.
- (vi): De $Q \in \Gamma$ e $R \in \Gamma$, por Ref temos que $\Gamma \,|\!\!\frac{\;\;}{\mathcal{L}}\, Q$ e $\Gamma \,|\!\!\frac{\;\;}{\mathcal{L}}\, R$. Por $\wedge$-int, $Q, R \,|\!\!\frac{\;\;}{\mathcal{L}}\, Q \wedge R$ e por Tran, $\Gamma \,|\!\!\frac{\;\;}{\mathcal{L}}\, Q \wedge R$.
- (vii): Suponha de $Q \in \Gamma$ que $\Gamma \,|\!\!\frac{\;\;}{\mathcal{L}}\, Q$, por Ref. Então $Q \,|\!\!\frac{\;\;}{\mathcal{L}}\, Q \vee R$ por $\vee$-int. Daí $\Gamma \,|\!\!\frac{\;\;}{\mathcal{L}}\, Q \vee R$.

$\square$

3.6 Definição.
Seja φ P-coleção de Henkin em L com respeito a LPC. Dizemos que I é a interpretação canônica para φ em L se $\begin{cases} \mathcal{D}(I) \text{ é a coleção de letras sentenciais de } L, \\ \text{para cada letra sentencial } p \in \mathcal{D}(I), \\ I(p) = \text{v sss } p \in \varphi. \end{cases}$

3.7 Lema.
Sejam $\begin{cases} \varphi \textit{ uma coleção de fórmulas de } L, \\ I \textit{ a } \text{LPC}\textit{-interpretação para } \varphi \textit{ em } L. \end{cases}$
Então, para cada fórmula Q de L, $Q \in \varphi$ sss $I_V(Q) = $ v.

Prova:
Para cada fórmula Q de L, seja $\Delta(Q)$ a conjunção das proposições $\begin{cases} Q \in \varphi \text{ implica que } I_V(p) = \text{Q}.\ Q \notin \varphi \text{ implica} \\ \text{que } I_V(Q) = \text{f}. \end{cases}$

- Caso Q é letra sentencial
  Se $Q \in \varphi$, por HI, $I_V(Q) = \text{v}$.
  Se $Q \notin \varphi$, por HI, $I_V(Q) = \text{f}$.
- Caso Q é $\top$ ou $\bot$.
  É imediato, pois toda interpretação, ao $\top$ atribui v, e ao $\bot$ atribui f.
  Então, $I_V(\top) = \text{v}$ e $I_V(\bot) = \text{f}$.
- Q é $\neg R$
  Se $\neg R \in \varphi$, então $R \notin \varphi$ porque φ é $\neg$-consistente, então, por HI, $I_V(R) = \text{f}$, donde $I_V(\neg R) = \text{v}$. Se $\neg R \notin \varphi$, então $R \in \varphi$, porque φ é $\neg$-completo, logo, por HI, $I_V(R) = \text{v}$, donde $I_V(\neg R) = \text{f}$.
- Caso Q é $R \to S$
  Se $(R \to S) \in \varphi$, então $R \notin \varphi$ ou $S \in \varphi$, daí $\neg R \in \varphi$ ou $S \in \varphi$, logo, por HI, $I_V(R) = \text{f}$ ou $I_V(S) = \text{v}$, logo $I_V(R \to S) = \text{v}$.
  Se $(R \to S) \notin \varphi$, então $R \in \varphi$ e $S \notin \varphi$, donde, $R \in \varphi$ e $\neg S \in \varphi$, logo, por HI, $I_V(R) = \text{v}$ e $I_V(S) = \text{f}$, logo $I_V(R \to S) = \text{f}$.

$\square$

3.8 Lema.
Seja Γ *uma coleção de fórmulas em* LPC.
Se $\Gamma \not\vdash_{\text{LPC}} P$*, então existe uma linguagem* L *para* LPC *e existe uma* P*-coleção de Henkin para* L *em* LPC *tal que* $\Gamma \subseteq \varphi$.

Prova:
Seja $\Omega = \{\Phi \mid \Phi$ é coleção de fórmulas de L e $\Phi \not\vdash_{\text{LPC}} P$ e $\Gamma \subseteq \varphi\}$.

(i) Queremos mostrar que, se $\mathcal{C}$ é cadeia em Ω, então $\bigcup \mathcal{C} \in \Omega$.
Prova: Suponha por absurdo que $\bigcup \mathcal{C} \vdash_{\text{LPC}} P$.
Pela regra da compacidade, existe um subconjunto finito $\{Q_1, \ldots, Q_n\}$ de $\bigcup \mathcal{C}$ tal que $Q_1, \ldots, Q_n \vdash_{\text{LPC}} P$, com $n \geq 1$.
Para cada $i \in \{1, \ldots, n\}$, existe $\Gamma_i \in \mathcal{C}$ tal que $Q_i \in \Gamma_i$.
Como $\mathcal{C}$ é cadeia, existe um $j \in \{1, \ldots, n\}$ tal que, para cada $i \in \{1, \ldots, n\}, \Gamma_i \subseteq \Gamma_j$, daí $\Gamma_j \vdash_{\text{LPC}} P$, o que é absurdo, pois $\Gamma_j \in \Omega$.
Logo, $\bigcup \mathcal{C} \not\vdash_{\text{LPC}} P$, portanto $\bigcup C \in \Omega$.

(ii) Queremos mostrar que, se $\mathcal{C}$ é cadeia em Ω, então $\mathcal{C}$ é limitado superiormente em Ω.
Por (i), temos que $\bigcup \mathcal{C} \in \Omega$. (1)
É óbvio que, para todo $\Phi \in \mathcal{C}$, $\Phi \subseteq \bigcup \mathcal{C}$. (2)
De (1) e (2) temos que $\mathcal{C}$ e limitado superiormente em Ω.

(iii) Ω possui um elemento máximo, pelo caso (ii) e Lema de Zorn.

(iv) Queremos mostrar a conclusão do lema.
Seja φ um elemento maximal de Ω.
Daí $\varphi \not\vdash_{\text{LPC}} P$ e $\Gamma \subseteq \varphi$.
Considere φ' uma coleção de fórmulas de L tal que $\varphi \subseteq \varphi'$ e $\varphi' \not\vdash_{\text{LPC}} P$.
Então $\varphi' \in \Omega$. Mas como φ é elemento maximal de Ω, temos que $\varphi' \subseteq \varphi$, logo $\varphi = \varphi'$. Daí φ é P-saturado em L com respeito a LPC.
Donde φ é P-coleção de Henkin para L em LPC.

$\square$

3.9 Teorema. ***(Completude)*** *Se* $\Gamma \models_{\text{LPC}} P$*, então* $\Gamma \vdash_{\text{LPC}} P$.

Prova:
É imediata pelo Teorema Geral da Completude apresentado em 7.3.2.
A primeira condição do Teorema Geral da Completude se verifica pelo lema 3.5.
A segunda condição se verifica pelo lema 3.7.
A terceira condição se verifica pelo lema 3.8. $\square$

§4. Um Sistema de Tablôs para LPC

Um *Sistema de Tablôs por Refutação para* LPC é definido nesta seção e será notado por STR.

Considere, para esta seção, os conceitos gerais sobre semântica de valorações apresentados em §3.3 e a semântica de valores para LPC descrita em §4.2.

4.1 Definição.
A *linguagem inicial* e a *linguagem de trabalho* em STR são iguais, e serão referenciadas, nesta seção, por L.

4.2 Definição.
A *função de inicialização* de STR associa uma fórmula P de L a um tablô com um único nó não marcado, consistindo da negação do teorema a ser demonstrado.

Considere o sequente $\{P_1, \ldots, P_n\} \vdash Q$. O seu tablô inicial será:

$$\begin{array}{c} P_1 \\ | \\ \vdots \\ | \\ P_n \\ | \\ \neg Q \end{array}$$

4.3 Definição. Um *ramo é fechado em* STR se ele possuir duas fórmulas contraditórias.

4.4 Teorema. $\Gamma \cup \{\neg P\}$ *é* LPC-*insatisfatível sss* $\Gamma \models_{\mathrm{LPC}} P$.

4.5 Definição.
Se todos os ramos fecham, então $\Gamma \cup \{\neg P\}$ *é insatisfatível*. Assim, $\Gamma \models_{\mathrm{LPC}} P$, ou seja, o sequente é correto.

4.6 Definição.
Se algum ramo não fecha, então $\Gamma \cup \{\neg P\}$ *é satisfatível*, ou seja, existe uma valoração ou interpretação que valida a forma negada e então o sequente é incorreto, ou seja, $\Gamma \not\models_{\mathrm{LPC}} P$.

A *coleção de regras de* STR possui 11 regras. O crescimento de um tablô é dado pela aplicação sucessiva das regras à nós ainda não *ticados*. A expansão se dá em todos os ramos abertos abaixo do nó utilizado. As regras aqui desenvolvidas possuem a característica de gerar até dois novos ramos.

As regras definidas abaixo devem ser aplicadas até que seja atingindo um tablô fechado ou um tablô exaurido aberto em STR, através de uma sequência de desenvolvimento completa em STR, a partir de um tablô inicial.

As definições das regras deste sistema estão relacionadas com os conectivos lógicos das fórmulas para as quais elas são aplicadas.

Todas as regras de STR sao definidas graficamente através do fornecimento das fórmulas dos nós que compõem os tablôs resultantes.

- Implicação $(P \rightarrow Q)$: Esta regra associa cada fórmula $P \rightarrow Q$ e cada coleção de fórmulas contendo $P \rightarrow Q$ a dois tablôs, cada um contendo um único nó não marcado, cujas fórmulas são respectivamente $\neg P$ e Q.

$$\begin{array}{c} (P \rightarrow Q)\surd \\ \diagup \quad \diagdown \\ \neg P \qquad Q \end{array}$$

Figura 9.1: Implicação

- Negação da Implicação $(\neg(P \rightarrow Q))$: Esta regra associa cada fórmula $\neg(P \rightarrow Q)$ e cada coleção de fórmulas contendo $\neg(P \rightarrow Q)$ a um único tablô com somente um ramo contendo dois nós não marcados, cujas fórmulas são respectivamente P e $\neg Q$.

$$\begin{array}{c} \neg(P \rightarrow Q)\surd \\ | \\ P \\ | \\ \neg Q \end{array}$$

Figura 9.2: Negação da Implicação

- Conjunção $(P \land Q)$: Esta regra associa cada fórmula $P \land Q$ e cada coleção de fórmulas contendo $P \land Q$ a um único tablô com somente um ramo contendo dois nós não marcados, cujas fórmulas são respectivamente P e Q.

$$\begin{array}{c} (P \land Q)\surd \\ | \\ P \\ | \\ Q \end{array}$$

Figura 9.3: Conjunção

- Negação da Conjunção $(\neg(P \land Q))$: Esta regra associa cada fórmula $\neg(P \land Q)$ e cada coleção de fórmulas contendo $\neg(P \land Q)$ a dois tablôs, cada um contendo um único nó não marcado, cujas fórmulas são respectivamente $\neg P$ e $\neg Q$.

$$\begin{array}{c} \neg(P \land Q)\surd \\ \diagup \quad \diagdown \\ \neg P \qquad \neg Q \end{array}$$

Figura 9.4: Negação da Conjunção

- Disjunção $(P \lor Q)$: Esta regra associa cada fórmula $P \lor Q$ e cada coleção de fórmulas contendo $P \lor Q$ a dois tablôs, cada um contendo um único nó não marcado, cujas fórmulas são respectivamente P e Q.

$$\begin{array}{c} (P \lor Q)\surd \\ \diagup \quad \diagdown \\ P \qquad Q \end{array}$$

Figura 9.5: Disjunção

- Negação da Disjunção $(\neg(P \vee Q))$: Esta regra associa cada fórmula $\neg(P \vee Q)$ e cada coleção de fórmulas contendo $\neg(P \vee Q)$ a um único tablô com somente um ramo contendo dois nós não marcados, cujas fórmulas são respectivamente $\neg P$ e $\neg Q$.

$$\begin{array}{c} \neg(P \vee Q)\surd \\ | \\ \neg P \\ | \\ \neg Q \end{array}$$

Figura 9.6: Negação da Disjunção

- Equivalência $(P \leftrightarrow Q)$: Esta regra associa cada fórmula $P \leftrightarrow Q$ e cada coleção de fórmulas contendo $P \leftrightarrow Q$ a dois tablôs, cada um contendo um ramo, ambos com dois nós não marcados, cujas fórmulas do primeiro são P e Q e as fórmulas do segundo são $\neg P$ e $\neg Q$.

$$\begin{array}{cc} \multicolumn{2}{c}{(P \leftrightarrow Q)\surd} \\ / & \backslash \\ P & \neg P \\ | & | \\ Q & \neg Q \end{array}$$

Figura 9.7: Equivalência

- Negação da Equivalência $(\neg(P \leftrightarrow Q))$: Esta regra associa cada fórmula $\neg(P \leftrightarrow Q)$ e cada coleção de fórmulas contendo $\neg(P \leftrightarrow Q)$ a dois tablôs, cada um contendo um ramo, ambos com dois nós não marcados, cujas fórmulas do primeiro são P e $\neg Q$ e as fórmulas do segundo são $\neg P$ e Q.

$$\begin{array}{c} \neg(P \leftrightarrow Q)\surd \\ \begin{array}{cc} / & \backslash \\ P & \neg P \\ | & | \\ \neg Q & Q \end{array} \end{array}$$

Figura 9.8: Negação da Equivalência

- Disjunção Exclusiva $(P \mathbin{\bar{\vee}} Q)$: Semelhante à regra para Negação de Equivalência.

$$\begin{array}{c} (P \mathbin{\bar{\vee}} Q)\surd \\ \begin{array}{cc} / & \backslash \\ P & \neg P \\ | & | \\ \neg Q & Q \end{array} \end{array}$$

Figura 9.9: Disjunção Exclusiva

- Negação da Disjunção Exclusiva $(\neg(P \mathbin{\bar{\vee}} Q))$: Semelhante à regra para Equivalência.

$$\begin{array}{c} \neg(P \mathbin{\bar{\vee}} Q)\surd \\ \begin{array}{cc} / & \backslash \\ P & \neg P \\ | & | \\ Q & \neg Q \end{array} \end{array}$$

Figura 9.10: Negação da Disjunção Exclusiva

- Negação da negação $(\neg\neg P)$: Esta regra associa cada fórmula $\neg\neg P$ e cada coleção de fórmulas contendo $\neg\neg P$ a um tablô com somente um nó não marcado cuja fórmula é P.

$$\begin{array}{c} \neg\neg(P)\surd \\ | \\ P \end{array}$$

Figura 9.11: Negação da negação

As regras anteriormente apresentadas são derivadas das seguintes tautologias[9]:

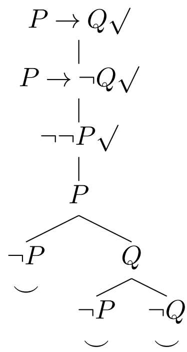
(i) $\vdash (P \to Q) \leftrightarrow \neg P \vee Q$,
(ii) $\vdash \neg(P \to Q) \leftrightarrow P \wedge Q$,
(iii) $\vdash (P \leftrightarrow Q) \leftrightarrow (P \wedge Q) \vee (\neg P \wedge \neg Q)$,
(iv) $\vdash \neg(P \leftrightarrow Q) \leftrightarrow (P \wedge \neg Q) \vee (\neg P \wedge Q)$,
(v) $\vdash (P \mathbin{\bar{\vee}} Q) \leftrightarrow (P \vee Q) \wedge \neg(P \wedge Q)$,
(vi) $\vdash (P \mathbin{\bar{\vee}} Q) \leftrightarrow \neg(P \leftrightarrow Q)$,
(vii) $\vdash \neg(P \mathbin{\bar{\vee}} Q) \leftrightarrow (P \leftrightarrow Q)$,
(viii) $\vdash (P \downarrow Q) \leftrightarrow \neg(P \vee Q)$.

4.1 Exemplos

Alguns exemplos são dados a fim de ilustrar como STR decide se uma fórmula é LPC-satisfatível ou LPC-insatisfatível.

(i) Sejam P, Q e R fórmulas atômicas distintas. Uma sequência de desenvolvimento em STR para a fórmula
$$S = P \to Q, P \to \neg Q \vdash \neg P$$
é apresentada na figura abaixo. Essa sequência termina com um tablô fechado em STR para a fórmula S, então S é LPC-insatisfatível, portanto $\vdash_{\text{LPC}} P \to Q, P \to \neg Q \vdash \neg P$.

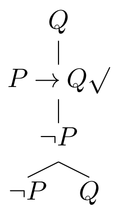
[9] Algumas são evidentes e o leitor pode facilmente percebê-las.



(ii) Sejam P, Q e R fórmulas atômicas distintas. Uma sequência de desenvolvimento em STR para a fórmula

$$S = Q, P \to Q \vdash P$$

é mostrada na figura abaixo. Essa sequência termina com um tablô exaurido em STR para a fórmula S, então S é LPC-satisfatível, portanto, não é verdade que $\vdash_{\mathrm{LPC}} Q, P \to Q \vdash P$.



Capítulo 10

Lógica Quantificacional Clássica com Indução

Este capítulo apresenta resultados relacionados à Lógica Quantificacional Clássica e que envolvem conceitos de indução e teoria dos conjuntos, apresentados nos capítulo 6 e 2, respectivamente.

Nesta seção nós falaremos somente da Lógica Quantificacional Clássica, assim, para dizer que P é consequência de Γ em LQC, notaremos isto por $\Gamma \vdash P$.

§1. Conceitos Gerais

1.1 Teorema. *(O valor veritativo de uma forma instanciada simplesmente)*
$I_V(P(x|t)) = I(x|I_D(t))_V(P)$.

1.2 Teorema. *(O valor veritativo de uma forma instanciada simultaneamente)*
$I_V P(x_1, \ldots, x_n|(t_1, \ldots, t_n)) = I(x_1, \ldots, x_n)|(I_D(t_1), \ldots, I_D(t_n))_V(P)$.

Instanciação Simultânea

Em diversas situações é importante considerar o número de ocorrências de conectivos e quantificadores em uma fórmula[1].

[1]Neste trabalho, o grau de uma fórmula é principalmente utilizado nas provas de correção e completude da Lógica Quantificacional Clássica.

1.3 Definição.
O *grau de uma fórmula* P em LQC, notado por $\mathrm{gr}\,(P)$, é o número de ocorrências de conectivos e quantificadores em P.

A instanciação simultânea de variáveis por termos é essencial para lidarmos com a quantificação simultânea, a qual será vista adiante.

1.4 Notação. De agora em diante adotamos as seguintes convenções, a menos que seja dito algo em contrário, considerando n e p números naturais (eventualmente nulos):

- $\overrightarrow{x}$ – uma lista $x_1, \ldots, x_n$ de variáveis.
- $\overrightarrow{y}$ – uma lista $y_1, \ldots, y_n$ de variáveis.
- $\overrightarrow{z}$ – uma lista $z_1, \ldots, z_p$ de variáveis.
- $\overrightarrow{w}$ – uma lista $w_1, \ldots, w_p$ de variáveis.
- $\overrightarrow{t}$ – uma lista $t_1, \ldots, t_n$ de termos.
- $\overrightarrow{u}$ – uma lista $u_1, \ldots, u_p$ de termos.
- $\overrightarrow{v}$ – uma lista $v_1, \ldots, v_r$ de termos.
- $\overrightarrow{D}$ – uma lista $D_1, \ldots, D_n$ de designadores.
- $\overrightarrow{E}$ – uma lista $E_1, \ldots, E_p$ de designadores.
- $\overrightarrow{\Psi}$ – uma lista $\Psi_1, \ldots, \Psi_n$ dos quantificadores $\forall$ ou $\exists$.
- $\overrightarrow{\Upsilon}$ – uma lista $\Upsilon_1, \ldots, \Upsilon_p$ dos quantificadores $\forall$ ou $\exists$.
- $\overrightarrow{\Psi'}$ – a lista $\Psi'_1, \ldots, \Psi'_p$ tal que, para cada $i \in \{1, \ldots, n\}$,
  $$\Psi'_i = \begin{cases} \exists, \text{ se } \Psi'_i = \forall, \\ \forall, \text{ se } \Psi'_i = \exists. \end{cases}$$
- $\overrightarrow{\Upsilon'}$ – a lista $\Upsilon'_1, \ldots, \Upsilon'_p$ definida analogamente à lista Ψ'.
- $\forall\overrightarrow{x}$ – a samblagem '$\forall x_1, \ldots, \forall x_n$'.
- $\exists\overrightarrow{x}$ – a samblagem '$\exists x_1, \ldots, \exists x_n$'.
- $\overrightarrow{\Psi}\,\overrightarrow{x}$ – a samblagem '$\Psi_1 x_1, \ldots, \Psi_n x_n$'.

1.5 Definições.

- $\overrightarrow{x}$ aceita $\overrightarrow{t}$ em $P \rightleftharpoons x_1, \ldots, x_n$ aceitam respectivamente $t_1, \ldots, t_n$ em P.
- $\overrightarrow{x}$ ocorre em $D \rightleftharpoons i \in \{1, \ldots, n\}$ tal que x_i ocorre em D.
- $\overrightarrow{x}$ é livre em $D \rightleftharpoons i \in \{1, \ldots, n\}$ tal que x_i é livre em D.
- $\overrightarrow{x}$ é livre em $(\overrightarrow{z})D \rightleftharpoons$ para cada $i \in \{1, \ldots, n\}$, existe $j \in \{1, \ldots, p\}$ tal que x_i é z_j ou x_i não é livre em D.[2]

[2]Esta definição serve para expressar de uma vez quando cada componente de uma lista de variáveis $\overrightarrow{x}$ coincide com algum componente de $\overrightarrow{z}$ ou é livre em D, o que é muito útil para a formulação de vários fatos sintáticos. Se D for uma fórmula P, isto pode ser expresso dizendo que $\overrightarrow{x}$ é livre em $\forall\overrightarrow{z}P$ ou, alternativamente, que $\overrightarrow{x}$ é livre em $\exists\overrightarrow{x}P$. Porém, se D for um termo, as samblagens $\forall\overrightarrow{x}P$ e $\exists\overrightarrow{x}P$ não são designadores. Note também que as listas $\overrightarrow{x}$ e $\overrightarrow{z}$ podem ser exatamente uma variável, ambas ou uma delas, daí podemos também dizer, por exemplo, que x é livre em $(z)D$ para expressar que $x = z$ ou x é livre em D.

- $\overrightarrow{x}$ ocorre em $\overrightarrow{E} \rightleftharpoons$ para algum $i \in \{1, \ldots, n\}$ e para algum $j \in \{1, \ldots, p\}, x_i$ ocorre em E_j.
- $\overrightarrow{x}$ ocorre em $\overrightarrow{E} \rightleftharpoons$ para algum $i \in \{1, \ldots, n\}$ e para algum $j \in \{1, \ldots, p\}, x_i$ é livre em E_j.
- $\overrightarrow{D}$ é (lista) própria $\rightleftharpoons$ para quaisquer $i, j \in \{1, \ldots, n\}$, se $i \neq j$, então $D_i \neq D_j$.
- $\overrightarrow{D}$ e $\overrightarrow{E}$ são (listas) disjuntas $\rightleftharpoons$ para cada $i \in \{1, \ldots, n\}$ e para cada $j \in \{1, \ldots, p\}, D_i \neq E_j$.
- $\overrightarrow{D}$ está contida em $\overrightarrow{E}$ $\rightleftharpoons$ para cada $i \in \{1, \ldots, n\}$, existe $j \in \{1, \ldots, p\}$ tal que $D_i \neq E_j$.

1.6 Definição.

- $\overrightarrow{D} - E \rightleftharpoons D_{i1}, \ldots, D_{ir}$, onde $i_1, \ldots\ i_r$ são os naturais $j \in \{1, \ldots, n\}$, em ordem crescente, tais que $D_j \neq E$.
- $\overrightarrow{D} - E \rightleftharpoons D_{i1}, \ldots, D_{ir}$, onde $i_1, \ldots\ i_r$ são os naturais $j \in \{1, \ldots, n\}$, em ordem crescente, tais que, para cada $k \in \{1, \ldots, p\}, D_j \neq E_k$.

1.7 Fato. *$\overrightarrow{z}$ é livre em $\overrightarrow{\Psi}\overrightarrow{y}P$ se, e somente se, existe $i \in \{1, \ldots, p\}$ tal que*
$$\begin{cases} z_i \text{ não ocorre em } \overrightarrow{y}, \\ z_i \text{ é livre em } P. \end{cases}$$

1.8 Definição.

Sejam $x_1, \ldots, x_n$ variáveis distintas $n \geq 0$. Obtemos *a instanciação simultânea de* $\overrightarrow{x}$ por $\overrightarrow{t}$ *em um termo ou uma fórmula* sem quantificadores, substituindo simultaneamente todas as ocorrências livres de $x_1, \ldots, x_n$, respectivamente, por $t_1, \ldots, t_n$.

Se a fórmula considerada for quantificada, então tal instanciação é definida pelas seguintes cláusulas:

- Se existe $i \in \{1, \ldots, n\}$ tal que $x_i = y$ ou $x_i = t_i$ ou x_i não é livre em P, então $(\Psi y P)(\overrightarrow{x}|\overrightarrow{t}) \rightleftharpoons (\Psi y P)(\overrightarrow{w}|\overrightarrow{u})$, onde
$$\begin{cases} i_1, \ldots, i_r \text{ são os naturais } i \in \{1, \ldots, n\} \text{ tais que } \begin{cases} * \quad x_i \neq y, \\ * \quad x_i \neq t_i, \\ * \quad x_i \text{ é livre em } P, \end{cases} \\ \overrightarrow{w} = x_{i1}, \ldots, x_{ir}, \\ \overrightarrow{u} = t_{i1}, \ldots, t_{ir}. \end{cases}$$
- Se, para cada $i \in \{1, \ldots, n\}$, $\begin{cases} x_i \neq y, \\ x_i \neq t_i, \\ x_i \text{ é livre em } P, \end{cases}$ então
$$(\Psi y P)(\overrightarrow{x}|\overrightarrow{t}) \rightleftharpoons \begin{cases} * \quad \Psi y P(\overrightarrow{x}|\overrightarrow{t}), \text{ se } y \text{ não é livre em } \overrightarrow{t}, \\ * \quad \Psi z P(y|z)(\overrightarrow{x}|\overrightarrow{t}), \text{ em caso contrário, onde } z \text{ é a} \\ \quad\ \text{primeira variável não livre em } \overrightarrow{t}, P. \end{cases}$$

1.9 Escólio. *Se* $n = 0$ *, então* $P(\overrightarrow{x}|\overrightarrow{t}) = P$.

1.10 Escólio. *Se* $n = 1$*, temos que a instanciação simultânea de* $x_1, \ldots, x_n$ *por* $t_1, \ldots, t_n$*, em um designador, resulta o mesmo que a instanciação simples de* x_1 *por* t_1 *no mesmo designador.*

1.11 Lema.

- *Se* $\begin{cases} i_1, \ldots, i_n \text{ é uma permutação de } \{1,\ldots,n\}, \\ \overrightarrow{y} = x_{i1}, \ldots, x_{in}, \\ \overrightarrow{u} = t_{i1}, \ldots, t_{in}, \end{cases}$

  então $P(\overrightarrow{x}|\overrightarrow{t}) = P(\overrightarrow{y}|\overrightarrow{u})$.

1.12 Lema.

- *Se, para cada* $i \in \{1, \ldots, n\}$, $\begin{cases} x_i \neq y, \\ x_i \neq t_i, \\ x_i \text{ é livre em } P, \end{cases}$

  então $(\Psi y P)(\overrightarrow{x}|\overrightarrow{t}) = \Psi z P(\overrightarrow{y}|\overrightarrow{z})(\overrightarrow{x}|\overrightarrow{t})$,

  onde $\begin{cases} z = y \text{ se } y \text{ não é livre em } \overrightarrow{t}, \\ z \text{ é a primeira variável não livre em } \overrightarrow{t}, P, \text{ em caso contrário.} \end{cases}$

1.13 Lema.

- *Se, para cada* $i \in \{1, \ldots, n\}$, $\begin{cases} x_i \neq y, \\ x_i \neq t_i, \\ x_i \text{ é livre em } P, \end{cases}$

  então existe uma variável z *tal que*

  $\begin{cases} z \text{ não é livre em } \{\overrightarrow{x}, \overrightarrow{t}, \forall y P\}, \\ (\Psi y P)(\overrightarrow{x}|\overrightarrow{t}) = \Psi z P(\overrightarrow{y}|\overrightarrow{z})(\overrightarrow{x}|\overrightarrow{t}). \end{cases}$

1.14 Fato.

$P(\overrightarrow{x}|\overrightarrow{t}) = P(\overrightarrow{z}|\overrightarrow{u})$, *onde* $\begin{cases} \bullet\ \overrightarrow{z} = x_{i1}, \ldots, x_{ir} \text{ está contida em } \overrightarrow{x}, \\ \bullet\ \text{para cada } j \notin \{i_1, \ldots, i_r\}, x_j = t_j \text{ ou } x_j \\ \quad \text{não é livre em } P, \\ \bullet\ \overrightarrow{u} = t_{i1}, \ldots, t_{ir}. \end{cases}$

1.15 Fato.

$P(\overrightarrow{x}|\overrightarrow{t}) = P(\overrightarrow{z}|\overrightarrow{u})$, *onde* $\begin{cases} \bullet\ i_1, \ldots, i_r \text{ são os naturais } i \in \{1, \ldots, n\} \\ \quad \text{tais que, } \begin{cases} x_i \neq t_i, \\ x_i \text{ é livre em } P, \end{cases} \\ \bullet\ \overrightarrow{z} = x_{i1}, \ldots, x_{ir}, \\ \bullet\ \overrightarrow{u} = t_{i1}, \ldots, t_{ir}. \end{cases}$

1.16 Fato. $P(\overrightarrow{x}|\overrightarrow{t}) = P$ *se, e somente se, para cada* $i \in \{1, \ldots, n\}, x_i = t_i$ *ou* x_i *não é livre em* P.

1.17 Fato.
As seguintes proposições são equivalentes:
- $\overrightarrow{y}$ *é livre em* $P(\overrightarrow{x}|\overrightarrow{t})$.
- *existe* $i \in \{1,\ldots,n\}, y_i$ *é livre em* $\forall\overrightarrow{x}P$ *ou* $\begin{cases} x_i \text{ é livre em } P, \\ y \text{ é livre em } t_i. \end{cases}$

Formulamos a seguir alguns fatos fundamentais relacionando *instanciação simultânea* com *aceitação simultânea*, os quais constituem um lema para fatos mais importantes concernentes à instanciação simultânea.

1.18 Lema.

(i) *Se* $\begin{cases} \overrightarrow{y} \text{ não é livre em } \forall\overrightarrow{x}P, \\ \overrightarrow{x} \text{ aceita } \overrightarrow{y} \text{ em } P, \\ \overrightarrow{x} \text{ aceita } \overrightarrow{t} \text{ em } P, \end{cases}$ *então* $P(\overrightarrow{x}|\overrightarrow{y})(\overrightarrow{y}|\overrightarrow{t}) = P(\overrightarrow{x}|\overrightarrow{t})$.

(ii) *Se* $\begin{cases} \overrightarrow{y} \text{ não é livre em } \forall\overrightarrow{x}P, \\ \overrightarrow{x} \text{ aceita } \overrightarrow{y} \text{ em } P, \end{cases}$ *então* $P(\overrightarrow{x}|\overrightarrow{y})(\overrightarrow{y}|\overrightarrow{x}) = P$.

(iii) *Se* $\begin{cases} \overrightarrow{x} \text{ e } \overrightarrow{z} \text{ são disjuntas}, \\ \overrightarrow{x} \text{ não é livre em } \overrightarrow{u}, \\ \overrightarrow{z} \text{ não é livre em } \overrightarrow{t}, \\ \overrightarrow{x} \text{ aceita } \overrightarrow{t} \text{ em } P, \\ \overrightarrow{z} \text{ aceita } \overrightarrow{u} \text{ em } P, \end{cases}$
então $P(\overrightarrow{x}|\overrightarrow{t})(\overrightarrow{z}|\overrightarrow{u}) = P(\overrightarrow{z}|\overrightarrow{u})(\overrightarrow{x}|\overrightarrow{t})$.

(iv) *Se* $\begin{cases} \overrightarrow{x} \text{ não é livre em } \overrightarrow{t}, \\ \overrightarrow{x} \text{ aceita } \overrightarrow{t} \text{ em } P, \end{cases}$ *então* $P(\overrightarrow{x}|\overrightarrow{t}) = P(x_1|t_1)\ldots(x_n|t_n)$.

(v) *Se* $\begin{cases} \overrightarrow{y} \text{ é própria}, \\ \overrightarrow{y} \text{ não é livre em } \overrightarrow{x}, \overrightarrow{t}, P, \\ \overrightarrow{x} \text{ aceita } \overrightarrow{y} \text{ em } P, \\ \overrightarrow{x} \text{ aceita } \overrightarrow{t} \text{ em } P, \end{cases}$
então $P(\overrightarrow{x}|\overrightarrow{t}) = P(x_1|y_1)\ldots(x_n|y_n)(y_1|t_1)\ldots(y_n|t_n)$.

1.19. **Leis Fundamentais da Instanciação Simultânea**

(i) Se $P \approx_c Q$, então $P(\overrightarrow{x}|\overrightarrow{t}) \approx_c Q(\overrightarrow{x}|\overrightarrow{t})$.

(ii) Se $\overrightarrow{y}$ não é livre em $\forall\overrightarrow{x}P$, então $P(\overrightarrow{x}|\overrightarrow{y})(\overrightarrow{y}|\overrightarrow{t}) \approx_c P(\overrightarrow{x}|\overrightarrow{t})$.

(iii) Se $\overrightarrow{y}$ não é livre em $\forall\overrightarrow{x}P$, então $P(\overrightarrow{x}|\overrightarrow{y})(\overrightarrow{y}|\overrightarrow{x}) \approx_c P$.

(iv) Se $\begin{cases} \overrightarrow{x} \text{ e } \overrightarrow{z} \text{ são disjuntas}, \\ \overrightarrow{x} \text{ não é livre em } \overrightarrow{u}, \\ \overrightarrow{z} \text{ não é livre em } \overrightarrow{t}, \end{cases}$
então $P(\overrightarrow{x}|\overrightarrow{t})(\overrightarrow{z}|\overrightarrow{u}) \approx_c P(\overrightarrow{z}|\overrightarrow{u})(\overrightarrow{x}|\overrightarrow{t})$.

(v) Se $\begin{cases} \overrightarrow{x} \text{ e } z \text{ são disjuntas}, \\ w \text{ não é livre em } \overrightarrow{x}, \overrightarrow{t}, \forall zP, \end{cases}$
então $(\Psi zP)(\overrightarrow{x}|\overrightarrow{t}) \approx_c (\Psi wP)(z|w)(\overrightarrow{x}|\overrightarrow{t})$.

1.20 Fato.

- *Se* $\overrightarrow{x}$ *não é livre em* $\overrightarrow{t}$, *então* $P(\overrightarrow{x}\,|\,\overrightarrow{t}\,) \approx_c P(\overrightarrow{x_1}|\overrightarrow{t_1}) \ldots (\overrightarrow{x_n}|\overrightarrow{t_n})$.

Finalmente dispomos de uma lei sintática relacionando instanciação simultânea e instanciação simples de uma forma geral.

1.21 Fato.

- *Se* $\begin{cases} \overrightarrow{y} \text{ é própria,} \\ \overrightarrow{y} \text{ não é livre em } \overrightarrow{x}, \overrightarrow{t}, P, \end{cases}$

  então $P(\overrightarrow{x}\,|\,\overrightarrow{t}\,) \approx_c P(x_1|y_1) \ldots (x_n|y_n)(y_1|t_1) \ldots (y_n|t_n)$.

Em algumas situações é relevante isolar uma instanciação simples a partir de uma instanciação simultânea:

1.22 Fato.

Se $\begin{cases} \bullet\ \overrightarrow{y} \text{ não é livre em } \overrightarrow{x}, \overrightarrow{t}, P, \\ \bullet\ i \in \{1, \ldots, n\}, \\ \bullet\ \overrightarrow{z} = \overrightarrow{y} - y_i, \\ \bullet\ \overrightarrow{u} = \overrightarrow{t} - t_i, \end{cases}$ *então* $P(\overrightarrow{x}\,|\,\overrightarrow{t}\,) \approx_c P(\overrightarrow{x}\,|\,\overrightarrow{y}\,)(\overrightarrow{z}\,|\,\overrightarrow{u}\,)(y_i|t_i)$.

Queremos saber quando duas fórmulas das formas $\overrightarrow{\Psi}\overrightarrow{x}P$ e $\overrightarrow{\Upsilon}\overrightarrow{z}Q$ são congruentes. Analisaremos a seguir tal questão. Entrementes veremos também como operar uma instanciação simultânea sobre uma fórmula quantificada iterativamente.

1.23 Fato.

- *Se* $\begin{cases} \overrightarrow{\Psi} \neq \overrightarrow{\Upsilon}, \\ P \text{ e } Q \text{ não possuem quantificadores,} \end{cases}$ *então* $\overrightarrow{\Psi}\overrightarrow{x}P \not\approx_c \overrightarrow{\Upsilon}\overrightarrow{z}Q$.

1.24 Lema. $\overrightarrow{\Psi}\overrightarrow{x}P \approx_c \overrightarrow{\Upsilon}\overrightarrow{x}Q$ *se, e somente se,* $P \approx_c Q$.

1.25 Lema.

- *Se* $\begin{cases} \overrightarrow{x} \text{ e } \overrightarrow{y} \text{ são próprias,} \\ \overrightarrow{y} \text{ não é livre em } \overrightarrow{x}, P, \end{cases}$

  então $\begin{cases} \overrightarrow{\Psi}\overrightarrow{x}P \approx_c \overrightarrow{\Psi}\overrightarrow{y}P(x_1|y_1) \ldots (x_n|y_n), \\ \overrightarrow{\Psi}\overrightarrow{x}P \approx_c \overrightarrow{\Psi}\overrightarrow{y}P(\overrightarrow{x}\,|\,\overrightarrow{y}\,). \end{cases}$

1.26 Lema.

(i) *Se* $\begin{cases} \overrightarrow{x} \text{ e } \overrightarrow{y} \text{ são próprias,} \\ \overrightarrow{y} \text{ não é livre em } \forall\overrightarrow{x}P, \end{cases}$ *então* $\overrightarrow{\Psi}\overrightarrow{x}P \approx_c \overrightarrow{\Psi}\overrightarrow{y}P(\overrightarrow{x}\,|\,\overrightarrow{y}\,)$.

(ii) *Se* $\begin{cases} \overrightarrow{x} \text{ e } \overrightarrow{y} \text{ são próprias,} \\ \overrightarrow{y} \text{ não é livre em } \forall\overrightarrow{x}P, \\ P(\overrightarrow{x}\,|\,\overrightarrow{y}\,) \approx_c Q, \end{cases}$ *então* $\overrightarrow{\Psi}\overrightarrow{x}P \approx_c \overrightarrow{\Psi}\overrightarrow{y}Q$.

1.27 Fato.

- *Se* $\begin{cases} \overrightarrow{x}, \overrightarrow{z} \text{ e } \overrightarrow{w} \text{ são próprias,} \\ \overrightarrow{x} \text{ e } \overrightarrow{z} \text{ são disjuntas,} \\ \overrightarrow{w} \text{ não é livre em } \overrightarrow{x}, \overrightarrow{t}, \forall \overrightarrow{z} P, \end{cases}$

  então $(\overrightarrow{\Upsilon} \overrightarrow{z} P)(\overrightarrow{x} | \overrightarrow{t}) \approx_c \overrightarrow{\Upsilon} \overrightarrow{w} P(\overrightarrow{z} | \overrightarrow{w})(\overrightarrow{x} | \overrightarrow{t})$.

1.28 Fato.

- *Se* $\overrightarrow{x}$ *e* $\overrightarrow{y}$ *são próprias, então* $\overrightarrow{\Psi} \overrightarrow{x} P \approx_c \overrightarrow{\Psi} \overrightarrow{y} Q$ *sss* $\begin{cases} \overrightarrow{y} \text{ não é livre em } \forall \overrightarrow{x} P, \\ P(\overrightarrow{x} | \overrightarrow{y}) \approx_c Q. \end{cases}$

1.29 Fato.

- *Se* $\overrightarrow{\Psi} \overrightarrow{x} P \approx_c \overrightarrow{\Psi} \overrightarrow{y} Q$, *então, para cada* $i \in \{1, \ldots, n\}$, x_i *é livre em* P *sss* y_i *é livre em* Q.

1.30 Fato.

- *Se* $\begin{cases} * \ \overrightarrow{x}, z \text{ é própria,} \\ * \ \overrightarrow{y}, w \text{ é própria,} \\ * \ z \text{ não é livre em } P, \\ * \ w \text{ não é livre em } Q, \end{cases}$

  então $\overrightarrow{\Psi} \overrightarrow{x} \Psi z P \approx_c \overrightarrow{\Psi} \overrightarrow{y} \Psi w Q$ *sss* $\overrightarrow{\Psi} \overrightarrow{x} P \approx_c \overrightarrow{\Psi} \overrightarrow{y} Q$.

1.31 Definição.
Seja $I = \langle \Delta, w, s \rangle$.
$I(x_1, \ldots, x_n | d_1, \ldots, d_n) = \langle \Delta, w, s' \rangle$, onde

$$s'(y) = \begin{cases} d_1, \text{ se } y = x_1 \\ \vdots \\ d_n, \text{ se } y = x_n \\ s(y), \quad \text{se} \\ y \notin \{x_1, \ldots, x_n\}. \end{cases}$$

1.32 Definição.
Seja $I = \langle \Delta, w, d \rangle$ uma LQC-interpretação para P.

- $I_V(\forall x_1 \ldots \forall x_n P) = \text{v}$ sss para qualquer $d_1, \ldots, d_n \in \Delta$, $I(x_1, \ldots, x_n | d_1, \ldots, d_n)_V(P) = \text{v}$.
- $I_V(\exists x_1 \ldots \exists x_n P) = \text{v}$ sss existem $d_1, \ldots, d_n \in \Delta$, $I(x_1, \ldots, x_n)_V(P) = \text{v}$.

Substituição e Escopo

Além da *instanciação de variáveis por termos em fórmulas*, existe outra operação importante de manipulação de fórmulas, que é a *substituição de fórmulas por fórmulas em fórmulas*. Enquanto a primeira operação sintática é fundamental para a formulação de três das quatro leis de introdução e eliminação de quantificadores e suas consequências, a segunda operação é essencial para a enunciação das leis de substituição para equivalência.

Definiremos também a *substituição de termos por termos*, a qual é necessária para a exposição das leis da substituição da igualdade, concernentes à Lógica Equacional.

Para especificar *substituição*, precisamos antes saber o que vem a ser *ocorrência real*.

1.33 Definição.
Uma *ocorrência* de um designador D em um designador E é dita *real em* E se a mesma não suceder em E um dos quantificadores '$\forall$' ou '$\exists$'. [3]

1.34 Definição.
Considere D_1 e D_2 sendo ambos termos ou ambos fórmulas, e Γ uma coleção de designadores. As cláusulas abaixo especificam *substituição de termos por termos* e *substituição de fórmulas por fórmulas*:

- A *substituição* de D_1 *por* D_2 *em* E, notada por $E(D_1 \| D2)$, é o designador obtido de E substituindo todas as ocorrências reais de D_1 em E por D_2.
- A *substituição* de D_1 *por* D_2 *em* Γ, notada por $\Gamma(D_1 \| D_2)$, é a coleção $\{E(D_1 \| D_2) | E \in \Gamma\}$.

Outra ideia imprescindível para a formulação das leis da substituição, dadas na página 193, é *escopo de uma variável*.

1.35 Definição.
Um termo v é dito *estar no escopo de uma variável* x *em uma fórmula* Q se Q possuir uma subfórmula de uma das formas $\forall x R$ ou $\exists x R$, tal que v é real em R. Uma fórmula S é dita *estar no escopo de uma variável* x *em uma fórmula* Q se Q possuir uma subfórmula de uma das formas $\forall x R$ ou $\exists x R$ tal que S ocorre em R. [4] [5]

1.36 Notação. Dado um designador D, uma variável x e uma fórmula Q, a expressão '$esc(D, x, Q)$' abrevia a proposição 'D está no escopo de x em Q'.

1.37 Fato.
As cláusulas seguintes especificam recursivamente quando um designador está no escopo de uma variável em uma fórmula:

---

[3] Tal ocorrência não é real se esta for uma variável que sucede '$\forall$' ou '$\exists$' em E.

[4] Se x está no escopo forte de y em P, então x está no escopo de y em P, mas a recíproca não é necessariamente verdadeira. Considere que a única subfórmula cuja variável quantificada é distinta de x que P possui é da forma $\forall y Q$, tal que x é real em Q, mas x não é livre em Q. Temos então que x está no escopo de y em P, mas x não está no escopo forte de y em P, pelo fato de x não possuir em P nenhuma ocorrência livre figurando em $\forall y Q$.

[5] Note que 'estar no escopo forte de' é uma relação entre variáveis em uma fórmula, enquanto que 'estar no escopo de' é uma relação entre um designador e uma variável em uma fórmula.

- *Não é o caso que* $esc(D, x, p(t_1, \ldots, t_n))$*,*
- $esc(D, x, \neg Q)$ *sss* $esc(D, x, Q)$*,*
- $esc(D, x, Q\#R)$ *sss* $esc(D, x, Q)$ *ou* $esc(D, x, R)$*,*
- $esc(D, x, \Psi x Q)$ *sss* D *é real em* Q*,*
- *Se* $x \neq y$*, então* $esc(D, x, \Psi y Q)$ *sss* $esc(D, x, Q)$*.*

Substituição Uniforme

Há ainda uma segunda forma de substituição, a *substituição uniforme*, a qual fundamenta a lei da substituição uniforme.

1.38 Definição.
Sejam f e p, respectivamente, um sinal funcional e um sinal predicativo, ambos de aridade n. Especificamos a seguir o que entendemos por *substituição uniforme em um dado termo* u, em uma dada fórmula Q e em uma dada coleção de fórmulas Γ:

- A substituição uniforme de $f(\overrightarrow{x})$ por t em u, notada por $u[f(\overrightarrow{x})|t]$, é a samblagem obtida de u substituindo todos os termos da forma $f(\overrightarrow{t})$ por $t(\overrightarrow{x}|\overrightarrow{t})$.
- A substituição uniforme de $f(\overrightarrow{x})$ por t em Q, notada por $Q[f(\overrightarrow{x})|t]$, é a samblagem obtida de Q substituindo todos os termos da forma $f(\overrightarrow{t})$ por $t(\overrightarrow{x}|\overrightarrow{t})$.
- A substituição uniforme de $p(\overrightarrow{x})$ por P em Q, notada por $Q[p(\overrightarrow{x})|P]$, é a samblagem obtida de Q substituindo todos os termos da forma $p(\overrightarrow{t})$ por $P(\overrightarrow{x}|\overrightarrow{t})$.
- A substituição uniforme de $f(\overrightarrow{x})$ por t em Γ, notada por $\Gamma[f(\overrightarrow{x})|t]$, é a coleção de samblagens da forma $Q[f(\overrightarrow{x})|t]$, tal que $Q \in \Gamma$.
- A substituição uniforme de $p(\overrightarrow{x})$ por P em Γ, notada por $\Gamma[p(\overrightarrow{x})|P]$, é a coleção de samblagens da forma $Q[p(\overrightarrow{x})|P]$, tal que $Q \in \Gamma$.

1.39 Definição.

- $\overrightarrow{D}(x|t) \rightleftharpoons D_1(x|t), \ldots, D_n(x|t)$.
- $\overrightarrow{D}(\overrightarrow{x}|\overrightarrow{t}) \rightleftharpoons D_1(\overrightarrow{x}|\overrightarrow{t}), \ldots, D_n(\overrightarrow{x}|\overrightarrow{t})$.

1.40. **Leis Sintáticas da Substituição Uniforme**
Se t e P possuem no máximo $x_1, \ldots, x_n$ como variáveis livres, então:

- $t(\overrightarrow{x}|\overrightarrow{t})(y|v) = t(\overrightarrow{x}|\overrightarrow{t}(y|v))$.
- $P(\overrightarrow{x}|\overrightarrow{t})(y|v) \approx_c P(\overrightarrow{x}|\overrightarrow{t}(y|v))$.
- $u[f(\overrightarrow{x})|t](y|v) = u(y|v)[f(\overrightarrow{x})|t]$.
- $Q[f(\overrightarrow{x})|t](y|v) \approx_c Q(y|v)[f(\overrightarrow{x})|t]$.
- $Q[p(\overrightarrow{x})|P](y|v) \approx_c Q(y|v)[p(\overrightarrow{x})|P]$.

§2. Cálculo de Sequentes

As leis de substituição para equivalência permitem trabalhar com fórmulas equivalentes de forma análoga ao que é feito com termos iguais em várias lógicas. Versões delas podem ser encontradas em [20, 49, 50].

2.1. **Esquema da Substituição para Equivalência**
Se $x_1, \ldots, x_n$ são as variáveis livres em $\{P_1, P_2\}$ tais que S está no seu escopo em Q, então
$\forall x_1, \ldots, \forall x_n(P_1 \leftrightarrow P_2) \vdash Q(S\|P_1) \leftrightarrow Q(S\|P_2)$.

2.2. **Regra da Substituição para Equivalência**
Se $\begin{cases} \Gamma \vdash P_1 \leftrightarrow P_2, \\ S \text{ não está, em } Q\text{, no escopo de nenhuma variável livre em } \Gamma \text{ e em } \{P_1, P_2\}, \end{cases}$
então $\Gamma \vdash Q(S\|P_1) \leftrightarrow Q(S\|P_2)$.

Prova:

1	$\Gamma \vdash P_1 \leftrightarrow P_2$
2	S não está, em Q, no escopo de nenhuma var. liv. em Γ e em $\{P_1, P_2\}$
3	$x_1, \ldots, x_n$ são as var. liv. em $\{P_1, P_2\}$ tal que S está em Q no seu escopo
4	para cada $i \in \{1, \ldots, n\}$, x_i não é livre em Γ
5	$\Gamma \vdash \forall x_1, \ldots, \forall x_n(P_1 \leftrightarrow P_2)$
6	$\forall x_1, \ldots, \forall x_n(P_1 \leftrightarrow P_2) \vdash Q(S\|\|P_1) \leftrightarrow Q(S\|\|P_2)$
7	$\Gamma \vdash Q(S\|\|P_1) \leftrightarrow Q(S\|\|P_2)$

□

O exemplo abaixo mostra a importância da restrição na formulação da regra da substituição para equivalência.

2.3 Exemplo. Temos que TNI $\vdash \forall x(x = 2 \rightarrow primo(x))$, e daí TNI$, x = 2 \vdash \forall x(x = 2 \rightarrow primo(x))$. Temos também que TNI$, x = 2 \vdash x = 2 \leftrightarrow par(x)$. Aplicando a regra da substituição para equivalência sem levar em conta a sua restrição, segue-se que TNI$, x = 2 \vdash \forall x(par(x) \rightarrow primo(x))$, e daí TNI $\vdash x = 2 \rightarrow \forall x(par(x) \rightarrow primo(x))$, donde, aplicando-se sucessivamente a regra da instanciação e modus ponens, TNI $\vdash \forall x(par(x) \rightarrow primo(x))$, o que é incorreto. Observe que a fórmula $x = 2$ está no escopo de x e, ao mesmo tempo, x é livre em

$(\text{TNI} \cup \{x = 2\}) \cap \{x = 2, par(x)\}$, ou seja, $x = 2$ não pode ser substituído por $par(x)$ em $\text{TNI}, x = 2 \vdash \forall x(x = 2 \to primo(x))$ sem gerar algo incorreto.

2.4 Corolário.

Se $\begin{cases} \Gamma \vdash Q(S \| P_1), \\ \Gamma \vdash P_1 \leftrightarrow P_2, \\ \textit{S não está, em Q, no escopo de nenhuma variável livre em } \Gamma \textit{ e em } \{P_1, P_2\}, \end{cases}$

então $\Gamma \vdash Q(S \| P_2)$.

Prova:

1	$\Gamma \vdash Q(S \| P_1)$	hip
2	$\Gamma \vdash P_1 \leftrightarrow P_2$	hip
3	S não está, em $Q_1, \ldots, Q_n$	hip
4	$\Gamma \vdash Q(S \| P_1) \leftrightarrow Q(S \| P_2)$	2, 3, RSE
5	$Q(S \| P_1), Q(S \| P_1) \leftrightarrow Q(S \| P_2)$	MPEq
6	$\Gamma \vdash Q(S \| P_2)$	4, 5, Tran

□

2.5 Corolário.

Se $\begin{cases} \Gamma \vdash Q(S \| P_1), \\ \vdash P_1 \leftrightarrow P_2 \end{cases}$ *então* $\Gamma \vdash Q(S \| P_2)$.

Algumas leis de substituição para implicação são fornecidas a seguir. Estes resultados são especialmente úteis quando, em qualquer ambiente de prova, uma fórmula e uma implicação ocorrerem, de tal modo que o antecedente ou o consequente da implicação figure dentro dessa fórmula.

O seguinte lema também será útil para demonstrar o Esquema da Substituição para Implicação:

2.6 Lema.

$\forall x(P \to Q) \vdash \forall x P \to \forall x Q$.

$\forall x(P \to Q) \vdash \exists x P \to \exists x Q$.

Esqueleto da Prova: Basta usar as leis estruturais, as leis de introdução e eliminação de quantificadores, a regra de dedução e a modus ponens. □

O Esquema da Substituição para Implicação, apresentado abaixo, é essencial para abreviar a prova da Regra da Substituição para Implicação.

2.7. **Esquema da Substituição para Implicação**.

- Se $x_1, \ldots, x_n$ são as variáveis livres em $\{P_1, P_2\}$ tais que S está em Q no seu escopo, então
$$\begin{cases} S \text{ é positivo em } Q\text{, implica que } \forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash Q(S\|P_1) \to Q(S\|P_2). \\ S \text{ é negativo em } Q\text{, implica que } \forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash Q(S\|P_2) \to Q(S\|P_1). \end{cases}$$

Prova:

Seja $\Delta(Q)$ a sentença "para cada lista $x_1, \ldots, x_n$ de variáveis, a proposição anterior é válida".

- Caso S não ocorre em Q.
  Então $Q(S\|P_1) = Q(S\|P_2) = Q$, daí, por RI e MON, não há mais o que provar.
- Caso $S = Q$
  Então $\begin{cases} Q(S\|P_1) = P_1, \\ Q(S\|P_2) = P_2, \end{cases}$ daí, pelo $\forall$-el e Ref, não há mais o que provar.
  Podemos daí considerar, spg, que $S \neq Q$ e S ocorre em Q.

  Sejam $x_1, \ldots, x_n$ as variáveis livres em $\{P_1, P_2\}$ tal que S está em Q no seu escopo.
- Caso Q é fórmula atômica.
  Então $S = Q$ ou S não ocorre em Q, o que já foi considerado.
- Caso Q é $\neg R$.
  Então $x_1, \ldots, x_n$ são as variáveis livres em $\{P_1, P_2\}$ tal que S está em R no seu escopo.
  (i) Subcaso S é positivo em Q.
  Então S é negativo em R.
  Por HI, $\forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash R(S\|P_2) \to R(S\|P_1)$.
  Como $R(S\|P_2) \to R(S\|P_1) \vdash (\neg R)(S\|P_1) \to (\neg R)(S\|P_2)$,
  então $\forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash Q(S\|P_1) \to Q(S\|P_2)$.
  (ii) Subcaso S é negativo em Q.
  Então S é positivo em R.
  Por HI, $\forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash R(S\|P_1) \to R(S\|P_2)$.
  Como $R(S\|P_1) \to R(S\|P_2) \vdash (\neg R)(S\|P_2) \to (\neg R)(S\|P_1)$,
  então $\forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash Q(S\|P_2) \to Q(S\|P_1)$.
- Caso Q é $R_1 \to R_2$.
  Sejam $y_1, \ldots, y_p$ as variáveis livres em $\{P_1, P_2\}$ tal que S está em R_1 no seu escopo.
  Sejam $z_1, \ldots, z_r$ as variáveis livres em $\{P_1, P_2\}$ tal que S está em R_2 no

seu escopo.
Sejam $x_1, \ldots, x_n$ as variáveis distintas de $y_1, \ldots, y_p, z_1, \ldots, z_r$.
Então, $\{x_1, \ldots, x_n\} = \{y_1, \ldots, y_p\} \cup \{z_1, \ldots, z_r\}$.

(i) Subcaso S é positivo em Q.
Então S é negativo em R_1 e S é positivo em R_2.
Por HI, temos que $\begin{cases} \forall y_1 \ldots \forall y_p(P_1 \to P_2) \vdash R_1(S\|P_2) \to R_1(S\|P_1), \\ \forall z_1 \ldots \forall z_r(P_1 \to P_2) \vdash R_2(S\|P_1) \to R_2(S\|P_2). \end{cases}$
Por $\forall$-el e Gen $\begin{cases} \forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash \forall y_1 \ldots \forall y_p(P_1 \to P_2), \\ \forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash \forall z_1 \ldots \forall z_r(P_1 \to P_2). \end{cases}$
Daí, por Tran $\begin{cases} \forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash R_1(S\|P_2) \to R_1(S\|P_1), \\ \forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash R_2(S\|P_1) \to R_2(S\|P_2). \end{cases}$
Por RD e MP, $R_1(S\|P_2) \to R_1(S\|P_1), R_2(S\|P_1) \to R_2(S\|P_2) \vdash Q(S\|P_1) \to Q(S\|P_2)$,
donde, por Tran, $\forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash Q(S\|P_1) \to Q(S\|P_2)$.

(ii) Subcaso S é negativo em Q.
Daí S é positivo em R_1 e S é negativo em R_2.
Por HI, temos que $\begin{cases} \forall y_1 \ldots \forall y_p(P_1 \to P_2) \vdash R_1(S\|P_1) \to R_1(S\|P_2), \\ \forall z_1 \ldots \forall z_r(P_1 \to P_2) \vdash R_2(S\|P_2) \to R_2(S\|P_1). \end{cases}$
Por $\forall$-el e Gen $\begin{cases} \forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash \forall y_1 \ldots \forall y_p(P_1 \to P_2), \\ \forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash \forall z_1 \ldots \forall z_r(P_1 \to P_2). \end{cases}$
Então, por Tran $\begin{cases} \forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash R_1(S\|P_1) \to R_1(S\|P_2), \\ \forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash R_2(S\|P_2) \to R_2(S\|P_1). \end{cases}$
Por RD e MP, $R_1(S\|P_1) \to R_1(S\|P_2), R_2(S\|P_2) \to R_2(S\|P_1) \vdash Q(S\|P_2) \to Q(S\|P_1)$,
donde, por Tran, $\forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash Q(S\|P_2) \to Q(S\|P_1)$.

- Caso Q é $R_1 \# R_2$.
Sejam $y_1, \ldots, y_p$ as variáveis livres em $\{P_1, P_2\}$ tal que S está em R_1 no seu escopo.
Sejam $z_1, \ldots, z_r$ as variáveis livres em $\{P_1, P_2\}$ tal que S está em R_2 no seu escopo.
Então $\{x_1, \ldots, x_n\} = \{y_1, \ldots, y_p\} \cup \{z_1, \ldots, z_r\}$.

(i) Subcaso S é positivo em Q.
Então S é positivo em R_1 e S é positivo em R_2.
Por HI, temos que $\begin{cases} \forall y_1 \ldots \forall y_p(P_1 \to P_2) \vdash R_1(S\|P_1) \# R_1(S\|P_2), \\ \forall z_1 \ldots \forall z_r(P_1 \to P_2) \vdash R_2(S\|P_1) \# R_2(S\|P_2). \end{cases}$

Por $\forall$-el e Gen $\begin{cases} \forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash \forall y_1 \ldots \forall y_p(P_1 \to P_2), \\ \forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash \forall z_1 \ldots \forall z_r(P_1 \to P_2). \end{cases}$

Então, por Tran $\begin{cases} \forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash R_1(S\|P_1)\#R_1(S\|P_2), \\ \forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash R_2(S\|P_1)\#R_2(S\|P_2). \end{cases}$

Por RD e MP, $R_1(S\|P_1)\#R_1(S\|P_2), R_2(S\|P_1)\#R_2(S\|P_2) \vdash Q(S\|P_1) \to Q(S\|P_2)$.

Então, por Tran, $\forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash Q(S\|P_1) \to Q(S\|P_2)$.

(ii) Subcaso S é negativo em Q.
Daí S é negativo em R_1 e S é negativo em R_2.

Por HI, temos que $\begin{cases} \forall y_1 \ldots \forall y_p(P_1 \to P_2) \vdash R_1(S\|P_2)\#R_1(S\|P_1), \\ \forall z_1 \ldots \forall z_r(P_1 \to P_2) \vdash R_2(S\|P_2)\#R_2(S\|P_1). \end{cases}$

Por $\forall$-el e Gen $\begin{cases} \forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash \forall y_1 \ldots \forall y_p(P_1 \to P_2), \\ \forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash \forall z_1 \ldots \forall z_r(P_1 \to P_2). \end{cases}$

Então, por Tran $\begin{cases} \forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash R_1(S\|P_2)\#R_1(S\|P_1), \\ \forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash R_2(S\|P_2)\#R_2(S\|P_1). \end{cases}$

Por RD e MP, $R_1(S\|P_2)\#R_1(S\|P_1), R_2(S\|P_2)\#R_2(S\|P_1) \vdash Q(S\|P_2) \to Q(S\|P_1)$.

Então, por Tran, $\forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash Q(S\|P_2) \to Q(S\|P_1)$.

- Caso Q é $\Psi z R$.
  Sejam $y_1, \ldots, y_p$ as variáveis livres em $\{P_1, P_2\}$ tal que S está no seu escopo em R.
  (i) Subcaso 1: $z \in \{y_1, \ldots, y_p\}$ ou z não é livre em $\{P_1, P_2\}$.
  Então $\{x_1, \ldots, x_n\} = \{y_1, \ldots, y_p\}$. (1)
  (ii) Subcaso 1a: S é positivo em Q.
  Então S é positivo em R.
  Por HI, $\forall y_1 \ldots \forall y_p(P_1 \to P_2) \vdash R(S\|P_1) \to R(S\|P_2)$. (2)
  De (1) e (2), $\forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash R(S\|P_1) \to R(S\|P_2)$.
  Em qualquer caso, z não é livre em $\forall x_1 \ldots \forall x_n(P_1 \to P_2)$, donde, por Gen,
  $\forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash \forall z(R(S\|P_1) \to R(S\|P_2))$.
  De Gen, $\forall$-el, $\exists$-int, $\exists$-el, temos que $\forall z(R(S\|P_1) \to R(S\|P_2)) \vdash (\Psi z R)(S\|P_1) \to (\Psi z R)(S\|P_2)$.
  Por Tran, $\forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash Q(S\|P_1) \to Q(S\|P_2)$.
  (iii) Subcaso 1b: S é negativo em Q.
  Daí S é negativo em R.
  Por HI, $\forall y_1 \ldots \forall y_p(P_1 \to P_2) \vdash R(S\|P_2) \to R(S\|P_1)$. (3)
  De (1) e (3) $\forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash R(S\|P_2) \to R(S\|P_1)$.

Em qualquer caso, z não é livre em $\forall x_1 \ldots \forall x_n(P_1 \to P_2)$, donde, por Gen,
$\forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash \forall z(R(S\|P_2) \to R(S\|P_1))$.
De Gen, $\forall$-el, $\exists$-int, $\exists$-el, temos que $\forall z(R(S\|P_2) \to R(S\|P_1)) \vdash (\Psi z R)(S\|P_2) \to (\Psi z R)(S\|P_1)$.
Por Tran, $\forall x_1 \ldots \forall_n(P_1 \to P_2) \vdash Q(S\|P_2) \to Q(S\|P_1)$.

(iv) Subcaso 2: $z \notin \{y_1, \ldots, y_p\}$ e z é livre em $\{P_1, P_2\}$.
Então $\{x_1, \ldots, x_n\} = \{y_1, \ldots, y_p\} \cup \{z\}$.

(v) Subcaso 2a: S é positivo em Q.
Então, S é positivo em R.
Por HI, $\forall y_1 \ldots \forall y_p(P_1 \to P_2) \vdash R(S\|P_1) \to R(S\|P_2)$.
De Tran e $\forall$-el, $\forall z \forall y_1 \ldots \forall y_p(P_1 \to P_2) \vdash R(S\|P_1) \to R(S\|P_2)$.
De Tran, $\forall$-el e Gen, $\forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash R(S\|P_1) \to R(S\|P_2)$.
Temos que z não é livre em $\forall x_1 \ldots \forall x_n(P_1 \to P_2)$, assim, por Gen,
$\forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash \forall z(R(S\|P_1) \to R(S\|P_2))$.
De Gen, $\forall$-el, $\exists$-int, $\exists$-el, temos que $\forall z(R(S\|P_1) \to R(S\|P_2)) \vdash (\Psi z R)(S\|P_1) \to (\Psi z R)(S\|P_2)$.
Por Tran, $\forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash Q(S\|P_1) \to Q(S\|P_2)$.

(vi) Subcaso 2b: S é negativo em Q.
Então, S é negativo em R.
Por HI, $\forall y_1 \ldots \forall y_p(P_1 \to P_2) \vdash R(S\|P_2) \to R(S\|P_1)$.
De Tran e $\forall$-el, $\forall z \forall y_1 \ldots \forall y_p(P_1 \to P_2) \vdash R(S\|P_2) \to R(S\|P_1)$.
De Tran, $\forall$-el e Gen, $\forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash R(S\|P_2) \to R(S\|P_1)$.
Temos que z não é livre em $\forall x_1 \ldots \forall x_n(P_1 \to P_2)$, assim, por Gen,
$\forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash \forall z(R(S\|P_2) \to R(S\|P_1))$.
De Gen, $\forall$-el, $\exists$-int, $\exists$-el, temos que $\forall z(R(S\|P_1) \to R(S\|P_2)) \vdash (\Psi z R)(S\|P_2) \to (\Psi z R)(S\|P_1)$.
Por Tran, $\forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash Q(S\|P_2) \to Q(S\|P_1)$.

$\square$

A regra da substituição para implicação é uma versão análoga à regra da substituição para equivalência.

2.8. **Regra da Substituição para Implicação**.

Se $\begin{cases} \Gamma \vdash P_1 \to P_2, \\ S \text{ não está, em } Q\text{, no escopo de nenhuma variável livre em } \Gamma \text{ e em } \{P_1, P_2\}, \end{cases}$

então $\begin{cases} S \text{ é positivo em } Q \text{ implica que } \Gamma \vdash Q(S\|P_1) \to Q(S\|P_2), \\ S \text{ é negativo em } Q \text{ implica que } \Gamma \vdash Q(S\|P_2) \to Q(S\|P_1). \end{cases}$

Prova:
Assuma a hipótese.
Sejam $x_1, \ldots, x_n$ as variáveis livres em $\{P_1, P_2\}$ tal que S está no seu escopo em Q.
Pela hipótese temos que $x_1, \ldots, x_n$ não são livres em Γ.

- Caso S é positivo em Q.
  Pelo ESI, $\forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash Q(S\|P_1) \to Q(S\|P_2)$. (1)
  Pela hipótese e por Gen, $\Gamma \vdash \forall x_1 \ldots \forall x_n(P_1 \to P_2)$. (2)
  De (2),(1) e Tran, $\Gamma \vdash Q(S\|P_1) \to Q(S\|P_2)$.
- Caso S é negativo em Q.
  Pelo ESI, $\forall x_1 \ldots \forall x_n(P_1 \to P_2) \vdash Q(S\|P_2) \to Q(S\|P_1)$. (3)
  Pela hipótese e por Gen, $\Gamma \vdash \forall x_1 \ldots \forall x_n(P_1 \to P_2)$ (4).
  De (4) e (3), $\Gamma \vdash Q(S\|P_2) \to Q(S\|P_1)$.

□

2.9 Corolário.

(i) *Se* $\begin{cases} \Gamma \vdash Q(S\|P_1), \\ \Gamma \vdash P_1 \to P_2, \\ \textit{S não está, em Q, no escopo de nenhuma variável livre em } \Gamma \textit{ e em } \{P_1, P_2\}, \\ \textit{S é positivo em Q,} \end{cases}$

então $\Gamma \vdash Q(S\|P_2)$.

(ii) *Se* $\begin{cases} \Gamma \vdash Q(S\|P_2), \\ \Gamma \vdash P_1 \to P_2, \\ \textit{S não está, em Q, no escopo de nenhuma variável livre em } \Gamma \textit{ e em } \{P_1, P_2\}, \\ \textit{S é negativo em Q,} \end{cases}$

então $\Gamma \vdash Q(S\|P_1)$.

2.10 Corolário.

(i) *Se* $\begin{cases} \vdash P_1 \to P_2, \\ \textit{S é positivo em Q,} \end{cases}$ *então* $\vdash Q(S\|P_1) \to Q(S\|P_2)$.

(ii) *Se* $\begin{cases} \vdash P_1 \to P_2, \\ \textit{S é negativo em Q,} \end{cases}$ *então* $\vdash Q(S\|P_2) \to Q(S\|P_1)$.

Esqueleto da Prova: asta aplicar a regra de substituição para implicação. □

2.11 Corolário.

(i) *Se* $\begin{cases} \vdash P_1 \to P_2, \\ \Gamma \vdash Q(S\|P_1), \\ \textit{S é positivo em Q,} \end{cases}$ *então* $\Gamma \vdash Q(S\|P_2)$.

$$\text{(ii) } \textit{Se} \begin{cases} \vdash P_1 \to P_2, \\ \Gamma \vdash Q(S \| P_2), \\ S \textit{ é negativo em } Q, \end{cases} \qquad \textit{então } \Gamma \vdash Q(S \| P_1).$$

Prova de (i):
Sejam $x_1, \ldots, x_n$ as variáveis livres em $\{P_1, P_2\}$ tal que S está em seu escopo em Q.
Pelo esquema de substituição para implicação, temos que $\forall x_1, \ldots, x_n(P_1 \to P_2) \vdash Q(S \| P_1) \to Q(S \| P_2)$.
Pela hipótese e Gen, $\vdash \forall x_1, \ldots, x_n(P_1 \to P_2)$, donde, por Tran, $\vdash Q(S \| P_1) \to Q(S \| P_2)$, então, por Mon, $\Gamma \vdash Q(S \| P_1) \to Q(S \| P_2)$, portanto, pela hipótese, $\Gamma \vdash Q(S \| P_1)$, então concluímos que $\Gamma \vdash Q(S \| P_2)$.
□

Prova de (ii):
É análoga à prova de (i). □

§3. Correção e Completude do cálculo de sequentes de LQC com respeito à semântica de LQC

Tudo que foi provado para LPC, com algumas alterações, também é válido para LQC.

3.1 Correção do cálculo de sequentes de LQC com respeito à semântica de LQC

Para a correção do cálculo de LQC, além da semântica apresentada em §5.2, considere também, a definição 5.2.23 sobre a coincidência entre interpretações para termos e para fórmulas, as definições 5.2.14, 5.2.17, 5.2.19, e os lemas definidos a seguir.

3.1 Lema.

- $\textit{Se} \begin{cases} I = \langle \Delta, w, s \rangle \textit{ é uma } \mathrm{LQC}\textit{-interpretação para } P, \\ y \textit{ não é livre em } \forall x P, \\ d \in \Delta, \end{cases}$
  então $I(x|d)_V(P) = I(y|d)_V(P(x|y))$[6].

[6]Um exemplo prático:
$I(x|Lula)_V(presidente(x)) = I(y|Lula)_V(presidente(x)(x|y))$.

Prova:
Assuma a hipótese.
Vamos nos concentrar nos casos que possuem quantificadores porque os demais casos são triviais.
Temos que $y = x$, ou $y \neq x$ e y não é livre em P.
Spg, considere que $\begin{cases} y \text{ é distinto de } x, \\ y \text{ não é livre em } P. \end{cases}$
Também podemos considerar, spg, que x é livre em P.

- Caso P é da forma $\Psi z Q$ e $z \neq y$.
  $I(x|d)_V(P) = B_\Psi(\{I(x|d)(z|e)_V(Q) \mid e \in \Delta\})$ que é igual a $B_\Psi(\{I(z|e)(x|d)_V(Q) \mid e \in \Delta\})$.
  $I(y|d)_V(P(x|y)) = I(y|d)_V((\Psi z Q)(x|y)) = I(y|d)_V(\Psi z Q(x|y)) = B_\Psi(\{I(y|d)(z|e)_V(Q(x|y)) \mid e \in \Delta\}) =$
  $B_\Psi(\{I(z|e)(y|d)_V(Q(x|y)) \mid e \in \Delta\}) = B_\Psi(\{I(z|e)(x|d)_V(Q) \mid e \in \Delta\})$.
- Caso P é da forma $\Psi y Q$.
  Seja z a primeira variável não livre em x, y, Q.
  Então $P(x|y) = (\Psi y Q)(x|y) = \Psi z Q(y|z)(x|y) = (\Psi z Q(y|z))(x|y)$.
  Pelo caso anterior, temos que
  $I(x|d)_V(\Psi z Q(x|y)) = I(y|d)_V(((\Psi z Q)(x|y))(y|z))$.

$\square$

3.2 Lema.
- *Se* $\begin{cases} I \text{ é LQC-}\textit{interpretação para } P \textit{ e } Q, \\ P \approx_c Q \end{cases}$ *então* $I_V(P) = I_V(Q)$[7].

Prova:
Assuma a hipótese.
- Caso P é da forma $p(t_1, \ldots, t_n), \top$ ou $\bot$.
  Então P e Q são a mesma fórmula, daí $I_V(P) = I_V(Q)$.
- Caso P é $\neg R$.
  Então Q é da forma $\neg S$, onde $R \approx_c S$.
  Por HI, $I_V(R) = I_V(S)$. Daí, $I_V(\neg R) \neq I_V(R) = I_V(S) \neq I_V(\neg S)$, donde $I_V(\neg R) = I_V(\neg S)$[8].
- Caso P é da forma $R \# S$, onde $\# \in \{\rightarrow, \wedge, \vee\}$.
  Então Q é da forma $T \# U$, onde T e U são fórmulas .
  Caso $\begin{cases} R \approx_c T, \\ S \approx_c U, \end{cases}$ temos, por HI, que $\begin{cases} I_V(R) = I_V(T), \\ I_V(S) = I_V(U). \end{cases}$

[7]Ou seja, se duas fórmulas são congruentes e existe uma interpretação para as mesmas, então essas duas fórmulas, para essa interpretação, possuem os mesmos valores veritativos.
[8]Por $x, y, z \in \{v, f\}$ e $x \neq y \neq z$ então $x = z$

Logo $I_V(R\#S) = B_{\#}(I_V(R), I_V(S)) = B_{\#}(I_V(T), I_V(U)) = I_V(T\#U)$, pela definição 5.2.19.

- Caso P é da forma $\Psi x R$, onde $\Psi \in \{\forall, \exists\}$.
  Então Q é $\Psi y R(x|y)$, onde y não é livre em $\forall x R$.
  Por 5.2.19, $I_V(\Psi x R) = B_\Psi(\{I(x|d)_V(R) \mid d \in \Delta\}) = B_\Psi(\{I(y|d)_V(R(x|y)) \mid d \in \Delta\})$, e pelo lema 3.1, $I_V(\Psi y R(x|y))$.

$\square$

3.3 Lema. ***(Da avaliação de termos instanciados e fórmulas instanciadas em*** LQC[9]***)***

- *Se I é uma* LQC*-interpretação para t, u, então* $I_D(u|(x|t)) = I(x|I_D(t))_D(u)$,
- *Se I é uma* LQC*-interpretação para t, P, então* $I_V(P(x|t)) = I(x|I_D(t))_V(P)$.

Prova de (i)[10]*:*

- Caso u é uma constante c.
  $u(x|t) = c(x|t) = c$, daí $I_D(u(x|t)) = I_D(c) = w(c)$.
  Temos também que $I(x|I_D(t))_D(u) = I(x|I_D(t))_D(c) = w(c)$.
- Caso u é x.
  $I_D(u(x|t)) = I_D(x(x|t)) = I_D(t)$.
  $I(x|I_D(t))(u) = I(x|I_D(t))(x) = s(x|I_D(t))(x) = I_D(t)$.
- Caso u é y e $y \neq x$.
  $u(x|t) = y(x|t) = y$, daí $I_D(u|(x|t)) = I_D(y) = s(y)$.
  Temos também que $I(x|I_D(t))_D(u) = I(x|I_D(t))_D(y) = s(x|I_D(t))(y) = s(y)$.
- Caso u é da forma $f(t_1, \ldots, t_n)$.
  Por HI temos que, para cada $i \in \{1, \ldots, n\}$, $I_D(t_i(x|t)) = I(x|I_D(t))_D(t_i)$.
  Temos que $f(t_1, \ldots, t_n)(x|t) = f(t_1(x|t), \ldots, t_n(x|t))$.
  $I_D(f(t_1, \ldots, t_n)(x|t)) = I_D(f(t_1(x|t), \ldots, t_n(x|t))) = w(f)(I_D(t_1(x|t)), \ldots, I_D(t_n(x|t)))$, por HI, isso é igual a $w(f)(I(x|I_D(t))(t_1), \ldots, I(x|I_D(t))(t_n)) = I(x|I_D(t))(f(t_1, \ldots, t_n))$.

$\square$

Prova de (ii)[11]*:*

- Caso P é da forma $\top$.
  $P(x|t) = \top(x|t) = \top$.
  $I_V(P(x|t)) = I_V(\top) = \mathrm{v}$.
  $I(x|I_D(t))_V(P) = I(x|I_D(t))_V(\top) = \mathrm{v}$.

---

[9]Exemplo de termo instanciado: $u(x|t)$ e fórmula instanciada: $P(x|t)$.

- Caso P é da forma $\perp$.
  $P(x|t) = \perp(x|t) = \perp$.
  $I_V(P(x|t)) = I_V(\perp) = \text{f}$.
  $I(x|I_D(t))_V(P) = I(x|I_D(t))_V(\perp) = \text{f}$.
- Caso P é da forma $p(t_1, \ldots, t_n)$.
  Por HI, temos que para cada $i \in (1, \ldots, n)$, $I_D(t_i(x|t)) = I(x|I_D(t))_D(t_i)$.
  Temos que $p(t_1, \ldots, t_n)(x|t) = p(t_1(x|t), \ldots, t_n(x|t))$.
  $I_V(p(t_1, \ldots, t_n)(x|t)) = \text{v}$ sss $I_V(p(t_1(x|t), \ldots, t_n(x|t))) = \text{v}$ sss $\langle I_D(t_1(x|t)), \ldots, I_D(t_n(x|t))\rangle \in w(p)$.
  $\langle I(x|I_D(t))_D(t_1), \ldots, I(x|I_D(t))(t_n)\rangle \in w(p)$ sss $I(x|I_D(t))_V(p(t_1, \ldots, t_n)) = \text{v}$.
  Logo, $I_V(p(t_1, \ldots, t_n)(x|t)) = I(x|I_D(t))_V(p(t_1, \ldots, t_n))$.
- Caso P é $\neg Q$.
  $(\neg Q)(x|t) = \neg Q(x|t)$.
  Por HI, $I_V(Q(x|t)) = I(x|I_D(t))_V(Q)$.
  $I_V((\neg Q)(x|t)) = I_V(\neg Q(x|t)) \neq I_V(Q(x|t))$ que por HI é igual a $I(x|I_D(t))_V(Q)$ que é diferente de $I(x|I_D(t))_V(\neg Q)$.
  Temos que[12] $I_V((\neg Q)(x|t)) \neq I_V(Q(x|t)) \neq I(x|I_D(t))_V(\neg Q)$, logo $I_V((\neg Q)(x|t)) = I(x|I_D(t))_V(\neg Q)$.
- P é da forma $Q \to R$.
  $P(x|t) = Q(x|t) \to R(x|t)$.
  Por HI, $\begin{cases} I_V(Q(x|t)) = I(x|I_D(t))_V(Q), \\ I_V(R(x|t)) = I(x|I_D(t)_V(R)). \end{cases}$
  $I_V((Q \to R)(x|t)) = I_V(Q(x|t) \to R(x|t)) = \begin{cases} I_V(R(x|t)), \text{ se } I_V(Q(x|t)) = \text{v}, \\ \text{v, se } I_V(Q(x|t)) = \text{f}. \end{cases}$
  Mas estes dois ítens, por HI, são iguais a $\begin{cases} I(x|I_D(t))_V(R), \text{ se } I(x|I_D(t))_V(Q) = \text{v}, \\ \text{v, se } I(x|I_D(t))_V(Q) = \text{f}, \end{cases}$
  mas $I(x|I_D(t))_V(Q \to R)$, ou seja, $I_V((Q \to R)(x|t)) = I(x|I_D(t))_V(Q \to R)$.
- Caso P é da forma $Q \wedge R$.
  $P(x|t) = Q(x|t) \wedge R(x|t)$.
  Por HI, $\begin{cases} I_V(Q(x|t)) = I(x|I_D(t))_V(Q), \\ I_V(R(x|t)) = I(x|I_D(t)_V(R)). \end{cases}$
  $I_V((Q \wedge R)(x|t)) = I_V(Q(x|t) \wedge R(x|t)) = \min\{I(x|I_D(t))_V(Q), I(x|I_D(t))_V(R))\}$.
  Mas $I(x|I_D(t))_V(Q \wedge R)$, ou seja, $I_V((Q \wedge R)(x|t)) = I(x|I_D(t))_V(Q \wedge R)$.

---

[12] Por $x, y, z \in \{v, f\}$ e $x \neq y \neq z$ então $x = z$.

- Caso P é da forma $Q \vee R$.
  $P(x|t) = Q(x|t) \vee R(x|t)$.
  Por HI, $\begin{cases} I_V(Q(x|t)) = I(x|I_D(t))_V(Q), \\ I_V(R(x|t)) = I(x|I_D(t)_V(R)). \end{cases}$
  $I_V((Q \vee R)(x|t)) = I_V(Q(x|t) \vee R(x|t)) = \max\{I(x|I_D(t))_V(Q), I(x|I_D(t))_V(R))\}$.
  Mas $I(x|I_D(t))_V(Q \vee R)$, ou seja, $I_V((Q \vee R)(x|t)) = I(x|I_D(t))_V(Q \vee R)$.
- Caso Q é da forma ΨxQ e $y = x$.
  $P(x|t) = (\Psi xQ)(x|t) = \Psi xQ$.
  $I_V((\Psi xQ)(x|t)) = I_V(\Psi xQ)$.
  $I(x|I_D(t))_V(\Psi xQ)$, pelo lema 5.2.23, $I_V(\Psi xQ)$.
- Caso P é da forma $\forall yQ$ e $y = x$.
  Seja z uma variável não livre em x, t, Q.
  Daí $\forall yQ \approx_c \forall zQ(y|z)$. (1)
  $(\forall yQ)(x|t) \approx_c \forall zQ(y|z)(x|t)$, então pelo lema 3.2, $I_V((\forall yQ)(x|t)) = I_V(\forall zQ(y|z)(x|t))$ que é igual a
  $\min\{I(z|d)_V(Q(y|z)(x|t)) \mid d \in \Delta\}$, igual a
  $\min\{I(z|d)(x|I(z|d)_D(t))_V(Q(y|z)) \mid d \in \Delta\}$, pois
  $I(z|d)_D(t) = I_D(t)$, pelo lema 5.2.23 é igual a
  $\min\{I(z|d)(x|I_D(t))_V(Q(y|z)) \mid d \in \Delta\}$ que é igual a
  $\min\{I(x|I_D(t))(z|d)_V(Q(y|z)) \mid d \in \Delta\}$ pelo escólio 5.2.21 é igual a $I(x|I_D(t))(\forall zQ(y|z))$, que pelo lema 3.2 e (1) é igual a $I(x|I_D(t))(\forall yQ)$.
- Caso P é da forma $\exists yQ$ e $y = x$.
  Seja z uma variável não livre em x, t, Q.
  Daí $\exists yQ \approx_c \exists zQ(y|z)$. (1)
  $(\exists yQ)(x|t) \approx_c \exists zQ(y|z)(x|t)$, então pelo lema 3.2, $I_V((\exists yQ)(x|t)) = I_V(\exists zQ(y|z)(x|t))$ que é igual a
  $\max\{I(z|d)_V(Q(y|z)(x|t)) \mid d \in \Delta\}$, igual a
  $\max\{I(z|d)(x|I(z|d)_D(t))_V(Q(y|z)) \mid d \in \Delta\}$, pois
  $I(z|d)_D(t) = I_D(t)$, pelo lema 5.2.23 é igual a
  $\max\{I(z|d)(x|I_D(t))_V(Q(y|z)) \mid d \in \Delta\}$ que é igual a
  $\max\{I(x|I_D(t))(z|d)_V(Q(y|z)) \mid d \in \Delta\}$ pelo escólio 5.2.21 é igual a $I(x|I_D(t))(\exists zQ(y|z))$, que pelo lema 3.2 e (1) é igual a $I(x|I_D(t))(\exists yQ)$.

$\square$

Podemos agora, demonstrar a correção dos cálculo de sequentes de LQC com respeito às semânticas de valorações de LQC.

É suficiente provar que todas as leis (esquemas e regras) do cálculo de sequentes para LQC são corretos. Para as leis estruturais e para as leis de introdução e eliminação de conectivos o raciocínio é análogo à prova do teorema 9.3.1 feita no caso proposicional. Resta evidenciar a correção das leis concernentes aos quantificadores: leis de introdução e eliminação de quantificadores.

3.4 Teorema. ***(Correção)***
Se $\Gamma \underset{\mathrm{LQC}}{\vdash} P$, *então* $\Gamma \underset{\mathrm{LQC}}{\models} P$.
Prova:

- Queremos mostrar que a regra da generalização é correta.
  Seja $\dfrac{\Gamma \vdash P}{\Gamma \vdash \forall x P}$ uma aplicação de Gen.
  Daí x não é livre em Γ.
  Suponha que $\Gamma \models P$ e seja $I = \langle \Delta, w, s \rangle$ uma LQC-interpretação para $\Gamma \cup \{\forall x P\}$ que satisfaz Γ.
  Temos que I também é uma LQC-interpretação para $\Gamma \cup \{P\}$, daí I satisfaz P, ou seja, $I_V(P) = \mathrm{v}$.
  Dado $d \in \Delta$, temos que I e $I(x|d)$ coincidem em Γ. (1)
  Note também que $I(x|d)$ é uma LQC-interpretação para $\Gamma \cup \{P\}$.
  De (1) e 5.2.23, $I(x|d)$ satisfaz Γ, logo $I(x|d)$ satisfaz P, ou seja, $I(x|d)_V(P) = \mathrm{v}$.
  Mostramos que, para cada $d \in \Delta$, $I(x|d)_V(P) = \mathrm{v}$, logo $I_V(\forall x P) = \mathrm{v}$, donde I satisfaz $\forall x P$, logo, $\Gamma \models \forall x P$.
- Queremos mostrar que o esquema do $\forall$-eliminação é correto.
  Seja '$\forall x P \vdash P(x|t)$' um exemplar do esquema do $\forall$-eliminação.
  Seja $I = \langle \Delta, w, s \rangle$ uma LQC-interpretação para $\{\forall x P\} \cup \{P(x|t)\}$ que satisfaz $\forall x P$, daí, para cada $d \in \Delta$, $I(x|d)_V(P) = \mathrm{v}$.
  Como $I_D(t) \in \Delta$, daí $I(x|I_D(t))_V(P) = \mathrm{v}$, donde, pelo lema 3.3, item (ii), $I_V(P(x|t)) = \mathrm{v}$, ou seja, I satisfaz $P(x|t)$, portanto $\forall x P \models P(x|t)$.
- Queremos mostrar que o esquema do $\exists$-introdução é correto.
  Seja '$P(x|t) \vdash \exists x P$' um exemplar do esquema do $\exists$-introdução.
  Seja $I = \langle \Delta, w, s \rangle$ uma LQC-interpretação para $\{P(x|t) \cup \{\exists x P\}\}$ que satisfaz $P(x|t)$, daí $I_V(P(x|t)) = \mathrm{v}$, donde pelo lema 3.3, item (ii), $I(x|I_D(t))_V(P) = \mathrm{v}$, ou seja, existe $d \in \Delta (d = I_D(t))$ tal que $I(x|d)_V(P) = \mathrm{v}$, logo $I_V(\exists x P) = \mathrm{v}$, ou seja, I satisfaz $\exists x P$, portanto $P(x|t) \models \exists x P$.
- Queremos mostrar que a regra do $\exists$-eliminação é correta.
  Seja $\dfrac{\Gamma \vdash \exists x P \qquad \Gamma \cup P(x|y) \vdash Q}{\Gamma \vdash Q}$ uma aplicação da regra do $\exists$-eliminação.
  Daí y não é livre em $\Gamma, \exists x P, Q$.
  Suponha que $\begin{cases} \Gamma \models \exists x P, \\ \Gamma \cup P(x|y) \models Q. \end{cases}$

Seja $I = \langle \Delta, w, s \rangle$ uma LQC-interpretação para $\Gamma \cup \{Q\}$ que satisfaz Γ.
Seja w' um mundo sobre Δ, tal que $\mathcal{D}(w') = \mathcal{D}(w) \cup SNL(P)$[13] tal que w' é uma extensão de w.
Seja $I' = \langle \Delta, w', s \rangle$.
Daí I' é uma LQC-interpretação para P.
Como I e I' coincidem em Γ, temos pelo lema 5.2.23, que I' satisfaz Γ, donde I' satisfaz $\exists x P$, ou seja, existe $d \in \Delta$ tal que $I'(x|d)_V(P) = \text{v}$.
Seja s' uma Δ-atribuição para variáveis, temos que $s' = s(y|d)$.
Seja $I'' = \langle \Delta, w', s' \rangle$ uma LQC-interpretação.
Como I' e I'' coincidem em Γ, pois y não é livre em Γ, temos que I'' satisfaz Γ.
Temos que $I'(x|d)$ e $I''(x|d)$ coincidem em P, daí, pelo lema 5.2.23, $I''(x|d)_V(P) = \text{v}$, e daí, como $I''_D(y) = d$, temos que $I''(x|I''_D(y))_V(P) = \text{v}$, ou seja, $I''_V(P(x|y)) = \text{v}$, logo $I''_V(Q) = \text{v}$.
Como I' e I'' coincidem em Q, daí $I'_V(Q) = \text{v}$.
Como I e I' coincidem no Q, temos, pelo lema 5.2.23, que $I_V(Q) = \text{v}$.
Portanto $\Gamma \models Q$.

$\square$

3.2 Completude do cálculo de sequentes de LQC com respeito à semântica de LQC

Mostraremos aqui que o cálculo LQC é completo com respeito à semântica de LQC.

O lema abaixo elucida as condições que devem ser atendidas por coleções que são ao mesmo tempo saturadas e de Henkin.

Uma coleção que é P-saturada em uma linguagem L com respeito a LQC, também é coleção de Henkin em L com respeito a LQC, e possui as seguintes propriedades:

3.5 Lema. *Seja φ P-saturado em L com respeito a* LQC *e φ coleção de Henkin em L com respeito a* LQC. *Então φ possui as seguintes propriedades:*

(i) *Considere os demais casos já apresentamos em* LPC*, lema 9.3.5.*
(ii) *$\forall x P \in \varphi$ sss para todo termo t de L, $P(x|t) \in \varphi$.*
(iii) *$\exists x P \in \varphi$ sss existe um termo t de L tal que $P(x|t) \in \varphi$.*

Prova:
Assuma a hipótese.
Pelo lema 7.4.13 e (ii), já temos que para todo termo t de L, $P(x|t) \in \varphi$ se $\forall x P \in \varphi$.

[13] Onde SNL são os *s*inais *n*ão *l*ógicos em P.

Pela definição 7.4.21 e (iii), já temos que $\exists xP \in \varphi$ se existe um termo t de L tal que $P(x|t) \in \varphi$.
Resta provar que:
(i) $\forall xP \in \varphi$ se para todo termo t de L, $P(x|t) \in \varphi$,
(ii) existe um termo t de L tal que $P(x|t) \in \varphi$ se $\exists xP \in \varphi$.
Suponha que $\forall xP \in \varphi$ e t é termo de L.
Daí $P(x|t)$ é fórmula de L.
Pelo $\forall$-el, $\forall xP \vdash P(x|t)$, logo $\varphi \vdash P(x|t)$, daí, por 7.4.18, $P(x|t) \in \varphi$.
Suponha agora que, para um termo t de L, $P(x|t) \in \varphi$.
Daí $\exists xP$ é fórmula de L.
Pelo $\exists$-int, $P(x|t) \vdash \exists xP$, daí $\varphi \vdash \exists xP$, donde, por 7.4.18, $\exists xP \in \varphi$. $\square$

Vamos mostrar que se φ for P-saturado em L e coleção de Henkin em L com respeito a LQC, então ele é satisfatível.

3.6 Lema.

- *Sejam* $\begin{cases} \varphi\ P\text{-saturado em } L \text{ com respeito a } \mathrm{LQC}, (1) \\ \varphi \text{ coleção de Henkin em } L, (2) \end{cases}$ *então* φ *é* LQC-*satisfatível.*

Prova:
Assuma a hipótese.
Precisamos definir uma interpretação, segundo a definição 5.2.12 e mostrar que ela satisfaz todas as fórmulas de φ.
Seja Δ a coleção de todos os termos de L.
Seja w um mundo sobre Δ, cujo domínio é a coleção de constantes, sinais funcionais e sinais predicativos do alfabeto de L, satisfazendo as seguintes condições:

(i) para cada constante c de L, $w(c) = c$.
(ii) para cada sinal funcional n-ário f de L, $\begin{cases} w(f) : \Delta^n \to \Delta, \\ \langle t_1, \ldots, t_n \rangle \vdash f(t_1, \ldots, t_n). \end{cases}$
(iii) para cada sinal predicativo n-ário p de L, $w(p) = \{\langle t_1, \ldots, t_n \rangle | t_1, \ldots, t_n\}$ são termos de L e $p(t_1, \ldots, t_n) \in \varphi$.
(iv) Seja s uma Δ-atribuição para variáveis tal que $s(x) = x$ para qualquer variável x.

Considere a LQC-interpretação $I = \langle \Delta, w, s \rangle$.
Temos que I é uma LQC-interpretação para φ, porque φ é uma coleção de fórmulas em L.
É fácil verificar que, para todo termo t de L, $I_D(t) = t$.
Vamos agora mostrar por indução sobre o grau de fórmula

em LQC que I satisfaz todas as fórmulas de φ, ou seja:
$\begin{cases} Q \in \varphi \text{ implica que } I_V(Q) = \text{v}, \\ \neg Q \in \varphi \text{ implica que } I_V(Q) = \text{f}. \end{cases}$
Com isso, φ será satisfatível.
Então, continuando a prova:

- O caso em que Q é $\top$ ou $\bot$ são óbvios. Toda interpretação atribui ao verum o valor verdadeiro e ao falsum o valor falso.
- Caso Q é $p(t_1, \ldots, t_n)$.
  Se $p(t_1, \ldots, t_n) \in \varphi$, então $\langle t_1, \ldots, t_n \rangle \in w(p)$, daí $\langle I_D(t_1), \ldots, I_D(t_n) \rangle \in w(p)$, isso equivale a dizer, pela semântica de LQC, que $I_V(p(t_1, \ldots, t_n)) = \text{v}$. Se $\neg p(t_1, \ldots, t_n) \in \varphi$, daí como φ é $\neg$-consistente, então $p(t_1, \ldots, t_n) \notin \varphi$, donde $\langle t_1, \ldots, t_n \rangle \notin w(p)$, daí $\langle I_D(t_1), \ldots, I_D(t_n) \rangle \notin w(p)$, logo $I_V(p(t_1, \ldots, t_n)) = \text{f}$.
- Os casos em que Q é $P \wedge S$, $P \vee S$ ou $P \to S$ já foram feitos na prova de completude para LPC em 9.3.7.
- Caso Q é $\forall x R$.
  Se $\forall x R \in \varphi$, então, pelo lema 3.5, para todo termo t de L, $R(x|t) \in \varphi$.
  Por HI sobre o grau de fórmula, para todo termo t de L, $I_V(R(x|t)) = \text{v}$.
  Mas $I_V(R(x|t)) = I(x|I_D(t))_V(R)$.
  Ou seja, para qualquer termo t de L, $I(x|I_D(t))_V(R) = \text{v}$.
  Que é igual a qualquer termo t de L, $I(x|t)_V(R) = \text{v}$.
  Que corresponde a qualquer $d \in \Delta$, $I(x|d)_V(R) = \text{v}$.
  Que é $I_V(\forall x R) = \text{v}$.
  Se $\neg \forall x R \in \varphi$, então como φ é $\neg$-consistente, $\forall x R \notin \varphi$.
  Daí pelo lema 3.5, existe um termo t de L tal que $R(x|t) \notin \varphi$, donde como φ é $\neg$-completo em L, $\neg R(x|t) \in \varphi$, logo, por HI, $I_V(R(x|t)) = \text{f}$, daí $I(x|I_D(t))(R) = \text{f}$, ou seja, existe um termo t de L tal que $I(x|I_D(t))_V(R) = \text{f}$, ou seja, $I_V(\forall x R) = \text{f}$.
- Caso Q é $\exists x R$.
  Se $\exists x R \in \varphi$, então, pelo lema 3.5, existe um termo t de L, tal que $R(x|t) \in \varphi$.
  Por HI sobre o grau de fórmula, existe um termo t de L, $I_V(R(x|t)) = \text{v}$.
  Mas $I_V(R(x|t)) = I(x|I_D(t))_V(R)$.
  Ou seja, existe um termo t de L onde $I(x|I_D(t))_V(R) = \text{v}$.
  Que é igual a existe $d \in \Delta$ onde $I(x|d)_V(R) = \text{v}$.
  Que corresponde $I_V(\exists x R) = \text{v}$. Se $\neg \exists x R \in \varphi$, então como φ é $\neg$-consistente, $\exists x R \notin \varphi$.
  Daí pelo lema 3.5, existe um termo t de L tal que $R(x|t) \notin \varphi$, donde como φ é $\neg$-completo em L, $\neg R(x|t) \in \varphi$, logo, por HI, $I_V(R(x|t)) = \text{f}$, daí $I(x|I_D(t))(R) = \text{f}$, ou seja, existe um termo t de L tal que $I(x|I_D(t))_V(R) = \text{f}$, ou seja, $I_V(\exists x R) = \text{f}$.

□

Com esses resultamos acabamos de demonstrar que φ é um conjunto LQC-satisfatível, considerando uma interpretação que rigorosamente satisfaz φ e utilizando ideias de conjunto P-saturado e coleção de Henkin.

Falta mostrar que se um conjunto inicial Γ em LQC não infere P, então ele pode ser estendido para uma coleção φ P-saturada e de Henkin.

3.7 Lema.
Seja Γ *uma coleção de fórmulas em* LQC *tal que* $\Gamma \not\vdash P$.
Então existe uma coleção de fórmulas φ *em* LQC *e existe uma linguagem* L *para* LQC

tal que $\begin{cases} \Gamma \text{ e } \varphi \text{ são coleções de fórmulas de } L, \\ \Gamma \subseteq \varphi, \\ \varphi \text{ é } P\text{-saturado em } L \text{ com respeito a LQC.} \end{cases}$

Para provar esse lema precisamos de alguns resultados de teoria dos conjuntos, tais como recursos de números ordinais, cardinais e correspondência biunívoca, dentre outros.

Prova:
Seja L_0 a linguagem para LQC cujos sinais não lógicos são os que figuram em Γ, P. Então existe um único cardinal η que está em correspondência biunívoca com a coleção de todas as fórmulas de L_0.
Para cada ordinal $\lambda \in \eta$, considere uma nova constante c_λ não figurando em Γ e em P [14].
Seja L_1 a linguagem para LQC cujos sinais não lógicos são os de L_0 mais uma infinidade de novas constantes c_λ, para cada $\lambda \in \eta$.
Como existem η sinais não lógicos em L_1 e η é infinito, temos então, pelo lema 5.1.15, que a cardinalidade da coleção de fórmulas de L_1 é η.
Então existe uma correspondência biunívoca, que a cada ordinal $\lambda \in \eta$ associa uma fórmula Q_λ de L_1.

[14] Durante a prova, fórmulas existenciais podem ser encontradas em Γ ou em uma extensão de Γ. Então temos que conseguir uma constante que seja testemunha de uma fórmula existencial $\exists x P$, assim existirá um objeto no universo de discurso tal que considerando o x como sendo esse objeto, é possível obter um $P(x|c)$. Só que essa constante c pode não existir na linguagem L_0, ou seja, o estoque de constantes de L_0 pode não servir ou ser suficiente para testemunhar, então temos que acrescentar uma quantidade de constantes a L_0 igual à cardinalidade da coleção de fórmulas de L_0. L_0 tem η fórmulas, então acrescentamos η constantes. Isso garante a existência de testemunhas em L_0, mesmo que seja redundante em alguns casos. Vamos portanto, enriquecer o alfabeto de L_0: cada ordinal que é elemento de η vai marcar uma fórmula de L_0. Acrescentaremos, para cada ordinal elemento de η, uma nova constante c_λ que não figura em Γ. Essas novas constantes serão acrescentadas no alfabeto de L_0, criando assim uma linguagem L_1. Neste ponto ainda não sabemos se Γ é uma coleção de Henkin, então com essas ideias temos que conseguir criar um Γ que possua as características de coleção de Henkin. E tomamos c_λ como uma constante nova, pois para as constantes antigas não podemos impor certas propriedades.

Vamos agora associar cada ordinal $\lambda \in \eta$ a uma coleção Ψ_λ de fórmulas de L_1, segundo as cláusulas:

(i) $\Psi_0 = \Gamma$.

(ii)

$$\Psi_{\lambda+1}{}^{15} = \begin{cases} \Psi_\lambda, \text{ se } \Psi_\lambda \cup \{Q_\lambda\} \vdash P. \\ \Psi_\lambda \cup \{Q_\lambda\}, \text{ se } \Psi_\lambda \cup \{Q_\lambda\} \nvdash P \text{ e } Q_\lambda \text{ não é fórmula existencial.} \\ \Psi_\lambda \cup \{Q_\lambda, R(x|c_i)\}, \text{ se } \begin{cases} \bullet\ \Psi_\lambda \cup \{Q_\lambda\} \nvdash P, \\ \bullet\ Q_\lambda \text{ é da forma } \exists x R, \\ \bullet\ \text{existe } j \in \eta \text{ tal que } c_j \text{ não figura} \\ \quad \text{em } \Psi_\lambda \cup \{Q_\lambda\}, \text{ onde } i \text{ é o menor} \\ \quad \text{ordinal } j \in \eta \text{ tal que } c_j \text{ não figura} \\ \quad \text{em } \Psi_\lambda \cup \{Q_\lambda\}. \end{cases} \\ \Psi_\lambda \cup \{Q_\lambda\}, \text{ se } \begin{cases} \bullet\ \Psi_\lambda \cup \{Q_\lambda\} \nvdash P, \\ \bullet\ Q_\lambda \text{ é da forma } \exists x R, \\ \bullet\ \text{não existe } j \in \eta \text{ tal que } c_j \text{ não figura} \\ \quad \text{em } \Psi_\lambda \cup \{Q_\lambda\}, \text{ onde } i \text{ é o menor} \\ \quad \text{ordinal } j \in \eta \text{ tal que } c_j \text{ não figura em} \\ \quad \Psi_\lambda \cup \{Q_\lambda\}. \end{cases} \end{cases}$$

(iii) Se λ é ordinal limite, então $\Psi_\lambda = \bigcup_{i<\lambda} \Psi_i$.

Seja C_λ a coleção de novas constantes que figuram em Ψ_λ e Q_λ.
Queremos mostrar que, para cada $\lambda \in \eta$, $\#(C_\lambda)$ é finito ou $\#(C_\lambda) \leq \#(\lambda)$.
Como Γ não possui novas constantes e $\Psi_0 = \Gamma$, temos que $\#(C_0)$ é o número de novas constantes de Q_0, que é finito.
Suponha, por HI, que dado $\lambda \in$
eta, $\#(C_\lambda)$ é finito ou $\#(C_\lambda) \leq \#(\lambda)$.
Temos que $\Psi_{\lambda+1} \cup \{Q_{\lambda+1}\}$ contém, no máximo, duas fórmulas a mais que $\Psi_\lambda \cup \{Q_\lambda\}$, donde $C_{\lambda+1}$ possui uma quantidade finita de novas constantes a mais que C_λ, logo, se C_λ é finito, então $C_{\lambda+1}$ também é finito, e se C_λ é infinito, daí $\#(C_\lambda) \leq \#(\lambda)$, então λ é infinito e

$$\begin{cases} \#(C_{\lambda+1}) = \#(C_\lambda), \\ \#(\lambda) = \#(\lambda+1), \end{cases}$$

portanto $\#(C_{\lambda+1}) \leq \#(\lambda+1)$.
Considere agora que λ é ordinal limite e que esta propriedade vale para qualquer ordinal $i < \lambda$.
Por HI, se $i < \lambda$, $\#(C_i)$ é finito ou $\#(C_i) \leq \#(i)$.
Suponha que $i < \lambda$, se $\#(C_i)$ é finito, então, como λ é ordinal infinito, temos que $\#(C_i) < \#(\lambda)$.
Se $\#(C_i) \leq \#(i)$, então como $\#(i) \leq \#(\lambda)$, porque $i \in \lambda$, então $\#(C_i) \leq \#(\lambda)$.
Daí, em qualquer caso, se $i < \lambda$, então $\#(C_i) \leq \#(\lambda)$, donde

[15] Ordinal sucessor.

$\sup_{i \in \lambda}(\#(C_i)) \leq \#(\lambda)$, portanto $\#(C_\lambda) \leq \#(\lambda)$.
Acabamos de mostrar que, para cada $\lambda \in \eta$, $\#(C_\lambda)$ é finito ou $\#(C_\lambda) \leq \#(\lambda)$.
Temos que, para cada $\lambda \in \eta$, a quantidade de novas constantes de $\Psi_\lambda \cup Q_\lambda$ é finita ou menor ou igual a $\#(\lambda)$, donde sempre existe $j \in \eta$ tal que c_j não figura em $\Psi_\mu \cup \{Q_\lambda\}$, portanto temos que, se $\begin{cases} \Psi_\lambda \cup \{Q_\lambda\} \not\vdash P, \\ Q_\lambda \text{ é a forma } \exists x R, \end{cases}$
então $\Psi_{\lambda+1} = \Psi_\lambda \cup \{Q_\lambda,\ R(x|c_i)\}$, onde i é o menor ordinal j tal que $j \in \eta$ e c_j não figura em $\Psi_\lambda \cup \{Q_\lambda\}$.

Vamos mostrar agora que, para cada $\lambda \in \eta$, $\Psi_\lambda \not\vdash P$.
Caso $\Psi_0 = \Gamma$, $\Psi_0 \not\vdash P$. Logo vale para Ψ_0.
Suponha, por HI, que $\Psi_\lambda \not\vdash P$.
Se $\Psi_\lambda, \{Q_\lambda\} \vdash P$ e Q_λ não é existencial, então $\Psi_{\lambda+1} = \Psi_\lambda$, daí $\Psi_{\lambda+1} \not\vdash P$.
Se $\Psi_\lambda, \{Q_\lambda\} \not\vdash P$ e Q_λ é da forma $\exists x R$, então $\Psi_{\lambda+1} = \Psi_\lambda \cup \{Q_\lambda,\ R(x|c_i)\}$, onde c_i não figura em $\Psi_\lambda \cup \{Q_\lambda\}$.
Se $\Psi_\lambda, \exists x R,\ R(x|c_i) \vdash P$, então pelo lema 5.3.7, $\Psi_\lambda \cup \{Q_\lambda\} \vdash P$, o que é absurdo, logo $\Psi_\lambda, Q_\lambda,\ R(x|c_i) \not\vdash P$, ou seja, $\Psi_{\lambda+1} \not\vdash P$.

Considere agora que $\lambda \in \eta$ é ordinal limite, e, por HI, que para cada $i \in \lambda$, $\Psi_i \not\vdash P$.
Suponha por absurdo que $\Psi_\lambda \vdash P$, então existiria um certo subconjunto finito Υ_λ de Ψ_λ tal que $\Upsilon \vdash P$, pela lei da compacidade (em 9.2.1).
Sendo Υ finito, ele teria uma certa quantidade de fórmulas $\Upsilon = \{R_1, \ldots, R_n\}$, daí $\{R_1, \ldots, R_n\} \vdash P$.
$\{R_1, \ldots, R_n\} \in \Psi_\lambda$ e Ψ_λ sendo ordinal limite, ele é a união de todos os Ψ_i,
então $\begin{cases} R_1 \in \Psi_{i_1} \\ R_2 \in \Psi_{i_2} \\ \ldots \\ R_n \in \Psi_{i_n}. \end{cases}$
Então $\{R_1, \ldots, R_n\} \in \Psi_j$, onde Ψ_j é o máximo de $i_1, \ldots, i_n$, ou seja, Ψ_j é um superconjunto de $\{R_1, \ldots, R_n\}$, então $\Psi_j \vdash P$, o que é absurdo, porque $j < \lambda$, então ele não poderia acarretar o P.
Então, para cada $\lambda \in \eta$ concluímos que $\Psi_\lambda \not\vdash P$. Agora vamos considerar a união dos Ψ_λ's e chamar de φ.
E φ é o candidato a a ser um conjunto P-saturado e de Henkin, que é um superconjunto de Γ.

Vamos iniciar que φ é um conjunto P-saturado.
Então seja $\varphi = \bigcup\limits_{\lambda \in \eta} \Psi_\lambda$.
Se $\varphi \vdash P$, então existe $\Upsilon \subseteq \varphi$ e Υ é finito, tal que $\Upsilon \vdash P$.
Temos que Υ é da forma $\{R_1, \ldots, R_n\}$, ou seja, $\{R_1, \ldots, R_n\} \vdash P$.
Temos que existem ordinais $i_1, \ldots, i_n \in \eta$ tal que
$\begin{cases} R_i \in \Psi_{i_1} \\ \ldots \\ R_n \in \Psi_{i_n}. \end{cases}$
Sendo $j = \mathbf{max}\{i_1, \ldots, i_n\}$, temos que $R_{i_1}, \ldots, R_{i_n} \in \Psi_j$, logo, $\Psi_j \vdash P$, o que é absurdo, portanto $\varphi \nvdash P$.

Suponha que $\begin{cases} \varphi' \text{ é a coleção de fórmulas de } L_1, \\ \varphi' \nvdash P, \\ \varphi \subseteq \varphi'. \end{cases}$
Suponha que $R \in \varphi'$.
Como R é fórmula de L_1, existe um $\lambda \in \eta$ tal que $R = Q_\lambda$.
$\Psi_\lambda \subseteq \varphi$, daí $\Psi_\lambda \subseteq \varphi'$, donde $\Psi_\lambda \cup \{R\} \subseteq \varphi'$, logo $\Psi_\lambda \cup \{R\} \nvdash P$, ou seja, $\Psi_\lambda \cup \{Q_\lambda\} \nvdash P$, donde $\Psi_{\lambda+1} = \Psi_\lambda \cup \{Q_\lambda\}$.
Mas $Q_\lambda \in \Psi_{\lambda+1}$ e $\Psi_{\lambda+1} \subseteq \varphi$, logo $Q_\lambda \in \varphi$, mas Q_λ é R, portanto $R \in \varphi$.
Daí $\varphi' \subseteq \varphi$.
Portanto, φ é P-saturado em L_1 com respeito a LQC.

Vamos agora mostrar que φ é coleção de Henkin.
Suponha que $\exists x R \in \varphi$.
Como $\exists x R$ é fórmula de L_1, mas toda fórmula de L_1 possui m ordinal, então existe $\lambda \in \eta$ tal que $Q_\lambda = \exists x R$.
Como $\Psi_\lambda \cup \{\exists x R\} \subseteq \varphi$, daí $\Psi_\lambda, \exists x R \nvdash P$, daí $\Psi_{\lambda+1} = \Psi_\lambda \cup \{\exists x R,\ R(x|c_i)\}$.
Donde, existe uma constante em L_1 tal que $R(x|c) \in \varphi$.
Suponha que, para toda constante c em L_1, $R(x|t) \in \varphi$.
Daí, pelo lema 3.5, φ é $\neg$-completo em L_1, donde $\forall x P \in \varphi \vee \neg\forall x P \in \varphi$.
Suponha por absurdo que $\neg\forall x P \in \varphi$, daí $\varphi \vdash \exists x \neg P$, logo, pelo lema 7.4.18, $\exists x \neg R \in \varphi$, daí, conforme o que acabamos de mostrar acima, existe uma nova constante c tal que $(\neg R)(x|c) \in \varphi$, ou seja, $\neg R(x|c) \in \varphi$.
Temos também que $R(x|c) \in \varphi$, daí, $\varphi \vdash P$, o que é absurdo, portanto $\neg\forall x R \notin \varphi$, donde $\forall x R \in \varphi$.
Portanto φ é coleção de Henkin em L_1. $\square$

3.8 Lema.

Sejam $\begin{cases} L \textit{ uma linguagem para } \mathrm{LQC}, \\ \Sigma \textit{ a coleção de sinais não lógicos de } L, \\ \mathcal{F} \textit{ a coleção de fórmulas de } L. \end{cases}$

(i) *Se* Σ *é finito, então* $\#(\mathcal{F}) = \aleph_0$.

(ii) *Se* Σ *é infinito, então* $\#(\mathcal{F}) = \#(\Sigma)$.

Seja Σ_1 *a coleção de sinais lógicos de* L *(conectivos, quantificadores, variáveis e os sinais de pontuação).*
Seja $\Sigma_2 = \Sigma \cup \Sigma_1$*, onde* Σ_2 *é o alfabeto completo de* L*.*
Seja S *a coleção de* Σ_2*-samblagens finitas.*

Prova de (i):
Temos que $\#(\Sigma_2) = \#(\Sigma) + \#(\Sigma_1)$.
Mas Σ é finito e Σ_1 é infinito, então $\#(\Sigma_2) = \#(\Sigma) + \#(\Sigma_1) = \aleph_0$, ou seja, existem $\aleph_0$ sinais no alfabeto, daí, $\#(S) = \#(\Sigma_2)$.
Como toda fórmula é uma Σ_2-samblagem, temos que $\mathcal{F} \subseteq S$, daí $\#(\mathcal{F}) \leq \#(S)$, ou seja, o cardinal da coleção de todas as fórmulas em L deve ser menor ou igual ao cardinal da coleção de todas as samblagens no alfabeto de L, mas $\#(S) = \aleph_0$, então $\#(\mathcal{F}) \leq \aleph_0$.
Por outro lado, considere uma função de $\mathbb{N}$ em $\mathcal{F}$ que associa cada $n \in \mathbb{N}$ à fórmula $\neg \ldots \neg \top$, onde $\neg \ldots \neg$ são n sequências do conectivo da negação, $\neg$.
Temos aí uma função injetiva de $\mathbb{N}$ em $\mathcal{F}$, então $\#(\mathbb{N}) \leq \#(\mathcal{F})$, ou seja $\#(\mathcal{F}) \geq \#(\mathbb{N})$, mas $\#(\mathbb{N}) = \aleph_0$, então $\#(\mathcal{F}) \geq \aleph_0$, mas $\#(\mathcal{F}) \leq \aleph_0$, então $\#(\mathcal{F}) = \aleph_0$. □

Prova de (ii):
$\#(\Sigma_2) = \#(\Sigma) + \#(\Sigma_1) = \#(\Sigma)$, daí $\#(S) = \#(\Sigma)$.
Como toda fórmula é uma S-samblagem, temos que $\mathcal{F} \subseteq S$, daí $\#(\mathcal{F}) \leq \#(\Sigma)$.
Agora vamos mostrar o contrário, ou seja, que para cada sinal não lógico existe uma fórmula e vamos fazer uma função injetiva da coleção de sinais não lógicos para a coleção de fórmulas, isso se existir pelo menos um sinal predicativo de aridade positiva.
Considere a seguinte função f de Σ em $\mathcal{F}$, que para cada constante c, para cada sinal funcional f e para cada sinal predicativo p, ambos do alfabeto de L, ela associa uma fórmula, de modo que todas as fórmulas obtidas e associadas sejam diferentes para cada sinal não lógico.
Seja p um sinal predicativo de L de aridade n, onde $n > 0$.

$c \mapsto p(c, x, \ldots, x)$, sendo $n - 1$ ocorrências de $x \ldots x$.
$f \mapsto p(f(x, \ldots, x), x, \ldots, x)$, sendo $x, \ldots, x$ um número de ocorrências de x igual à aridade de f.

$q \mapsto q(x, \ldots, x)$.

Temos daí uma função injetiva de Σ em $\mathcal{F}$, daí $\#(\Sigma) \leq \#(\mathcal{F})$, mas também vimos que $\#(\mathcal{F}) \leq \#(\Sigma)$, então $\#(\mathcal{F}) = \#(\Sigma)$. $\square$

3.9 Definição.
Seja φ uma coleção P-saturada em L com respeito a LQC tal que φ é coleção de Henkin em L.
Dizemos que a interpretação que satisfaz φ definida no lema 3.6 é a LQC-*interpretação canônica para* φ[16].

3.10 Lema.
Seja φ uma coleção P-saturada em L com respeito a LQC *tal que φ é coleção de Henkin em L.*
Temos então que a LQC-*interpretação canônica I para φ é a função característica (ver 2.3.85), de φ com respeito a L, isto é:*
$I_V(Q) = \begin{cases} \text{v, } se\ Q \in \varphi, \\ \text{f, } se\ Q \notin \varphi. \end{cases}$

3.11 Lema.
O $I_D(t)$ de uma interpretação canônica é igual ao t, ou seja, $I_D(t) = t$.

Prova:
Seja $I = \langle \Delta, w, s \rangle$, onde Δ, w, s são conforme foi definido no lema 3.6. Vamos mostrar, por indução sobre grau de fórmulas, para qualquer fórmula Q de L, que $\begin{cases} Q \in \varphi \text{ implica que } I_V(Q) = \text{v}, \\ \neg Q \in \varphi \text{ implica que } I_V(Q) = \text{f}. \end{cases}$

- Caso Q é $p(t_1, \ldots, t_n)$.
  Se $p(t_1, \ldots, t_n) \in \varphi$, então $\langle t_1, \ldots, t_n \rangle \in w(p)$, daí $I_V(p(t_1, \ldots, t_n)) = \text{v}$.
  Se $\neg p(t_1, \ldots, t_n) \in \varphi$, então pelo lema 3.5, $p(t_1, \ldots, t_n) \notin \varphi$, daí $\langle t_1, \ldots, t_n \rangle \notin w(p)$, donde
  $I_V(p(t_1, \ldots, t_n)) = \text{f}$.
- Caso Q é $\neg R$.
  Se $\neg R \in \varphi$, então por HI, $I_V(R) = \text{f}$, donde $I_V(\neg R) = \text{v}$.
  Se $\neg\neg \in \varphi$, temos que $\varphi \vdash R$, daí pelo lema 7.4.18, $R \in \varphi$, donde por HI, $I_V(R) = \text{v}$, logo $I_V(\neg R) = \text{f}$.
- Caso Q é $R \to S$.
  Se $R \to S \in \varphi$, então pelo lema 3.5, $R \notin \varphi$ ou $S \in \varphi$, daí ainda pelo lema 3.5, $\neg R \in \varphi$ ou $S \in \varphi$, daí, por HI, $I_V(R) = \text{f}$ ou $I_V(S) = \text{v}$, donde $I_V(R \to S) = \text{v}$.
  Se $\neg(R \to S) \in \varphi$, então $R \to S \notin \varphi$, daí $R \in \varphi$ e $S \notin \varphi$, donde, pelo lema

[16] Canônica no sentido em que pode ser obtida naturalmente ou de imediato.

3.5, $R \in \varphi$ e $\neg S \in \varphi$, logo, por HI, $I_V(R) = $ v e $I_V(S) = $ f, portanto $I_V(R \to S) = $ f.

- Caso Q é $R \vee S$.
  Se $R \vee S \in \varphi$, então pelo lema 3.5, $R \in \varphi$ e $S \in \varphi$, daí, por HI, $I_V(R) = $ v e $I_V(S) = $ v, donde $I_V(R \vee S) = $ v.
  Se $\neg(R \vee S) \in \varphi$, então $R \vee S \notin \varphi$, daí $R \notin \varphi$ e $S \notin \varphi$, donde pelo lema 3.5, $\neg R \in \varphi$ e $\neg S \in \varphi$.
  Logo, por HI, $I_V(R) = $ f e $I_V(S) = $ f, portanto $I_V(R \vee S) = $ f.
- Caso Q é $R \wedge S$.
  Se $R \wedge S \in \varphi$, então pelo lema 3.5, $R \in \varphi$ e $S \in \varphi$, daí, por HI, $I_V(R) = $ v e $I_V(S) = $ v, donde $I_V(R \wedge S) = $ v.
  Se $\neg(R \wedge S) \in \varphi$, então $R \wedge S \notin \varphi$, daí $R \notin \varphi$ e $S \notin \varphi$, donde pelo lema 3.5, $\neg R \in \varphi$ e $\neg S \in \varphi$.
  Logo, por HI, $I_V(R) = $ f ou $I_V(S) = $ f, portanto $I_V(R \wedge S) = $ f.
- Caso Q é $\forall x R$.
  Se $\forall x R \in \varphi$, então pelo lema 3.5, como φ é coleção de Henkin em L, temos que $R(x|t) \in \varphi$, para qualquer termo t de L.
  Por HI, temos que $I_V(R(x|t)) = $ v, para qualquer termo t de L.
  Daí, $I(x|t)_V(R) = $ v, para qualquer termo t de L, segundo o teorema 1.1 e o lema 3.11, logo $I_V(\forall x R) = $ v.
  Se $\neg\forall x R \in \varphi$, então como φ é $\neg$-consistente, pelo lema 3.5, $\forall x R \notin \varphi$, donde ainda pelo lema 3.5, existe um termo t de L tal que $R(x|t) \notin \varphi$, daí $\neg R(x|t) \in \varphi$, donde, por HI, $I_V(R(x|t)) = $ f, logo pela definição 1.1, $I(x|t)_V(R) = $ f, para algum termo t de L, ou seja, $I_V(\forall x R) = $ f.
- Caso Q é $\exists x R$.
  Se $\exists x R \in \varphi$, então pelo lema 3.5, como φ é coleção de Henkin em L, temos que $R(x|t) \in \varphi$, para um termo t de L.
  Por HI, temos que $I_V(R(x|t)) = $ v, para um termo t de L.
  Daí, $I(x|t)_V(R) = $ v, para um termo t de L, segundo as definições 1.1 e 3.9, se tem interpretação canônica, então $I_D(t) = t$, logo $I_V(\exists x R) = $ v.
  Se $\neg\exists x R \in \varphi$, então como φ é $\neg$-consistente, pelo lema 3.5, $\exists x R \notin \varphi$, donde ainda pelo lema 3.5, existe um termo t de L tal que $R(x|t) \notin \varphi$, daí $\neg R(x|t) \in \varphi$, donde, por HI, $I_V(R(x|t)) = $ f, logo pela definição 1.1, $I(x|t)_V(R) = $ f, para um termo t de L, ou seja, $I_V(\exists x R) = $ f.

Acabamos de mostrar para toda forma Q de L que
$$\begin{cases} Q \in \varphi, \text{ então } I_V(Q) = \text{v}, \\ \neg Q \in \varphi, \text{ então } I_V(Q) = \text{f}. \end{cases}$$

Agora suponha que Q é uma fórmula de L e $Q \notin \varphi$. Mas se $Q \notin \varphi$, então $\neg Q \in \varphi$ porque φ é $\neg$-completo em L, com respeito a LQC. Mas se $\neg Q \in \varphi$, então $I_V(Q) = $ f. $\square$

Com isso demonstramos que a função canônica é a função característica do φ, com respeito a L. *Característica* no sentido em que, se $Q \in \varphi$, então $I_V(Q) = \text{v}$ e se $Q \notin \varphi$, então $I_V(Q) = \text{f}$.

Podemos agora, provar a completude do cálculo de sequentes de LQC.

3.12 Teorema. ***(Completude)*** *Se* $\Gamma \models_{\text{LQC}} P$*, então* $\Gamma \vdash_{\text{LQC}} P$.

Prova:

Faremos uma prova não construtiva, então vamos supor a negação do consequente e tentar chegar à negação do antecedente.

Suponha $\Gamma \not\vdash P$, então Γ pode ser estendido para uma coleção saturada e de Henkin.

Então existe uma coleção de fórmulas em uma linguagem L tal que

$$\begin{cases} \bullet\ \Gamma \subseteq \varphi, \\ \bullet\ \varphi \text{ é } P\text{-saturado em } L \text{ com respeito a LQC}, \\ \bullet\ \varphi \text{ é coleção de Henkin em } L, \\ \bullet\ P \text{ é fórmula de } L. \end{cases}$$

Daí, pelo lema 3.6, φ é LQC-satisfatível.

Seja I a LQC-interpretação canônica para φ que satisfaz φ.

Daí pelo lema 3.10, temos que I é a função característica de φ com respeito a L, isto é: para cada fórmula Q de L, $I_V(Q) = \text{v}$ sss $Q \in \varphi$.

Como φ é P-saturado, temos que $\varphi \not\vdash P$, daí, pelo lema 7.4.18, $P \notin \varphi$ e P sendo fórmula de L, temos pelo lema 3.10, que $I_V(P) = \text{f}$.

I é uma interpretação para φ, então I também é interpretação para Γ, pois $\Gamma \subseteq \varphi$.

I satisfaz todas as fórmulas de φ, então I satisfaz todas as fórmulas de Γ.

Mas existe uma interpretação I para $\Gamma \cup \{P\}$ que satisfaz Γ e não satisfaz P, porque $I_V(P) = \text{f}$, portanto $\Gamma \not\models_{\text{LQC}} P$.

Provamos então, de forma não construtiva, que se $\Gamma \models_{\text{LQC}} P$, então $\Gamma \vdash_{\text{LQC}} P$. $\square$

§4. Uma Sistema de Tablôs para LQC

Um *Sistema de Tablôs por Refutação* para LQC é definido nesta seção e será notado por STR.

Considere, para esta seção, os conceitos gerais sobre semântica de valorações apresentados em §3.3 e a semântica de valorações para LQC descrita em §5.2.

4.1 Definição.

A *linguagem inicial* e a *linguagem de trabalho* em STR são iguais, e serão referenciadas, nesta seção, por L.

4.2 Definição. (Função de Inicialização)
A *função de inicialização de* STR associa cada fórmula P em LQC a um tablô contendo somente um nó, não marcado, cuja fórmula é obtida de P pela substituição de todas as variáveis livres de P por novas constantes, e eliminando todos os seus quantificadores vácuos.

4.3 Exemplo.
Para o sequente $P = \forall x\, p(x, z), \exists y\, r(x, z) \vdash s(x, w)$, as variáveis livres de P serão substituídas por novas constantes c_1, c_2, c_3 e os quantificadores vácuos serão eliminados.
O resultado será $P = \forall x\, p(x, c_1), r(c_2, c_1) \vdash s(c_2, c_3)$.
A seguir, o tablô inicial para P.

$$\begin{array}{c} \forall x\, p(x, c_1) \\ | \\ r(c_2|c_1) \\ | \\ \neg s(c_2|c_3) \end{array}$$

Figura 10.1: Tablô inicial para P

4.4 Exemplo.
Para o sequente $Q = \forall x\, r(z), \exists y\, s(w) \vdash \exists z \forall x\, t(z, x)$, os quantificadores vácuos serão eliminados.
O resultado será $Q = r(z), s(w) \vdash \exists z \forall x\, t(z, x)$.
A seguir, o tablô inicial para Q.

$$\begin{array}{c} r(z) \\ | \\ s(w) \\ | \\ \neg \exists z \forall x\, t(z, x) \end{array}$$

Figura 10.2: Tablô inicial para Q

4.5 Definição. (Critério de Fechamento)
Um ramo é fechado neste sistema sss o mesmo possuir duas fórmulas contraditórias ou uma ocorrência de $\perp$.

4.6 Definição. (**Ocorrência positiva de fórmula**)
P é positivo em Q se existir uma ocorrência de P em Q estando um número par de vezes sob o escopo de negações e antecedentes de implicação.

4.7 Definição. (**Ocorrência negativa de fórmula**)
P é negativo em Q se existir uma ocorrência de P em Q estando um número ímpar de vezes sob o escopo de negações e antecedentes de implicação.

4.8 Exemplo.
Seja $Q = \neg(p(x,y) \rightarrow r(x,z)) \vee (s(w) \rightarrow p(x,y))$.
Pela definição acima, $p(x,y)$ é positivo em Q, pois ocorre um par de vezes sendo uma sob o escopo de negação e outra sob o escopo do antecedente de implicação.

4.9 Definição. (**Coleção de fórmulas criativa**)
Uma coleção de fórmulas é criativa se ela possuir uma ocorrência positiva de fórmula existencial ou ocorrência negativa de fórmula universal.

4.10 Definição. (**Ramo criativo**)
Um ramo é criativo se as suas coleções de fórmulas não marcadas, forem criativas.

STR possui 16 regras, as quais trabalham com todos os tipos de fórmulas em LQC: implicação, conjunção, disjunção, equivalência, negação de implicação, negação de conjunção, negação de disjunção, negação de equivalência, negação de negação, fórmula universal, fórmula existencial (aqui existem dois casos, dependendo da coleção de fórmulas considerada), negação de fórmula universal e negação de fórmula existencial.

Destas, 11 regras foram apresentadas para o sistema de tablôs da Lógica Proposicional Clássica e continuam valendo para a Lógica Quantificacional Clássica. Serão especificadas a seguir, somente as regras de STR referentes aos quantificadores.

- **Fórmula Universal** ($\forall x\, P$), *quando houver pelo menos uma constante no ramo considerado*
  Neste caso, $c_1, \ldots, c_n$ são as constantes ocorrendo no ramo considerado. Se novas constantes são geradas, deve-se repetir a fórmula universal para todas as constantes que estão no ramo, para garantir que o método está completo. Essas novas constantes normalmente são geradas por ocorrência positiva de fórmula existencial ou ocorrência negativa de fórmula universal em nós não marcados.

$$
\begin{array}{c}
| \\
\vdots \\
\forall x\, P \surd \\
\vdots \\
P(x|c_1) \\
\vdots \\
P(x|c_n) \\
\vdots \\
\forall x\, P
\end{array}
$$

Figura 10.3: Fórmula Universal

- **Fórmula Universal** ($\forall x\, P$), *quando não houver constantes no ramo considerado*
  Neste caso, c_0 é uma constante fixa, escolhida arbitrariamente e que não ocorre no ramo considerado. Da mesma forma que a regra anterior, uma nova ocorrência do $\forall x\, P$ será incluída no ramo novamente se novas constantes aparecerem.

$$\begin{array}{c} | \\ \vdots \\ \forall x\, P \checkmark \\ \vdots \\ P(x|c_0) \\ \vdots \\ \forall x\, P \end{array}$$

Figura 10.4: Fórmula Universal

- **Fórmula Existencial** ($\exists x\, P$): Esta regra associa cada fórmula $\exists x\, P$ e cada coleção de fórmulas contendo $\exists x\, P$ a um tablô com somente um nó não marcado cuja fórmula é $P(x|c)$, onde c é uma constante não ocorrendo na coleção de fórmulas consideradas.

$$\begin{array}{c} | \\ \vdots \\ \exists x\, P \checkmark \\ \vdots \\ P(x|c) \\ \vdots \end{array}$$

Figura 10.5: Fórmula Existencial

- **Negação de Fórmula Universal** $(\neg\forall x\, P)$:
  Esta regra associa $\neg\forall x\, P$ e cada coleção de fórmulas contendo $\neg\forall x\, P$ a um tablô com somente um nó não marcado cuja fórmula é $\exists x \neg P$.

$$\begin{array}{c} \neg\forall x\, P\surd \\ \vdots \\ \exists x\neg P \end{array}$$

Figura 10.6: Negação de Fórmula Universal

- **Negação de Fórmula Existencial** $(\neg\exists x\, P)$:
  Esta regra associa cada fórmula $\neg\exists x\, P$ e cada coleção de fórmulas contendo $\neg\exists x\, P$ a um tablô com somente um nó não marcado cuja fórmula é $\forall x \neg P$.

$$\begin{array}{c} \neg\exists x\, P\surd \\ \vdots \\ \forall x\neg P \end{array}$$

Figura 10.7: Negação de Fórmula Existencial

4.11 Exemplo. (Negação de Fórmula Universal)

$$\text{Nem todo leão é feroz.}$$
$$\downarrow$$
$$\neg\forall x(\text{leão}(x) \rightarrow \text{ feroz}(x))$$
$$\leftrightarrow$$
$$\exists x\neg(\text{leão}(x) \rightarrow \text{ feroz}(x))$$
$$\leftrightarrow$$
$$\exists x(\text{leão}(x)\neg \text{ feroz}(x))$$
$$\downarrow$$
$$\text{Existe um leão que não é feroz.}$$

4.12 Exemplo. (Negação de Fórmula Existencial)

$$\text{Não existe um leão que seja vegetariano.}$$
$$\downarrow$$
$$\neg\exists x(\text{leão}(x) \wedge \text{ vegetariano}(x))$$
$$\leftrightarrow$$
$$\forall x\neg(\text{leão}(x) \wedge \text{ vegetariano}(x))$$
$$\leftrightarrow$$
$$\forall x(\text{leão}(x) \rightarrow \neg \text{ vegetariano}(x))$$
$$\downarrow$$
$$\text{Todo leão não é vegetariano.}$$

Capítulo 11

Lógica Equacional Clássica

Este capítulo apresenta a *Lógica Equacional Clássica* ou simplesmente LEC, também conhecida como *Lógica Elementar Clássica, Lógica de Primeira Ordem com Igualdade* ou *Cálculo de Predicados de Primeira Ordem com Igualdade*. Esta lógica estuda todas as propriedades clássicas dos conectivos, dos quantificadores, da igualdade e suas possíveis interações.

§1. Uma Linguagem para LEC

1.1 Definição.
Um *alfabeto para* LEC contém todos os sinais de um alfabeto para LQC, definido em 5.1.7, mais um sinal predicativo diádico, o '$=$'.

1.2 Definição.
Os termos e fórmulas em LEC são todos os termos e fórmulas obtidos pelas regras de formação de LQC, mais as fórmulas da forma =(t_1, t_2), onde t_1 e t_2 são termos em LEC.

Valem aqui todas as convenções anteriores, com as devidas adaptações, sempre que necessário. Seguindo a escrita informal de fórmulas em LQC, temos que a fórmula =(t_1, t_2) pode ser notada por $(t_1 = t_2)$, e podemos prescindir do seu par exterior de parênteses quando esta não for subfórmula de outra fórmula ou for um dos componentes de uma fórmula formada por um conectivo diádico.

1.3 Leitura. A fórmula $t_1 = t_2$ pode ser lida de uma das seguintes formas:
- 't_1 *é igual a* t_2',
- 't_1 *é idêntico a* t_2'.

§2. Uma Semântica para LEC

2.1 Definição. Os *valores veritativos em* LEC são v e f, onde o único *valor distinguido* é v. Uma ordem é considerada na coleção de valores veritativos em LEC, em que f $<$ v.

2.2 Definição.
$I = \langle \Delta, w, s\rangle$ é uma LEC-interpretação se I é uma LQC-interpretação satisfazendo as seguintes condições adicionais[1]:
(i) $= \in \mathcal{D}(w)$,
(ii) $w(=) = \{\langle d, d\rangle | d \in \Delta\}$.

2.3 Definição. Sejam I uma LEC-interpretação, $x_1, \ldots, x_n$ variáveis distintas e $d, d_1, \ldots, d_n$ elementos de Δ. Definimos:
- $I(x|d)$ é uma LEC-interpretação definida por $I(x|d) = \langle \Delta, w, s(x|d)\rangle$.
- $I(x_1, \ldots, x_n|d_1, \ldots, d_n)$ é uma LEC-interpretação definida por $I(x_1, \ldots, x_n|d_1, \ldots, d_n) = \langle \Delta, w, s(x_1, \ldots, x_n|d_1, \ldots, d_n)\rangle$.

2.4 Definição.
Dada uma LEC-interpretação $I = \langle \Delta, w, s\rangle$, as funções I_D e I_V, as quais são respectivamente a LEC-*denotação definida por* I e a LEC-*valoração definida por* I, também podem ser definidas através das cláusulas especificadas em 5.2.16.

2.5 Escólio. $I_V(t = u) =$ v *sss* $I_D(t) = I_D(u)$.

2.6 Proposição.
$I_V(\overline{\exists} xP) =$ v sss existe um máximo $d \in \Delta$ tal que $I(x|d)_V(P) =$ v.

Seja $I = \langle \Delta, w, s\rangle$.
Suponha que $I_V(\overline{\exists} xP) =$ v.
Sendo x_1 e x_2 as primeiras duas variáveis não livres em P, temos que $\overline{\exists} xP$ é $\forall x_1 \forall x_2 (P(x|x_1) \wedge P(x|x_2) \to x_1 = x_2)$.

Prova da ida:
Daí $I_V(\forall x_1 \forall x_2 (P(x|x_1) \wedge P(x|x_2) \to x_1 = x_2)) =$ v. (1)
Seja $d_1, d_2 \in \Delta$ tal que $\begin{cases} I(x|d_1)_V(P) = \text{v}, \\ I(x|d_2)_V(P) = \text{v}. \end{cases}$
Seja $I' = \langle \Delta, w, s'\rangle$, onde $s'(y) = \begin{cases} s(y), \text{ se } y \neq x_1 \text{ e } y \neq x_2, \\ d_1, \text{ se } y = x_1, \\ d_2, \text{ se } y = x_2. \end{cases}$
De (1) e LC, temos que $I'_V(\forall x_1 \forall x_2 (P(x|x_1) \wedge P(x|x_2) \to x_1 = x_2)) =$ v.

[1]Essas condições podem ser resumidas assim: $w(=) = I_\Delta$, de acordo com a definição 2.3.83.

Daí $I'_V(P(x|x_1) \wedge P(x|x_2) \to x_1 = x_2) = \text{v}$.
$I'_V(P(x|x_1)) = I'(x|d_1)_V(P)$, que pelo lema 5.2.23 é igual a $I(x|d_1)_V(P) = \text{v}$.
Analogamente, temos que $I'_V(P(x|x_2)) = \text{v}$.
$I'_V(P(x|x_1) \wedge P(x|x_2)) = \text{v}$, donde $I'_V(x_1 = x_2) = \text{v}$, logo $I'_D(x_1) = I'_D(x_2)$, portanto $d_1 = d_2$. □

Prova da volta:
Suponha que $d_1, d_2 \in \Delta$ tal que $I(x_1, x_2|d_1, d_2)_V(P(x|x_1) \wedge P(x|x_2)) = \text{v}$.
Daí $\begin{cases} I(x_1, x_2|d_1, d_2)_V(P(x|x_1)) = \text{v}, \\ I(x, x_2|d_1, d_2)_V(P(x|x_2)) = \text{v}, \end{cases}$
$\left.\begin{cases} I(x_1|d_1)_V(P(x|x_1)) = \text{v}, \\ I(x_2|d_2)_V(P(x|x_2)) = \text{v}, \end{cases}\right\} (1)$.
De (1) e por 5.2.18, $\begin{cases} I(x|d_1)_V(P) = \text{v}, \\ I(x|d_2)_V(P) = \text{v}, \end{cases}$
logo, $d_1 = d_2$, donde $I(x_1, x_2|d_1, d_2)_D(x_1) = I(x_1, x_2|d_1, d_2)_D(x_2)$, logo $I(x_1, x_2|d_1, d_2)_V(x_1 = x_2) = \text{v}$.
Logo $I(x_1, x_2|d_1, d_2)_V(P(x|x_1) \wedge P(x|x_2) \to x_1 = x_2) = \text{v}$.
Ou seja, para qualquer $d_1, d_2 \in \Delta$,
$I(x_1, x_2|d_1, d_2)_V(P(x|x_1) \wedge P(x|x_2) \to x_1 = x_2) = \text{v}$.
Portanto $I_V(\forall x_1 \forall x_2(P(x|x_1) \wedge P(x|x_2)) \to x_1 = x_2) = \text{v}$.
Portanto $I_V(\overline{\exists} x P) = \text{v}$. □

2.7 Proposição.
$I_V(\exists! x P) = \text{v}$ sss existe um único $d \in \Delta$ tal que $I(x|d)_V(P) = \text{v}$.

Prova:
É imediata, através da proposição 2.6 e da definição de semântica geral para $\exists x P$[2]. □

2.8 Definição. (Satisfabilidade de fórmula e de coleção de fórmulas em LEC)
Sejam $\begin{cases} I \text{ uma LEC-interpretação,} \\ P \text{ uma } I\text{-fórmula,} \\ \Gamma \text{ uma } I\text{-coleção de fórmulas.} \end{cases}$

- *I satisfaz uma fórmula P em* LEC, ou P é LEC-*satisfatível*, se $I_V(P)$ é um valor distinguido em LEC, ou seja, $I_V(P) = \text{v}$,
- *I satisfaz uma coleção Γ de fórmulas em* LEC, ou Γ é LEC-*satisfatível*, se I satisfaz cada fórmula de Γ em LEC,
- Caso contrário, P é LEC-insatisfatível e Γ é LEC-insatisfatível.

[2] $I_V(\exists x P) = \text{v}$ sss existe $d \in \Delta$ tal que $I(x|d)_V(P) = \text{v}$.

2.9 Definição. (**Validade de fórmula e de coleção de fórmulas em** LEC)

(i) *I valida Γ em* LEC se cada fórmula de Γ é uma I-fórmula e se existe uma fórmula de Γ tal que I satisfaz essa fórmula.

(ii) P é LEC-*válido* se P é uma fórmula em LEC e toda LEC-interpretação para P satisfaz P, caso contrário, P é dito ser LEC-inválido.

(iii) Γ é LEC-*válido* se toda LEC-interpretação para Γ valida Γ, caso contrário Γ é dito ser LEC-inválido.

2.10 Definição. (LEC-**Consequência Semântica**)
P é consequência de Γ em LEC, $\Gamma \models_{LEC} P$, se toda LEC-interpretação para $\Gamma \cup \{P\}$ que satisfaz Γ, também satisfaz P.

2.11 Leituras. $\Gamma \models_{LEC} P$
- 'Γ acarreta semanticamente P em LEC',
- 'P é consequência semântica de Γ em LEC',
- 'De Γ afirma-se semanticamente P em LEC'.

§3. Um Cálculo de Sequentes para LEC

Esta seção apresenta um cálculo de sequentes para LEC.

Este cálculo é constituído de todas as leis primitivas de LQC, mais os *Esquemas Primitivos da Igualdade* definidos a seguir:
- Leis primitivas de LQC
- Esquemas primitivos da Igualdade $\begin{cases} * \text{ Reflexividade da Igualdade,} \\ * \text{ Esquema da Substituição em termos funcionais,} \\ * \text{ Esquema da Substituição em fórmulas atômicas.} \end{cases}$

As demais leis podem ser obtidas dos esquemas primitivos de LEC.

Todas as leis válidas em LPC e LQC, devidamente traduzidas para LEC, são ainda válidas em LEC, com exceção do Esquema da Substituição da Equivalência, apresentado na página 193.

Nesta seção nós falaremos somente da Lógica Equacional Clássica; assim, para dizer que P é consequência de Γ em LEC, notaremos isto por $\Gamma \vdash P$.

Esquemas Primitivos da Igualdade

3.1. **Reflexividade da Igualdade** $\vdash t = t$.

3.2. **Substituição em Termos Funcionais**
$t_1 = u_1, \ldots, t_n = u_n \vdash f(t_1, \ldots, t_n) = f(u_1, \ldots, u_n)$.

3.3. **Substituição em Fórmulas Atômicas**
$t_1 = u_1, \ldots, t_n = u_n \vdash p(t_1, \ldots, t_n) \to p(u_1, \ldots, u_n)$.

Prova:

1	$t_1 = u_1, \ldots, t_n = u_n$	pr
2	$p(t_1, \ldots, t_n) \to p(u_1, \ldots, u_n)$	1, SFA
3	$u_1 = t_1, \ldots, u_n = t_n$	1, SI
4	$p(u_1, \ldots, u_n) \to p(t_1, \ldots, t_n)$	3, SFA
5	$p(t_1, \ldots, t_n) \leftrightarrow p(u_1, \ldots, u_n)$	2, 4, $\wedge$-int

□

Leis Básicas da Igualdade

3.4. **Simetria da Igualdade** $t = u \vdash u = t$.

Prova:

1	$t = u$	pr
2	$t = t$	RI
3	$t = t \to u = t$	1, 2, SFA
4	$u = t$	2, 3, MP

□

A conclusão de um exemplar do esquema da substituição em fórmulas atômicas pode ser uma equivalência.

3.5. **Substituição em Fórmulas Atômicas**
$t_1 = u_1, \ldots, t_n = u_n \vdash p(t_1, \ldots, t_n) \leftrightarrow p(u_1, \ldots, u_n)$.

Prova:

1	$t_1 = u_1, \ldots, t_n = u_n$	pr
2	$u_1 = t_1, \ldots, u_n = t_n$	1,SI
3	$p(t_1, \ldots, t_n) \to p(u_1, \ldots, u_n)$	1,SFA
4	$p(u_1, \ldots, u_n) \to p(t_1, \ldots, t_n)$	2,SFA
5	$p(t_1, \ldots, t_n) \leftrightarrow p(u_1, \ldots, u_n)$	4

□

3.6. **Transitividade da Igualdade** $t = u, u = v \vdash t = v$.

Prova:

1	$t = u$	pr
2	$u = v$	pr
3	$t = t$	RI
4	$t = u \rightarrow t = v$	3,2,SFA
5	$t = v$	1,4,MP

$\square$

3.7. **Simetria e Transitividade da Igualdade**

$$\begin{cases} \text{(i)} \;\; t = v, u = v \vdash t = u, \\ \text{(ii)} \;\; v = t, v = u \vdash t = u. \end{cases}$$

Os esquemas de substituição da igualdade em sinais funcionais e em sinais predicativos admitem suas correspondentes generalizações, conforme é descrito a seguir:

3.8. **Esquema da Substituição da Igualdade para Termos**
$t_1 = t_2 \vdash u(v \| t_1) = u(v \| t_2)$.[3]

3.9. **Esquema da Substituição da Igualdade para Fórmulas**

- Se $x_1, \ldots, x_n$ são as variáveis livres em $\{t_1, t_2\}$ tais que v está em Q no seu escopo, então
  $\forall x_1 \ldots \forall x_n (t_1 = t_2) \vdash Q(v \| t_1) \leftrightarrow Q(v \| t_2)$.

[3]Em linguagens nas quais termos não possuem variáveis ligadas, como é o caso de LQC e LEC, temos que $u(v \| t)$ é idêntico a $u(v | t)$, daí esta lei também poderia ser denominada ‘Esquema da Instanciação da Igualdade para Termos’.

3.10. **Regra da Substituição da Igualdade para Fórmulas**

- Se $\begin{cases} * \ \ \Gamma \vdash t_1 = t_2, \\ * \ \ v \text{ não está em } Q\text{, no escopo de nenhuma variável livre em } \Gamma \text{ e em} \\ \quad \{t_1, t_2\}, \end{cases}$

  então $\Gamma \vdash Q(v \| t_1) \leftrightarrow Q(v \| t_2)$.

3.11. **Exemplo de Falácia**

Temos que $TNI \vdash \forall x(x + 7 > x)$.
Daí $TNI, x = 14 \vdash \forall x(x + 7 > x)$. (1)
Temos também que $TNI, x = 14 \vdash x = 14$. (2)
Seja Q a fórmula '$\forall x(x + 7 > y)$' .
Então $\begin{cases} Q(y|x) = \forall x(x + 7 > x), \\ Q(y|14) = \forall x(x + 7 > 14). \end{cases}$
Daí, por (2) e RSIF,
$TNI, x = 14 \vdash Q(v\|x) \leftrightarrow Q(v\|14)$,
$TNI, x = 14 \vdash \forall x(x + 7 > x) \leftrightarrow \forall x(x + 7 > 14)$. (3)
De (1) e (3), $TNI, x = 14 \vdash \forall x(x + 7 > 14)$.
$TNI \vdash x = 14 \to \forall x(x + 7 > 14)$.
Daí $TNI \vdash 14 = 14 \to \forall x(x + 7 > 14)$.
$TNI \vdash \forall x(x + 7 > 14)$.
$TNI \vdash 0 + 7 > 14$.
Logo, $TNI \vdash 7 > 14$, o que é absurdo.

O erro encontra-se onde: y está no escopo de x em Q e x é livre em Γ e em $\{x, 14\}$.

A instanciação de variáveis por termos em fórmulas permite uma formulação correspondente ao esquema da substituição da igualdade para fórmulas sem quaisquer restrições.

3.12. **Esquema da Instanciação da Igualdade para Fórmulas**

- $t_1 = t_2 \vdash Q(x|t_1) \leftrightarrow Q(x|t_2)$.

3.13. **Esquema da Instanciação da Igualdade para Termos**

- $t_1 = t_2 \vdash u(v|t_1) = u(v|t_2)$.

E podem ser reduzidos ao *Esquema da Instanciação da Igualdade*:

3.14. **Esquema da Instanciação da Igualdade**

- $t_1 = t_2 \vdash F(x|t_1) \equiv F(x|t_2)$.

3.15 Corolário. $t_1 = t_2, Q(x|t_1) \vdash Q(x|t_2)$.

A Regra e o Esquema da Substituição da Equivalência, concernentes a LQC, formulados na página 193, valem também para LEC. Por comodidade repetimos novamente aqui os seus enunciados.

3.16 Definição.
Sejam D_1 e D_2 ambos termos ou ambos fórmulas:
$$D_1 \equiv D_2 \rightleftharpoons \begin{cases} D_1 = D_2, \text{ se } D_1 \text{ e } D_2 \text{ são ambos termos.} \\ D_1 \leftrightarrow D_2, \text{ se } D_1 \text{ e } D_2 \text{ são ambos fórmulas.} \end{cases}$$

3.17. **Esquema da Substituição da Equivalência**
Sejam D_1, D_2, E, F designadores tal que todos sao termos ou todos são fórmulas.
Se $x_1, \ldots, x_n$ são as variáveis livres em $\{D_1, D_2\}$ tais que E está em F no seu escopo, então
$\forall x_1 \ldots \forall x_n(D_1 \equiv D_2) \vdash F(E\|D_1) \equiv F(E\|D_2)$.

3.18. **Regra da Substituição da Equivalência**

- Se $\begin{cases} * \ \ \Gamma \vdash P_1 \leftrightarrow P_2, \\ * \ \ S \text{ não está em } Q \text{, no escopo de nenhuma variável livre em } \Gamma \text{ e em} \\ \quad\ \{P_1, P_2\}, \end{cases}$

  então $\Gamma \vdash Q(S\|P_1) \leftrightarrow Q(S\|P_2)$.

3.19. **Regra da Substituição da Correspondência**
Se $\begin{cases} \Gamma \vdash D_1 \equiv D_2, \\ E \text{ não está em } F \text{, no escopo de nenhuma variável livre em } \Gamma \text{ e } \{D_1, D_2\}, \end{cases}$
então $\Gamma \vdash F(E\|D_1) \equiv F(E\|D_1) \equiv F(E\|D_2)$.

3.20 Corolário.
Se $\begin{cases} \Gamma \vdash D_1 \equiv D_2, \\ \Gamma \vdash Q(E\|D_1), \\ E \textit{ não está, em } Q \textit{, no escopo de nenhuma variável livre em } \Gamma, \{D_1, D_2\}, \end{cases}$
então, $\Gamma \vdash Q(E\|D_2)$.

Uma forma bem importante de absorção de quantificadores envolvendo a igualdade é formulada a seguir.

3.21. **Quantificação Pontual**
Se x não é livre em t, então:
(i) $\vdash \forall x(x = t \to P) \leftrightarrow P(x|t)$,
(ii) $\vdash \exists x(x = t \wedge P) \leftrightarrow P(x|t)$.

Prova de (i) ida:

1	x não é livre em t	hip
2	$\forall x(x = t \to P)$	sup
3	$(x = t \to P)(x\|t)$	2, $\forall$-el
4	$t = t \to P(x\|t)$	3, 1
5	$t = t$	RI
6	$P(x\|t)$	5, 4, MP
7	$\forall x(x = t \to P) \to P(x\|t)$	2, 6, RD

□

Prova de (i) volta:

1	x não é livre em t	hip
2	$P(x\|t)$	sup
3	$x = t$	sup
4	$P \leftrightarrow P(x\|t)$	3, II
5	P	2, 4, MPEq
6	$x = t \to P$	3, 5, RD
7	$\forall x(x = t \to P)$	1, 6
8	$P(x\|t) \to \forall x(x = t \to P)$	2, 7, RD

□

Prova de (ii) ida:

1	x não é livre em t	hip
2	$\quad \exists x(x = t \wedge P)$	sup
3	$\quad\quad x = t \wedge P$	sup
4	$\quad\quad x = t$	3, $\wedge$-el
5	$\quad\quad P$	3, $\wedge$-el
6	$\quad\quad P \leftrightarrow P(x\vert t)$	4, II
7	$\quad\quad P(x\vert t)$	5, 6, MPEq
8	$\quad P(x\vert t)$	1, 2, 3, 7, $\exists$-el
9	$\exists x(x = t \wedge P) \to P(x\vert t)$	1, 2, 8, RD

□

Prova de (ii) volta:

1	x não é livre em t	hip
2	$\quad P(x\vert t)$	sup
3	$\quad t = t$	RI
4	$\quad t = t \wedge P(x\vert t)$	2, 3, $\wedge$-int
5	$\quad (x = t \wedge P)(x\vert t)$	4, 1
6	$\quad \exists x(x = t \wedge P)$	5, $\exists$-int
7	$P(x\vert t) \to \exists x(x = t \wedge P)$	2, 6, RD

□

Podemos estender a quantificação pontual para mais de um objeto, como é mostrado a seguir.

3.22. **Quantificação Pontual Plural**[4]
Se x não é livre em $\{t_1, t_2\}$, então:
(i) $\vdash \forall(x = t_1 \vee x = t_2)xP \leftrightarrow P(x|t_1) \wedge P(x|t_2)$,
(ii) $\vdash \exists(x = t_1 \vee x = t_2)xP \leftrightarrow P(x|t_1) \vee P(x|t_2)$.

3.23 Definição. $t_1 \neq t_2 \rightleftharpoons \neg(t_1 = t_2)$.

[4] Ou *Quantificação Pontual Estendida.*

3.24. **Unificação pela Substituição**.

- Se v não está, em Q, no escopo de nenhuma variável livre em $\{t_1, t_2\}$, então
  $Q(v\|t_1), \neg Q(v\|t_2) \vdash t_1 \neq t_2$.

Prova: Se, para cada $i \in \{1, \dots, n\}$, v_i não está, em Q, no escopo de nenhuma variável livre em $\{t_i, u_i\}$, então
$Q(\overrightarrow{v}\|\overrightarrow{t}), \neg Q(\overrightarrow{v}\|\overrightarrow{u}) \vdash t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n$. □

O esquema abaixo costuma ser utilizado no método dos tablôs, como forma de automatizar a igualdade.

3.25. **Unificação pela Instanciação** $Q(x|t_1), \neg Q(x|t_2) \vdash t_1 \neq t_2$.

Uma fórmula instanciada quantificada pode ser equivalente, sob certas condições, à mesma fórmula não instanciada quantificada, conforme expressa a lei seguinte.

3.26. **Reversão da Instanciação**

(i) Se y não é livre em x, t, P, então $\forall y \exists x(y = t) \vdash \Psi x P(x|t) \leftrightarrow \Psi x P$.

(ii) Se $\begin{cases} y \text{ não é livre em } x, t, P, \\ x \text{ não é livre em } R, \end{cases}$

então $\forall R y \exists x(y = t) \vdash \Psi R x P(x|t) \leftrightarrow \Psi R x P$.

(iii) não é livre em x, t, R, P, então $\forall y \exists x(y = t) \vdash \Psi x P(x|t) \leftrightarrow \Psi x P$.

Leis Básicas dos Quantificadores Numéricos

Na Lógica Equacional, para cada número inteiro n, positivo ou nulo, é possível definir três quantificadores existenciais especiais, ditos também *quantificadores numéricos*, usados para a representação formal de expressões dos tipos ‘existem pelo menos n objetos x tais que $P(x)$’, ‘existem no máximo n objetos x tais que $P(x)$’ e ‘existem exatamente n objetos x tais que $P(x)$’.

A fórmula $\exists x P$, já por nós conhecida, corresponde à expressão ‘existe pelo menos um objeto x tal que $P(x)$’. O caso em que $n = 1$ é o mais importante para aplicações imediatas. Definimos abaixo a representação formal de expressões do tipo ‘existe no máximo um objeto x tal que $P(x)$’ e ‘existe um único objeto x tal que $P(x)$’.

3.27 Definição. $\overline{\exists}$ **(existe um máximo)**

- $\overline{\exists} x P \rightleftharpoons \forall x_1 \forall x_2 (P(x|x_1) \wedge P(x|x_2) \rightarrow x_1 = x_2)$, onde x_1, x_2 são as primeiras variáveis não livres em P.

3.28 Definição. $\exists!$ (**existe um único**)

- $\exists!xP \rightleftharpoons \exists xP \wedge \overline{\exists}xP$.

3.29 Exemplo.
Considere o domínio dos números inteiros $\mathbb{Z}$.
$\overline{\exists}x(\text{primo}(x) \wedge \text{par}(x))$.
$\exists!x(\text{primo}(x) \wedge \text{par}(x))$.

A definição de $\overline{\exists}xP$ foi concebida levando em conta a sua uniformidade com as demais definições dos quantificadores numéricos lidando com diferentes pluralidades de objetos, porém o seu caso particular admite uma pequena simplificação. Tanto esta simplificação como a irrelevância na escolha das variáveis x_1 e x_2[5] são formulados a seguir.

Considere as seguintes formas equivalentes para $\overline{\exists}$.

3.30. **Maximidade**

Sejam $\begin{cases} y_1 \text{ e } y_2 \text{ variáveis distintas não livres em } \exists xP^a, \\ y \text{ não é livre em } x, P. \end{cases}$

[a] Um exemplo de tais duas variáveis está em x, y, onde y não é livre em x, P.

(i) $\overline{\exists}xP$,
(ii) $\forall y_1 \forall y_2 (P(x|y_1) \wedge P(x|y_2) \rightarrow y_1 = y_2)$,
(iii) $\exists x \forall y (P(x|y) \rightarrow x = y)$.

3.31 Corolário. ***(Lei da Maximidade)*** $\overline{\exists}xP, P(x|t_1), P(x|t_2) \vdash t_1 = t_2$.

Considere as seguintes formas equivalentes para $\exists!xP$.

3.32. **Unicidade**
Se y não é livre em $\{x, P\}$, então as seguintes fórmulas são equivalentes em LEC:

(i) $\exists!xP$,
(ii) $\exists x(P \wedge \overline{\exists}xP)$,
(iii) $\exists x(P \wedge \forall y(P(x|y) \rightarrow x = y))$,
(iv) $\exists x \forall y(P(x|y) \leftrightarrow x = y)$.

[5] A importância na determinação das variáveis x_1 e x_2 é, no entanto, fundamental para definir-se uma fórmula única em cada caso, e não uma classe de fórmulas.

Prova de (i) para (ii).

1	y não é livre em x, P	hip
2	$\exists! x P$	sup
3	$\exists x P \wedge \overline{\exists} x P$	2, def
4	$\exists x P$	3, $\wedge$-el
5	$\overline{\exists} x P$	3, $\wedge$-el
6	P	sup
7	$P \wedge \overline{\exists} x P$	6, 5, $\wedge$-int
8	$\exists x (P \wedge \overline{\exists} x P)$	7, $\exists$-int
9	$\exists x (P \wedge \overline{\exists} x P)$	4, 6, 9, $\exists$-el
10	$\exists! x P \rightarrow \exists x (P \wedge \overline{\exists} x P)$	2, 9, RD

□

Prova de (ii) para (iii).

1	y não é livre em x, P	hip
2	$\exists x (P \wedge \overline{\exists} x P)$	sup
3	$P \wedge \overline{\exists} x P$	sup
4	$P \wedge \forall y (P(x\|y) \rightarrow y = x)$	3, AUn, 1
5	$P \wedge \forall y (P(x\|y) \rightarrow x = y)$	4, SI
6	$\exists x (P \wedge \forall y (P(x\|y) \rightarrow x = y))$	5, $\exists$-int
7	$\exists x (P \wedge \forall y (P(x\|y) \rightarrow x = y)))$	2, 3, 6, $\exists$-el
8	$\exists x (P \wedge \overline{\exists} x P) \rightarrow \exists x (P \wedge \forall y (P(x\|y) \rightarrow x = y))$	2, 7, RD

□

Prova de (iii) para (iv).

1	y não é livre em x, P	hip
2	$\exists x(P \wedge \forall y(P(x\|y) \to x = y))$	sup
3	$\quad P \wedge \forall y(P(x\|y) \to x = y)$	sup
4	$\quad P \wedge \forall y(P(x\|y) \to y = x)$	3, SI
5	$\quad \forall y(P(x\|y) \leftrightarrow y = x)$	4, AUn
6	$\quad \forall y(P(x\|y) \leftrightarrow x = y)$	5, SI
7	$\quad \exists x \forall y(P(x\|y) \leftrightarrow x = y)$	6, $\exists$-int
8	$\exists x \forall y(P(x\|y) \leftrightarrow x = y)$	2, 3, 7, $\exists$-el
9	$\exists x(P \wedge \forall y(P(x\|y) \to x = y)) \to \exists x \forall y(P(x\|y) \leftrightarrow x = y)$	2, 8, RD

□

Prova de (iv) para (i).

1	y não é livre em x, P	hip
2	$\exists x \forall y(P(x\|y) \leftrightarrow x = y)$	sup
3	$\quad \forall y(P(x\|y) \leftrightarrow x = y)$	sup
4	$\quad \forall y(P(x\|y) \leftrightarrow y = x)$	3, SI
5	$\quad P \wedge \exists! x P$	4, AUn,1
6	$\quad \exists! x P$	5, $\wedge$-el
7	$\exists! x P$	2, 3, 6, $\exists$-el
8	$\exists x \forall y(P(x\|y) \leftrightarrow x = y) \to \exists! x P$	2, 7, 8, RD

□

3.33. **Absorção na Unicidade**[6]
Se y não é livre em $\{t, \exists x P\}$, então as seguintes fórmulas são equivalentes em LEC:

(i) $P(x|t) \wedge \exists! x P$,
(ii) $P(x|t) \wedge \overline{\exists} x P$,
(iii) $P(x|t) \wedge \forall y (P(x|y) \rightarrow y = t)$,
(iv) $\forall y (P(x|y) \leftrightarrow y = t)$.

3.34. **Vinculação** $\forall x (P \rightarrow Q), \overline{\exists} x Q \vdash \exists x P \wedge Q \leftrightarrow P \wedge Q$.

Prova:

1	$\forall x (P \rightarrow Q)$	pr
2	$\overline{\exists} x Q$	pr
3	y is not free in P, Q	dsg
4	$\exists x P \wedge Q$	sup
5	$\exists x P$	4, $\wedge$-el
6	Q	4, $\wedge$-el
7	$P(x|y)$	sup
8	$P(x|y) \rightarrow Q(x|y)$	2, $\forall$-el
9	$Q(x|y)$	7, 8, MP
10	$x = y$	3, 6, 9, CMx
11	P	10, 7, II
12	P	5, 7, 11, $\exists$-el, 3
13	$P \wedge Q$	12, 6, $\wedge$-int
14	$\exists x P \wedge Q \rightarrow P \wedge Q$	4, 13, RD

□

Também é possível, para cada número inteiro n, positivo ou nulo, definir três quantificadores universais especiais, ditos também *quantificadores numéricos*, usados para a representação formal de expressões dos tipos 'existem pelo menos n exceções a $P(n)$', 'existem no máximo n exceções a $P(n)$' e 'existem exatamente n exceções a $P(n)$'.

[6]Vale para a Unicidade e Maximidade. Como partimos da Unicidade, por convenção iremos adotar esta nomenclatura.

A fórmula $\forall xP$, já por nós conhecida, corresponde à expressão 'existem pelo menos n restrições a $P(n)$'. O caso em que $n = 0$ é o mais importante para aplicações imediatas.

3.35 Definição.

- $\exists(\leq n)xP \rightleftharpoons \forall y_1 \forall y_2 \ldots \forall y_{n+1}(\bigwedge(P(x|y_1), \ldots, P(x|y_{n+1})) \to = (y_1, \ldots, y_{n+1}))$.
- $\exists(\geq n)xP \rightleftharpoons \exists y_1 \ldots \exists y_n(\neq (y_1, \ldots, y_n) \wedge \bigwedge(P(x|y_1), \ldots, P(x|y_n)))$.
- $\exists(= n)xP \rightleftharpoons \exists(\leq n)xP \wedge \exists(\geq n)xP$.

- $\forall(\leq n)xP \rightleftharpoons \forall y_1 \ldots \forall y_{n+1}(\bigwedge(\neg P(x|y_1), \ldots, \neg P(x|y_{n+1})) \to = (y_1, \ldots, y_{n+1}))$.
- $\forall(\geq n)xP \rightleftharpoons \exists y_1 \ldots \exists y_n(\neq (y_1, \ldots, y_n) \wedge \bigwedge(\neg P(x|y_1), \ldots, \neg P(x|y_n)))$.
- $\forall(= n)xP \rightleftharpoons \forall(\leq n)xP \wedge \forall(\geq n)xP$.

3.36 Exemplo.

- $\forall(\geq 1)x(\text{primo}(x) \to \text{ímpar}(x))$.
- $\forall(= 1)x(\text{primo}(x) \to \text{ímpar}(x))$.
- $\forall(\geq 2)x(x \in \mathbb{N} \wedge x < 10 \to x < 6)$.
- $\forall(= 4)x(x \in \mathbb{N} \wedge x < 10 \to x < 6)$.

Concluindo temos que:

3.37 Escólio.

- $\vdash \forall xP \leftrightarrow \forall(\leq 0)xP \leftrightarrow \forall(= 0)P$.
- $\vdash \exists xP \leftrightarrow \exists(\geq 1)xP$.

3.38. **Distributividade e Fatorabilidade Disparadas**[7] **de Quantificadores**

(i) $\overline{\exists}x(\neg P \vee \neg Q) \vdash \forall x(P \to Q) \leftrightarrow (\forall xP \to \forall xQ)$,

(ii) $\overline{\exists}x(P \vee Q) \vee \overline{\exists}x(\neg P \vee \neg Q) \vdash \forall x(P \vee Q) \leftrightarrow \forall xP \vee \forall xQ$,

(iii) $\overline{\exists}x(\neg P \vee \neg Q) \vdash \forall x(P \leftrightarrow Q) \leftrightarrow (\forall xP \leftrightarrow \forall xQ)$,

(iv) $\overline{\exists}x\neg Q \vdash \exists x(P \to Q) \leftrightarrow (\exists xP \to \exists xQ)$,

(v) $\overline{\exists}x(P \vee Q) \vee \overline{\exists}x(\neg P \vee \neg Q) \vdash \exists x(P \wedge Q) \leftrightarrow \exists xP \wedge \exists xQ$,

(vi) $\overline{\exists}x\neg P \wedge \overline{\exists}x\neg Q \vdash \exists x(P \leftrightarrow Q) \to (\exists xP \leftrightarrow \exists xQ)$,

(vii) $\overline{\exists}x(P \vee Q) \vee \overline{\exists}x(\neg P \vee \neg Q) \vdash (\exists xP \leftrightarrow \exists xQ) \to \exists x(P \leftrightarrow Q)$,

(viii) $\overline{\exists}x(\neg P \vee \neg Q) \vdash \exists x(P \leftrightarrow Q) \leftrightarrow (\exists xP \leftrightarrow \exists xQ)$.

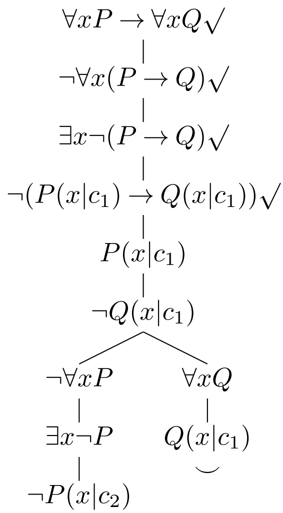
A partir dos resultados de *Distributividade e Fatorabilidade de Quantificadores* apresentados em 5.3.30 é possível chegar aos resultados de *Distributividade e Fatorabilidade Disparadas de Quantificadores*.

[7] Dispara uma consequência a partir de certas ocorrências.

A fim de exemplificar como chegamos a estes resultados, tomemos o item (i) da *Distributividade e Fatorabilidade de Quantificadores* e o item (i) da *Distributividade e Fatorabilidade Disparadas de Quantificadores*:

DFQ(i) : $\vdash \forall x(P \to Q) \to (\forall x P \to \forall x Q)$.
DFDQ(i): $\overline{\exists} x(\neg P \vee \neg Q) \vdash \forall x(P \to Q) \leftrightarrow (\forall x P \to \forall x Q)$.

Transformamos DFQ (i) em DFDQ (i) utilizando o método dos tablôs por refutação, quando
$\Gamma \vdash (\forall x P \to \forall x Q) \to \forall x(P \to Q)$:



Para $P(x|c_1)$ fechar com $\neg P(x|c_2)$, c_1 e c_2 teriam que ser iguais. Consideramos então que, se vale $\neg Q(x|c_1)$ então vale $\neg Q(x|c_1) \vee \neg P(x|c_1)$ e se vale $\neg Q(x|c_2)$ então vale $\neg Q(x|c_2) \vee \neg P(x|c_2)$. Então existe a premissa $\overline{\exists} x(\neg P \vee \neg Q)$. Sendo assim, de DFQ(i) chegamos a DFDQ(i).

Transformaremos agora DFQ (vi) em DFDQ (v), quando $\Gamma \vdash \exists x P \wedge \exists x Q \to \exists x(P \wedge Q)$:

$$
\begin{array}{c}
\exists xP \wedge \exists xQ \surd \\
| \\
\neg\exists x(P \wedge Q) \surd \\
| \\
\exists xP \surd \\
| \\
\exists xQ \surd \\
| \\
\forall x\neg(P \wedge Q) \surd \\
| \\
P(x|c_1) \\
| \\
Q(x|c_2) \\
| \\
\neg(P(x|c_1) \wedge Q(x|c_1)) \surd \\
| \\
\neg(P(x|c_2) \wedge Q(x|c_2)) \surd
\end{array}
$$

$$
\begin{array}{cc}
\neg(P(x|c_1)) & \neg Q(x|c_1) \\
 & \begin{array}{cc} \neg P(x|c_2) & \neg Q(x|c_2) \end{array}
\end{array}
$$

Para $\neg P(x|c_2)$ fechar com $P(x|c_1)$, c_1 e c_2 teriam que ser iguais. Consideramos então que, se vale $Q(x|c_2)$ então vale $Q(x|c_2) \vee \neg P(x|c_2)$ e se vale $Q(x|c_1)$ então vale $Q(x|c_1) \vee \neg P(x|c_1)$.
Então existe a premissa $\overline{\exists}x(P \vee Q) \vee \overline{\exists}x(\neg P \vee \neg Q)$. Sendo assim, de DFQ(vi) chegamos a DFDQ(v).

3.39. **Distributividade e Fatorabilidade Degeneradas Disparadas de Quantificadores**

(i) $\overline{\exists}x(P \vee \neg Q) \vdash \forall x(P \to Q) \leftrightarrow (\exists xP \to \forall xQ)$,

(ii) $\overline{\exists}x(P \vee Q) \vdash \forall x(P \to Q) \leftrightarrow (\exists xP \to \exists xQ)$,

(iii) $\overline{\exists}x(\neg P \vee Q) \vdash \forall x(P \vee Q) \leftrightarrow (\forall xP \vee \exists xQ)$,

(iv) $\overline{\exists}x(P \vee \neg Q) \vdash \forall x(P \vee Q) \leftrightarrow \exists xP \vee \forall xQ$,

(v) $\overline{\exists}x(P \vee \neg Q) \vee \overline{\exists}(\neg P \vee Q) \vdash \forall x(P \leftrightarrow Q) \to (\forall xP \leftrightarrow \exists xQ)$,

(vi) $\overline{\exists}xP \wedge \overline{\exists}x\neg Q \vdash (\forall xP \leftrightarrow \exists xQ) \to \forall x(P \leftrightarrow Q)$,

(vii) $\overline{\exists}x(P \vee \neg Q) \vdash \forall x(P \leftrightarrow Q) \leftrightarrow (\forall xP \leftrightarrow \exists xQ)$,

(viii) $\overline{\exists}x(P \vee \neg Q) \vee \overline{\exists}x(\neg P \vee Q) \vdash \forall x(P \leftrightarrow Q) \to (\exists xP \leftrightarrow \forall xQ)$,

(ix) $\overline{\exists}x\neg P \wedge \overline{\exists}xQ \vdash (\exists xP \leftrightarrow \forall xQ) \to \forall x(P \leftrightarrow Q)$,

$$
\begin{array}{rl}
\text{(x)} & \overline{\exists}x(\neg P \vee Q) \vdash \forall x(P \leftrightarrow Q) \leftrightarrow (\exists xP \leftrightarrow \forall xQ),\\
\text{(xi)} & \overline{\exists}x(P \vee Q) \vdash \forall x(P \leftrightarrow Q) \leftrightarrow (\exists xP \leftrightarrow \exists xQ),\\
\text{(xii)} & \overline{\exists}xP \vdash \exists x(P \to Q) \leftrightarrow \forall xP \to \forall xQ,\\
\text{(xiii)} & \overline{\exists}x(\neg P \vee Q) \vdash \exists x(P \wedge Q) \leftrightarrow \forall xP \wedge \exists xQ,\\
\text{(xiv)} & \overline{\exists}x(P \vee \neg Q) \vdash \exists x(P \wedge Q) \leftrightarrow \exists xP \wedge \forall xQ,\\
\text{(xv)} & \overline{\exists}xP \wedge \overline{\exists}xQ \vdash \exists x(P \leftrightarrow Q) \to (\forall xP \leftrightarrow \forall xQ),\\
\text{(xvi)} & \overline{\exists}x(P \vee Q) \vee \overline{\exists}x(\neg P \vee \neg Q) \vdash (\forall xP \leftrightarrow \forall xQ) \to \exists x(P \leftrightarrow Q),\\
\text{(xvii)} & \overline{\exists}x(P \vee Q) \vdash \exists x(P \leftrightarrow Q) \leftrightarrow (\forall xP \leftrightarrow \forall xQ),\\
\text{(xviii)} & \overline{\exists}x(\neg P \vee Q) \vdash \exists x(P \leftrightarrow Q) \leftrightarrow (\forall xP \leftrightarrow \exists xQ),\\
\text{(xix)} & \overline{\exists}x(P \vee \neg Q) \vdash \exists x(P \leftrightarrow Q) \leftrightarrow (\exists xP \leftrightarrow \forall xQ).
\end{array}
$$

Podemos definir de forma geral todos os resultados apresentados acima, sobre *Distributividade e Fatorabilidade Degenerada de Quantificadores*:

3.40 Definição. $\Psi x(P\#Q) \vdash_{\mathcal{L}} \Upsilon_1 xP\#\Upsilon_2 xQ$, onde $\Upsilon_1 \neq \Psi$ ou $\Upsilon_2 \neq \Psi$. Ou seja, distribui de forma degenerada.

3.41 Definição. $\Upsilon_1 xP\#\Upsilon_2 xQ \vdash_{\mathcal{L}} \Psi x(P\#Q)$, onde $\Upsilon_1 \neq \Psi$ ou $\Upsilon_2 \neq \Psi$. Ou seja, fatora de forma degenerada.

§4. Correção e Completude do cálculo se sequentes de LEC com respeito à semântica de LEC

4.1 Correção do cálculo de sequentes de LEC com respeito à semântica de LEC

4.1 Teorema. ***(Correção)***

- *Se* $\Gamma \vdash_{\mathrm{LEC}} P$, *então* $\Gamma \models_{\mathrm{LEC}} P$.

Podemos agora, demonstrar a correção do cálculo de sequentes de LEC com respeito à sua semântica.

Para as leis estruturais e para as leis de introdução e eliminação de conectivos o raciocínio é análogo à prova do teorema 9.3.1 feita no caso proposicional. Da mesma forma, para as leis de introdução e eliminação de quantificadores o raciocínio é análogo à prova do teorema 10.3.4 feita no caso quantificacional. Então é suficiente provar que as leis concernentes a igualdade são corretas.

Prova:

- Queremos mostrar que o esquema da Reflexividade da Igualdade é correto.
  Seja '$\vdash t = t$' um exemplar de RI.
  Seja I uma LEC-interpretação para t.
  $I_V(t = t) = $ v sss $I_D(t) = I_D(t)$.
  Mas sempre temos que $I_D(t) = I_D(t)$, portanto $I_V(t = t) = $ v.
- Queremos mostrar que o esquema da Substituição em Termos Funcionais é correto.
  Seja '$t_1 = u_1, \ldots, t_n = u_n \vdash f(t_1, \ldots, t_n) = f(u_1, \ldots, u_n)$' um exemplar de STF.
  Seja $I = \langle \Delta, w, s \rangle$ uma LEC-interpretação para $t_1 = u_1, \ldots, t_n = u_n$ que é igual a $f(t_1, \ldots, t_n) = f(u_1, \ldots, u_n)$ que satisfaz $t_1 = u_1, \ldots, t_n = u_n$, ou seja, $\begin{cases} I_V(t_1 = u_1) = \text{v}, \\ \vdots \\ I_V(t_n = u_n) = \text{v}, \end{cases}$ daí
  $$\left.\begin{cases} I_D(t_1) = I_D(u_1), \\ \vdots \\ I_D(t_n) = I_D(u_n). \end{cases}\right\} (1)$$
  Temos que $I_D(f(t_1, \ldots, t_n))$, por definição semântica, é igual a $w(f)(I_D(t_1), \ldots, I_D(t_n))$, que por (1) é igual a $w(f)(I_D(u_1), \ldots, I_D(u_n))$ que é igual a $I_D(f(u_1, \ldots, u_n))$, logo $I_V(f(t_1, \ldots, t_n) = f(u_1, \ldots, u_n)) = $ v.
- Queremos mostrar que o esquema da Substituição em Fórmulas Atômicas é correto.
  Seja '$t_1 = u_1, \ldots, t_n = u_n \vdash p(t_1, \ldots, t_n) = p(u_1, \ldots, u_n)$' um exemplar de SFA.
  Seja $I = \langle \Delta, w, s \rangle$ uma LEC-interpretação para $t_1 = u_1, \ldots, t_n = u_n, p(t_1, \ldots, t_n) = p(u_1, \ldots, u_n)$ que satisfaz $t_1 = u_1, \ldots, t_n = u_n$, ou seja, $\begin{cases} I_V(t_1 = u_1) = \text{v}, \\ \vdots \\ I_V(t_n = u_n) = \text{v}, \end{cases}$ daí
  $$\left.\begin{cases} I_D(t_1) = I_D(u_1), \\ \vdots \\ I_D(t_n) = I_D(u_n). \end{cases}\right\} (1)$$
  Se $I_V(p(t_1, \ldots, t_n)) = $ v, (2)
  então $\langle I_D(t_1), \ldots, I_D(t_n) \rangle \in w(p)$, daí $\langle I_D(u_1), \ldots, I_D(u_n) \rangle \in w(p)$, ou seja, $I_V(p(u_1, \ldots, u_n)) = $ v (3),
  logo de (1), (2) (3) e escólio 4.2.14 item (ii),

$I_V(p(t_1, \ldots, t_n) \rightarrow p(u_1, \ldots, u_n))$.

$\square$

4.2 Completude do cálculo de sequentes de LEC com respeito à semântica de LEC

Considere, adicionalmente, a definição de relação de equivalência apresentada em 2.3.11 e a definição de partição em 2.3.57.

4.2 Definição.
φ é dito uma P-coleção de Henkin em L com respeito a LEC se:

(i) φ é P-saturado em L com respeito a LEC.

(ii) Para qualquer fórmula Q tal que $\exists x Q \in \varphi$, então existe um termo t em L tal que $Q(x|t) \in \varphi$.

(iii) Se, para todo termo t em L, $Q(x|t) \in \varphi$, então $\forall x Q \in \varphi$.

4.3 Definição.
Seja φ uma coleção de fórmulas de L. $\sim_\varphi$ é uma relação entre termos de L definida por $t \sim_\varphi u$ sss $\varphi \vdash_{\overline{\text{LEC}}} t = u$.

4.4 Lema.
Para cada coleção φ de fórmulas de L, temos que $\sim_\varphi$ é relação de equivalência na coleção de termos de L.

4.5 Lema.
Seja $\begin{cases} f \text{ um sinal funcional em } L \text{ de aridade } n, \\ t_1, \ldots, t_n, u_1, \ldots, u_n \text{ termos de } L. \end{cases}$
Então $t_1 \sim_\varphi u_1, \ldots, t_n \sim_\varphi u_n$ *implicam que* $f(t_1, \ldots, t_n) \sim_\varphi f(u_1, \ldots, u_n)$.

Prova:
Assuma as hipóteses.
$\varphi \vdash_{\overline{\text{LEC}}} t_1 = u_1$,
$\vdots$
$\varphi \vdash_{\overline{\text{LEC}}} t_n = u_n$.
Por STF, $t_1 = u_1, \ldots, t_n = u_n \vdash_{\overline{\text{LEC}}} f(t_1, \ldots, t_n) = f(u_1, \ldots, u_n)$. Daí $\varphi \vdash_{\overline{\text{LEC}}} f(t_1, \ldots, t_n) = f(u_1, \ldots, u_n)$, ou seja, $f(t_1, \ldots, t_n) \sim_\varphi f(u_1, \ldots, u_n)$. $\square$

4.6 Notação. Notamos nesta seção, $\sim_\varphi$ por $\sim$, sempre que não houver motivo para confusão.

4.7 Definição.
Se φ é uma coleção de fórmulas de L, então $\Pi_\varphi = \{[t]_{\sim_\varphi} \mid t \text{ é termo em } L\}$.

4.8 Notação. Nesta seção vamos notar Π_φ por Π, sempre que não houver motivo para confusão.

4.9 Definição.
Para cada sinal funcional f de aridade n, definimos uma relação f_φ de Π^n em Π por $\langle [t_1]_\sim, \ldots, [t_n]_\sim \rangle f_\varphi [f(t_1, \ldots, t_n)]_\sim$.

4.10 Lema. *Para cada sinal funcional f em L de aridade n, $f_\varphi : \Pi^n \to \Pi$.*

Suponha que $\begin{cases} \alpha f_\varphi \beta, \\ \alpha f_\varphi \gamma. \end{cases}$

Como $\alpha f_\varphi \beta$, temos que existem termos $t_1, \ldots, t_n$ de L tal que $\{\alpha = \langle [t_1]_\sim, \ldots, [t_n]_\sim \rangle$, $\beta = [f(t_1, \ldots, t_n)]_\sim$.

Como $\alpha f_\varphi \gamma$, temos que existem termos $u_1, \ldots, u_n$ de L tais que $\{\alpha = \langle [u_1]_\sim, \ldots, [u_n]_\sim \rangle$, $\gamma = [f(u_1, \ldots, u_n)]_\sim$.

Daí $\langle [t_1]_\sim, \ldots, [t_n]_\sim \rangle = \langle [u_1]_\sim, \ldots, [u_n]_\sim \rangle$, *logo* $\begin{cases} [t_1]_\sim = [u_1]_\sim \\ \vdots \\ [t_n]_\sim = [u_n]_\sim \end{cases}$

donde $\begin{cases} t_1 \sim u_1, \\ \vdots \\ t_n \sim u_n, \end{cases}$ *logo* $f(t_1, \ldots, t_n) \sim f(u_1, \ldots, u_n)$, *portanto* $[f(t_1, \ldots, t_n)] = [f(u_1, \ldots, u_n)]$, *ou seja,* $\beta = \gamma$.

4.11 Definição.
Seja φ uma coleção de fórmulas de L.
Para cada sinal predicativo p de L de aridade n, definimos p_φ, por
$p_\varphi = \{\langle [t_1]_\sim, \ldots, [t_n]_\sim \rangle \mid \varphi \vdash_{\text{LEC}} p(t_1, \ldots, t_n)\}$.

4.12 Lema.
Sejam $\begin{cases} p \text{ um sinal predicativo em } L \text{ de aridade } n, \\ t_1, \ldots, t_n, u_1, \ldots, u_n \text{ termos em } L. \end{cases}$

Então $t_1 \sim u_1, \ldots, t_n \sim u_n$ implica que $\varphi \vdash_{\text{LEC}} p(t_1, \ldots, t_n)$ sss $\varphi \vdash_{\text{LEC}} p(u_1, \ldots, u_n)$.

Prova:
Assuma as hipóteses.
Suponha que $\varphi \vdash_{\text{LEC}} p(t_1, \ldots, t_n)$. (1)

Das hipóteses, temos que $\begin{cases} \varphi \vdash_{\text{LEC}} t_1 = u_1, \\ \vdots \\ \varphi \vdash_{\text{LEC}} t_n = u_n. \end{cases}$

De SFA, $t_1 = u_1, \ldots, t_n = u_n \vdash_{\text{LEC}} p(t_1, \ldots, t_n) \to p(u_1, \ldots, u_n)$.
Daí, por Tran, $\varphi \vdash_{\text{LEC}} p(t_1, \ldots, t_n) \to p(u_1, \ldots, u_n)$. (2)

De (1), (2) e MP, $\varphi \mid_{\overline{\text{LEC}}}\, p(u_1, \ldots, u_n)$.
Na recíproca, o raciocínio é análogo. $\square$

4.13 Lema.
Com as mesmas hipóteses do lema 4.12, temos que $\langle [t_1]_\sim, \ldots, [t_n]_\sim \rangle \in p_\varphi$ *sss* $\langle [u_1]_\sim, \ldots, [u_n]_\sim \rangle \in p_\varphi$.

4.14 Definição.
Seja φ uma P-coleção de Henkin em L com respeito a LEC.
$I = \langle \Delta, w, s \rangle$ é uma LEC-interpretação canônica para φ se :

- $\Delta = \{[t]_\sim \mid t$ é termo de $L\}$,
- $s(x) = [x]$,
- $w(c) = [c]_\sim$,
- $w(f) = f_\varphi$,
- $w(p) = p_\varphi$, para todo sinal predicativo distinto do sinal de igualdade.

4.15 Lema. *Se I é a* LEC*-interpretação canônica para φ em L, então, para cada termo t de L, $I_D(t) = [t]$.*

Prova:
Dado um termo t em L, temos três casos possíveis.

- Caso t é uma constante c.
  $I_D(t) = I_D(c) = w(c)$ que pela definição 4.14 é igual a $[c]_\sim = [t]_\sim$.
- Caso t é uma variável x.
  $I_D(t) = I_D(x) = s(x) = [x]_\sim = [t]_\sim$.
- Caso t é da forma $f(t_1, \ldots, t_n)$.
  $I_D(t) = I_D(f(t_1, \ldots, t_n)) = w(f)(I_D(t_1), \ldots, I_D(t_n))$, que por HI é igual a $f_\varphi([t_1], \ldots, [t_n]) = [f(t_1, \ldots, t_n)]$.

$\square$

4.16 Lema.
Sejam $\begin{cases} \varphi \textit{ uma } P\textit{-coleção de Henkin para } L \textit{ com respeito a } \text{LEC}, \\ I \textit{ a } \text{LEC}\textit{-interpretação canônica para } \varphi \textit{ em } L. \end{cases}$
Então, para cada fórmula Q de L, se $Q \in \varphi$, então $I_V(Q) =$ v, e se $Q \notin \varphi$, então $I_V(Q) =$ f.

- *Caso Q é da forma $p(t_1, \ldots, t_n)$, onde p é distinto de '$=$'.*
  Se $p(t_1, \ldots, t_n) \in \varphi$, então $\varphi \mid_{\overline{\text{LEC}}}\, p(t_1, \ldots, t_n)$ sss $\langle [t_1], \ldots, [t_n] \rangle \in p_\varphi$ sss $\langle I_D(t_1), \ldots, I_D(t_n) \rangle \in w(p)$ sss $I_V(p(t_1, \ldots, t_n)) =$ v.
- *Caso Q é da forma $t = u$.*
  $t = u \in \varphi$ *sss* $\varphi \mid_{\overline{\text{LEC}}}\, t = u$ *sss* $t \sim u$ *sss* $[t]_{\sim_\varphi} = [u]_{\sim_\varphi}$ *sss* $I_D(t) = I_D(u)$ *sss* $I_V(t = u) = \text{v}$ *sss* $w(f)([t_1], \ldots, [t_n]) = [f(t_1, \ldots, t_n)]$.
- *Os raciocínios são análogos para os demais casos, e já foram apresentados em 9.3.7 e em 10.3.6.*

4.17 Lema.
Seja Γ *uma coleção de fórmulas em* LEC *tal que* $\Gamma \not\vdash P$.
Então existe uma coleção de fórmulas φ *em* LEC *e existe uma linguagem* L *para* LEC

$$\text{tal que} \begin{cases} \Gamma \text{ e } \varphi \text{ são coleções de fórmulas de } L, \\ \Gamma \subseteq \varphi, \\ \varphi \text{ é } P\text{-saturado em } L \text{ com respeito a LEC.} \end{cases}$$

Prova: Análoga à prova apresentada em 10.3.7. □

4.18 Teorema. ***(Completude do Cálculo de Sequentes para*** LEC*)*

- *Se* $\Gamma \models_{\mathrm{LEC}} P$, *então* $\Gamma \vdash_{\mathrm{LEC}} P$.

Prova:
O cálculo de sequentes de LEC satisfaz as três condições gerais de completude definidas em 7.3.2.
Se φ é P-coleção de Henkin em L com respeito a LEC, então φ é P-coleção saturada em L com respeito a LEC, donde, pelo lema 7.4.19, $P \notin \varphi$. Ou seja, o cálculo de sequentes para LEC satisfaz a primeira condição do teorema geral da completude. (1)
Pelo lema 4.16, temos que o cálculo de sequentes para LEC satisfaz a segunda condição do teorema geral da completude. (2)
Pelo lema 4.17, temos que o cálculo de sequentes para LEC satisfaz a terceira condição do teorema geral da completude. (3)
De (1), (2) e (3) e 7.3.2, temos que $\Gamma \models_{\mathrm{LEC}} P$, então $\Gamma \vdash_{\mathrm{LEC}} P$. □

Capítulo 12

Lógica Descritiva Clássica

Este capítulo apresenta uma versão de uma *Lógica Descritiva*, construída sobre a Lógica Equacional Clássica. Como a mesma é erigida a partir de um sistema de Lógica Clássica, e preserva todas as suas leis, chamamos este sistema de *Lógica Descritiva Clássica* ou simplesmente LDC. Esta lógica estuda conjuntamente as propriedades dos conectivos, dos quantificadores, da igualdade e das descrições nela definidas.

As seções §1 e §2 abordam, respectivamente, uma visão geral sobre o tratamento das descrições e a elucidação feita por Russell sobre o tema. As demais seções formalizam a teoria das descrições definidas. A teoria das descrições indefinidas, introduzida neste capítulo, será formalizada posteriormente, no capítulo 13.

§1. Uma visão geral das Lógicas Descritivas

Conforme definido em 3.1.7, consideramos que uma linguagem formal pode possuir dois tipos de samblagens significativas: *termos* ou *fórmulas*. Em linguagem natural, termos são comparados aos *nomes* e fórmulas são comparadas às *frases*.

Enquanto um nome é um símbolo arbitrário atribuído a um objeto do domínio, o qual passa a ser a sua denotação, uma descrição é uma especificação que se aplica a qualquer objeto do domínio que satisfaça a condição formulada [30].

Descrições são frases do tipo 'o x tal que P', ditas *descrições definidas*, e 'um x tal que P', ditas *descrições indefinidas*. Elas ocorrem frequentemente na matemática principalmente no enunciado de constantes, variáveis e funções, dentre outros: '*o* conjunto dos números naturais', 'a variável x onde x não é livre em Q', 'uma função f tal que para qualquer x do domínio verificamos que $f(x) \neq 0$'.

A 'teoria das descrições' é constantemente associada a nomes bem conhecidos do cenário da lógica: Bertrand Russell, John Barkley Rosser e David Hilbert. Desta forma, existem algumas abordagens relacionadas à necessidade de formalizar o uso de descrições nominais presentes em linguagem natural.

A versão da Lógica Descritiva abordada neste trabalho é embasada na Lógica das Descrições de Rosser, mas um tratamento peculiar é feito nas seções §3, §4 e §5 deste capítulo.

O primeiro tratamento desse processo lógico fundamental foi feito por Bertrand Russell no artigo 'On Denoting', nos *Principia Mathematical Philosophy* e na *Introduction to Mathematical Philosophy*, em que a expressões do tipo 'o objeto x tal que Fx' Russell deu o nome de 'descrições'. Embora na *Introduction to Mathematical Philosophy*, Russell faça uma distinção entre descrições definidas e descrições indefinidas, a teoria lógica que lhe seguiu tem se ocupado essencialmente das descrições definidas [30].

A próxima seção apresenta uma breve introdução ao tema, segundo as ideias de Russell. Detalhes adicionais são encontrados em [51–55] e [30].

§2. Ideias centrais da Teoria das Descrições de Russell

Esta seção lida com um esboço da teoria das descrições de Russell. O principal objetivo desta teoria é oferecer uma interpretação semântica apropriada de frases contendo descrições definidas e indefinidas. Foi inicialmente apresentada em seu artigo 'On Denoting' publicado em 1905 na revista *Mind* [51] e posteriormente em [53] e [52].

2.1 Notação.

- x, y são variáveis,
- $C(x)$ é uma proposição ou função proposicional, onde x é essencial e inteiramente indeterminada.

Russell chama de *denoting phrases* (que traduziremos como *frases ou expressões denotativas*), as descrições definidas 'o maior homem no mundo' e as descrições indefinidas 'um número primo', sempre no singular. Também é comum encontrar frases contendo descrições definidas representadas por 'o F é G'. Entretanto elas podem ser representadas de outras formas, empregando pronomes possessivos ('minha mãe', 'seu pai') ou pronomes demonstrativos ('aquele livro', 'esta bicicleta').

Ele distingue três casos onde uma frase denota algo unicamente em virtude de sua forma:

(i) Uma frase pode denotar e não denotar qualquer coisa (exemplo: *o atual rei da França*),

(ii) Uma frase pode denotar um objeto definido (exemplo: *o atual rei da Inglaterra*),

(iii) Uma frase pode denotar algo indefinido ou ambíguo (exemplo: *um homem*).

A forma(i) pode ser considerada uma *Descrição definida vácua* pois é uma descrição que não se aplica a nada, pois, considerando o atual domínio das pessoas vivas não existe alguém que seja o atual rei da França. Outro exemplo é afirmar que 'o seu cachorro mordeu alguém' uma vez que você não possui tal animal. Vamos considerar que nesta teoria será atribuído valor falso para as descrições vácuas.

Vamos nesta seção, interpretar frases que sejam *descrições definidas* ou contenha o artigo definido *o*.

2.2 Princípio. *Uma descrição definida, nesta teoria, só tem sentido se estiver dentro de um contexto. Ou seja, não assume qualquer significado de forma isolada, mas recebe o significado que é atribuído a cada proposição nas quais ela ocorre.*

Um *nome*, por outro lado, possui um sentido e significado por si só, ou seja, o objeto que ele diretamente nomeia ou designa.

Russell introduziu a notação $\iota x : Fx$ para representar 'o x tal que Fx'. O símbolo ι é chamado de *descritor* ou *qualificador*. Segundo [1], um descritor é um 'operador que liga uma variável a uma fórmula para formar um termo'. Essa definição é conhecida pelo termo inglês *v.b.t.o*[1]. Descritores podem ser incluídos em uma teoria de diversas formas. Russell fez uso da *definição contextual* ou *definição em uso* para introduzir o descritor ι; ele apresenta vários contextos típicos do uso de descrições definidas e sugere como frases contendo descrições definidas devem ser analisadas em vez de analisá-las de forma explícita.

Além disso, segundo [56], no simbolismo de Russell, esta expressão 'o x tal que Fx' pode tomar o lugar de um nome, como na fórmula $G(\iota x : Fx)$. Mas mesmo essa expressão se comportando como um nome neste caso, na teoria de Russell ela não será tratada como tal. Pois é necessário ponderar sobre o que tem de ser verdade para que uma frase que contenha a descrição definida 'o x tal que Fx' seja verdadeira:

(i) existir pelo menos um x,

(ii) não existir mais do que um x tal que Fx,

[1] *Variable binding term operator.*

(iii) tudo que é x é Fx.

Baseando-se nessas três afirmações juntas, podemos reduzir essa descrição definida usando os quantificadores e o símbolo de identidade:

$$\exists x(Fx \wedge \forall y(Fy \rightarrow x = y) \wedge Gx)^2$$

Consequentemente observa-se pela fórmula acima que, quando uma descrição definida não é satisfeita, toda a fórmula será falsa.

Segundo [57], as descrições definidas são usadas para escolher objetos únicos acerca dos quais queremos falar. Não se pode ser bem sucedido ao escolher um objeto usando tal expressão, se existirem vários objetos que satisfaçam a descrição que se segue a *o* ou a. Quando alguém usa uma descrição definida, tem um domínio específico em mente.

Ou seja, como as descrições definidas podem ser usadas sem garantia de que exista o objeto por elas descrito, Russell defendeu que descrições definidas não são nomes, quer dizer, não nomeiam objetos. Em alternativa, ele argumentou que uma descrição como 'o F' é uma expressão do mesmo tipo que 'todo o F', 'nenhum F' e 'algum F' – uma expressão quantificada, e não um nome de um certo objeto.

Considere para as afirmações abaixo, uma interpretação e uma formalização utilizando quantificadores e o símbolo de igualdade.

f= O planeta vermelho é Marte.

Uma interpretação I para a afirmação f é:

- $\mathcal{D}(f)$:{conjunto dos planetas}.
- m: Marte.
- Vx: x é vermelho.

A formalização $\exists x(Vx \wedge \forall y(Vy \rightarrow x = y) \wedge (x = m))$ afirma que o único planeta que é vermelho tem a propriedade de ser idêntico a Marte.

g= O pai de Charles II foi executado.

Uma interpretação I para a afirmação g é:

- $\mathcal{D}(g)$:{a pessoa que possua um relacionamento paterno com Charles II}.
- c: Charles II.
- Px: x é o pai de Charles c.
- Ex: x foi executado

A formalização $\exists x(Px \wedge \forall y(Py \rightarrow x = y) \wedge Ex)$ afirma que existe um x que é pai de Charles II e ele foi executado.

As seções seguintes apresentam um tratamento formal das descrições definidas.

---

[2] Ou seja, 'existe um e somente um F e ele é G'.

§3. Uma Linguagem para LDC

3.1 Definição.
Um *alfabeto para* LDC contém todos os sinais de um *alfabeto* para LEC, mais um sinal especial, 'τ', dito *qualificador*, o qual forma termos a partir de uma variável e de uma fórmula. O qualificador 'τ' é denominado *artigo definido*.

3.2 Definição.
Os termos e fórmulas em LDC são todos os termos e fórmulas obtidos pelas regras de formação de LEC, mais os termos da forma $\tau x P$, onde P é uma fórmula em LDC além de ser chamada de *corpo da descrição*. Estes termos $\tau x P$ são chamados de *descrições* em LDC.

3.3 Leitura. O termo $\tau x P$ pode ser lido como "*o x tal que P*".[3]

3.4. **Interpretações para** $\tau x P$.
O termo $\tau x P$ pode ser interpretado de duas formas distintas:
- Em todos os contextos em que a fórmula $\exists! x P$ for verdadeira, $\tau x P$ denota o único objeto do universo de discurso que satisfaz P; nestes casos, a descrição $\tau x P$ é uma *descrição própria*.
- Quando a fórmula $\exists! x P$ for falsa, $\tau x P$ denota um objeto do universo de discurso escolhido arbitrariamente para corresponder a todas as descrições deste tipo; todas as descrições deste gênero são chamadas, nos contextos em que elas ocorrerem, de *descrições impróprias*.

A diferença entre descrições *próprias* e *impróprias* varia conforme a teoria considerada. Nesta teoria apresentada, uma descrição é dita *própria* quando está associada a um objeto do universo de discurso que satisfaz a propriedade especificada por ela. Caso contrário, a descrição é considerada *imprópria*.

3.5. **Descrições próprias e impróprias**
(i) $\tau x(par(x) \land primo(x)) = 2$.
(ii) $\tau x(italiano(x), viajou(x, China)) =$Marco Polo.
(iii) $\tau x(colecao(x) \land \forall y(y \in x \leftrightarrow par(y) \land 0 \leq y < 8)) = \{0, 2, 4, 6\}$.
(iv) $\tau y(y \in \mathbb{Z} \land 0 < y < 1) = \tau x(x \neq x)$.
(v) $\tau z(z \in \mathbb{R} \land 0 < z < 2) = \tau x(x \neq x)$.

Os ítens i, ii e iii acima são considerados descrições próprias. E iv e v, descrição impróprias, pois não denotam nem uma coleção, caso de iii, nem objetos únicos, casos i e ii. Eles são atribuídos a uma descrição que reflita uma ideia de "não existência", neste caso, definida como $\tau x(x \neq x)$. Outros

[3] Tal leitura não reflete precisamente o significado de $\tau x P$, como será visto a seguir.

autores utilizam 0 ou { } para representar a mesma ideia.

Nos termos da forma τxP, as ocorrências de x, livres em P, tornam-se ligadas em τxP, ou seja, como τ liga variáveis em seu escopo, então são necessárias algumas adaptações com respeito às definições presentes em LQC e LEC. Todas as demais definições e convenções presentes do capítulo 11.11, não alteradas nesta seção, continuam valendo para LDC.

3.6 Definição. Um *designador* em LDC é um termo ou uma fórmula em LDC.

3.7 Notação. D, E, F, G são designadores em LDC.

3.8 Definição.
Uma *ocorrência de uma variável x em um designador D* é dita ser *ligada em D* se a mesma ocorrer em um subdesignador[4] de D de uma das formas $\forall\, xP$, $\exists\, xP$ ou $\tau\, xP$, caso contrário, esta ocorrência é dita ser *livre em D*.

3.9 Exemplo.
No designador $D : x = 2 \wedge \tau x(p(x, y)) = z$, dizemos que em $x = 2$ existe uma ocorrência de x livre em D; z é uma ocorrência livre de z em D e em $\tau x(p(x, y))$, existe uma ocorrência ligada de x em D e uma ocorrência livre de y em D.

3.10 Definição.
Uma *variável* é dita ser *livre em um designador* se esta possuir pelo menos uma ocorrência livre neste designador; da mesma forma esta *variável* é dita ser *ligada em um designador* se esta possuir pelo menos uma ocorrência ligada neste designador. Uma variável é dita ser *livre/ligada em uma coleção de designadores* se ela for livre/ligada em pelo menos um elemento desta coleção.

3.11 Exemplo. Pela definição 3.10 e exemplo 3.9, a variável x é livre e também é ligada em D. As variáveis z e y são livres em D.

3.12 Definição.

- $\forall xP, \exists xP$ são ditas *fórmulas quantificadas*,
- τxP é dita *fórmula qualificada.*

A definição 5.1.27, concernente à instanciação de uma variável por um termo, e válida para LQC, é adaptada para LDC.

[4]Isto é, um designador que ocorre em D, o qual pode ser outro designador ou o próprio D.

3.13 Definição. (Instanciação de uma variável por um termo em um designador)
A *instanciação* de x por t em um designador D, notada por $D(x|t)$, é a fórmula obtida de D substituindo todas as ocorrências livres de x por t, se D não possuir quantificadores ou o descritor τ. Caso exista alguma ocorrência de quantificadores ou o descritor em D, então tal instanciação é definida conforme as seguintes cláusulas, onde x e y são variáveis distintas e $\Psi \in \{\forall, \exists, \tau\}$:

- $(\Psi\, xP)(x|t) = \Psi\, xP$[5];
- $(\Psi\, yP)(x|t) = \begin{cases} * \;\; \Psi\, yP(x|t), \text{ se } x \text{ não é livre em } P \text{ ou } y \text{ não é livre em } t; \\ * \;\; \Psi\, zP(y|z)(x|t), \text{ se } x \text{ é livre em } P \text{ e } y \text{ é livre em } t, \\ \quad\;\; \text{onde } z \text{ é a primeira variável não livre em } \{t, P\}. \end{cases}$

3.14 Definição.
Uma *ocorrência* de um designador em um designador E em LDC é dita *real em* E se a mesma não suceder em E um dos sinais '$\forall$', '$\exists$', ou 'τ'. Um designador D é dito ser *real em* E se D possuir pelo menos uma ocorrência *real em* E.

3.15 Definição.
A substituição de D por D' em E, onde D e D' são ambos termos ou ambos fórmulas, notada por $E(D\|D')$, é o designador obtido de E substituindo todas as ocorrências reais de D por D' em E.

3.16. **Lei da Indução estrutural sobre Designadores em** LDC
Seja $\Delta(D)$ uma propriedade sobre designadores D em LDC.

Se $\begin{cases} \bullet \text{ para cada variável } x, \Delta(x), \\ \bullet \text{ para cada constante } c, \Delta(c), \\ \bullet \text{ para qualquer termo } t_1, \ldots, t_n \text{ e para qualquer sinal funcional} \\ \quad f\ n\text{-ário, se } \Delta(t_1), \ldots, \Delta(t_n), \text{ então } \Delta(f(t_1, \ldots, t_n)), \\ \bullet \text{ para qualquer termo } t_1, \ldots, t_n \text{ e para qualquer sinal predicativo} \\ \quad p\ n\text{-ário, se } \Delta(t_1), \ldots, \Delta(t_n), \text{ então } \Delta(p(t_1, \ldots, t_n)), \\ \bullet \text{ para qualquer } P, \Delta(P) \text{ implica em } \Delta(\neg P), \\ \bullet \text{ para qualquer } P \text{ e } Q \text{ e para qualquer } \# \in \{\to, \wedge, \vee\}, \Delta(P) \text{ e } \Delta(Q) \\ \quad \text{implica em } \Delta(P\#Q), \\ \bullet \text{ para qualquer } x, \text{ para qualquer } P \text{ e para qualquer } \Psi \in \{\forall, \exists, \tau\}, \Delta(P) \\ \quad \text{implica em } \Delta(\Psi\, xP), \end{cases}$

então, para qualquer designador D em LDC, $\Delta(D)$.

3.17 Definição. (Sobre o Grau)

- $\mathrm{gr}(x) = 0.$
- $\mathrm{gr}(c) = 0.$

[5]Nada é feito pois não existem variáveis livres.

- $\mathrm{gr}(f(t_1,\ldots,t_n)) = 1 + \mathrm{gr}(t_1) + \ldots + \mathrm{gr}(t_n)$.
- $\mathrm{gr}(p(t_1,\ldots,t_n)) = 1 + \mathrm{gr}(t_1) + \ldots + \mathrm{gr}(t_n)$.
- $\mathrm{gr}(\neg P) = 1 + \mathrm{gr}(P)$.
- $\mathrm{gr}(P\#Q) = 1 + \mathrm{gr}(P) + \mathrm{gr}(Q)$, onde $\# \in \{\to, \wedge, \vee\}$.
- $\mathrm{gr}(\Psi\, xP) = 1 + \mathrm{gr}(P)$, onde $\Psi \in \{\forall, \exists, \tau\}$.

3.18. **Lei da Indução sobre o grau de Designadores em** LDC
Seja $\Delta(D)$ uma propriedade sobre designadores D em LDC.
Se, para qualquer designador D em LDC, a validade de Δ para qualquer designador E tal que $\mathrm{gr}(E) < \mathrm{gr}(D)$ implica em $\Delta(D)$, então, para qualquer designador D em LDC, $\Delta(D)$.

3.19 Definição. (**Designadores do mesmo tipo**)
Dois designadores são ditos serem do mesmo tipo se uma das seguintes condições forem satisfeitas:
- ambos são constantes.
- ambos são variáveis.
- ambos são termos funcionais.
- ambos são fórmulas atômicas.
- ambos são negações.
- ambos são implicações.
- ambos são conjunções.
- ambos são disjunções.
- ambos são fórmulas universais.
- ambos são fórmulas existenciais.
- ambos são descrições.

Especificamos a seguir uma relação entre designadores, a qual é uma extensão da relação de congruência entre fórmulas, definida na página 89.

3.20 Definição. Definimos nas cláusulas abaixo a congruência entre designadores em LDC, considerando $\Psi \in \{\forall, \exists, \tau\}$ e $\# \in \{\wedge, \vee, \to\}$:
- $x \approx_c y$ se, e somente se, x e y são variáveis idênticas ($x = y$),
- $b \approx_c c$ se, e somente se, b e c são constantes idênticas ($b = c$),
- $f(t_1,\ldots,t_n) \approx_c g(u_1,\ldots,u_r)$ se, e somente se, $f = g, n = r$ e, para cada $i \in \{1,\ldots,n\}, t_i \approx_c u_i$,
- $p(t_1,\ldots,t_n) \approx_c q(u_1,\ldots,u_r)$ se, e somente se, $p = q, n = r$ e, para cada $i \in \{1,\ldots,n\}, t_i \approx_c u_i$,
- $\neg P \approx_c \neg Q$ se, e somente se, $P \approx_c Q$,
- $P_1\#Q_1 \approx_c P_2\#Q_2$ se, e somente se, $P_1 \approx_c P_2$ e $Q_1 \approx_c Q_2$,
- $\Psi\, xP_1 \approx_c \Psi\, yP_2$ se, e somente se, $\begin{cases} y \text{ não é livre em } \forall\, xP_1, \\ P_1(x|y) \approx_c P_2. \end{cases}$

- Se D_1 e D_2 forem designadores de tipos diferentes, então $D_1 \not\approx_c D_2$[6].

§4. Uma Semântica para LDC

As definições referentes à semântica de LEC, enunciadas na seção 11.2, continuam válidas em LDC, mas uma extensão da definição de LEC-interpretação para LDC é apresentada a seguir.

4.1 Definição. Uma LDC-interpretação é uma quádrupla $I = \langle \Delta, w, s, d_0 \rangle$, onde $\begin{cases} \langle \Delta, w, s \rangle \text{ é uma LEC-interpretação,} \\ d_0 \in \Delta. \end{cases}$

4.2 Definição.
Seja $I = \langle \Delta, w, s, d_0 \rangle$ uma LDC-interpretação.
Definimos duas funções I_D e I_V conforme já é feito para $\langle \Delta, w, s \rangle$, se as seguintes condições são satisfeitas:

(i) se existe um único $d_1 \in \Delta$ tal que $I(x|d_1)_V(P) = v$, então $I_D(\tau\, xP) = d_1$,

(ii) se não existe $d \in \Delta$ tal que $I(x|d)_V(P) = v$ ou existem pelo menos dois $d \in \Delta$ tal que $I(x|d)_V(P) = v$, então $I_D(\tau\, xP) = d_0$.

Nesta semântica, como apresentado em 3.4, d_0 denota um objeto do universo de discurso escolhido arbitrariamente para corresponder a todas as descrições impróprias. Ou seja, todas as descrições impróprias são iguais a algum objeto dentro da interpretação, neste caso, d_0. Em uma outra semântica definida por [13], d_0 é o conjunto vazio, se a teoria em questão aceita o conjunto vazio.

4.3 Definição.
Considera-se que todas as descrições impróprias são iguais a algum objeto dentro da interpretação. d_0 é um objeto dentro da interpretação que identifica variáveis impróprias. Cada sistema lógico pode atribuir um valor distinto para d_0 dentro do seu domínio. Alguns sistemas atribuem o valor de d_0 ao conjunto vazio, mas, isso é válido desde que a teoria aceite o conjunto vazio.

§5. Um Cálculo de Sequentes para LDC

Esta seção apresenta um cálculo de sequentes para LDC.

O cálculo de sequentes para LDC possui todas as leis primitivas de LEC, mais algumas leis primitivas concernentes a descrições definidas.

Um esboço geral do cálculo de sequentes para LDC é apresentado:

---

[6] Ou seja, não é verdade que $D_1 \approx_c D_2$.

- Leis estruturais
  - $*$ reflexividade,
  - $*$ transitividade,
  - $*$ monotonicidade.
- Leis de introdução de conectivos
  - $*$ regra da dedução,
  - $*$ $\wedge$-introdução,
  - $*$ $\vee$-introdução,
  - $*$ $\neg$-introdução.
- Leis de eliminação de conectivos
  - $*$ modus ponens,
  - $*$ $\wedge$-eliminação,
  - $*$ prova por casos,
  - $*$ $\neg$-eliminação.
- Leis de introdução de quantificadores
  - $*$ generalização,
  - $*$ $\exists$-introdução.
- Leis de eliminação de quantificadores
  - $*$ $\forall$-eliminação,
  - $*$ $\exists$-eliminação.
- Esquemas primitivos da Igualdade
  - $*$ reflexividade da Igualdade,
  - $*$ esquema da Substituição em termos funcionais,
  - $*$ esquema da Substituição em fórmulas atômicas.
- Esquemas primitivos das descrição
  - $*$ descrição própria,
  - $*$ descrições congruentes,
  - $*$ descrições impróprias.

As leis definidas em LEC, devidamente traduzidas, valem também para LDC, com exceção do *esquema da substituição da igualdade para termos* apresentado na página 259. Este esquema não é válido nesta forma em LDC, por supor implicitamente um fato não necessariamente presente em LDC, devido à sua distinta estrutura sintática, na qual termos podem ter variáveis ligadas.

Nesta seção nós falaremos somente da Lógica Descritiva Clássica. Assim, para dizer que P é consequência de Γ em LDC, notaremos isto por $\Gamma \vdash P$.

Esquemas Primitivos da Descrição

5.1. **Descrição Própria** $\vdash \exists!\, xP \to P(x|\tau\, xP)$.

5.2. **Descrições Equivalentes** $\vdash \forall\, x(P \leftrightarrow Q) \to \tau\, xP = \tau\, xQ$.

5.3. **Descrições Congruentes**
Se y não é livre em P, então $\vdash \tau\, xP = \tau\, yP(x|y)$.

5.4. **Descrições Impróprias**[7] $\neg\exists!\, xP,\ \neg\exists!\, yQ \vdash \tau\, xP = \tau\, yQ$.

Os resultados abaixo podem ser provados a partir das leis primitivas de LDC.

Leis Básicas da Descrição

5.5. **Descrição na Existência e Unicidade**. Se y não é livre em $\tau\, xP$, então valem os seguintes sequentes que são equivalentes em LDC:

(i) $\exists! xP$.

(ii) $\forall\, y(P(x|y) \leftrightarrow y = \tau\, xP)$.

(iii) $P(x|\tau\, xP) \wedge \forall\, y(P(x|y) \rightarrow y = \tau\, xP)$.

Esqueleto da Prova: A prova desta lei decorre da Lei da Descrição Própria e da Lei da Absorção na Unicidade. □

5.6. **Congruência**
As seguintes proposições referentes à congruência de designadores em LDC são válidas:

(i) Se t é congruente a t', então $\vdash t = t'$.

(ii) Se P é congruente a P', então $\vdash P \leftrightarrow P'$.

5.7 Notação. Sejam D_1 e D_2 dois designadores, tal que ambos são termos ou ambos são fórmulas. A samblagem '$D_1 \equiv D_2$' representa a fórmula '$D_1 = D_2$', se D_1 e D_2 são ambos termos, e representa a fórmula '$D_1 \leftrightarrow D_2$' se D_1 e D_2 são ambos fórmulas. A samblagem '$D_1 \not\equiv D_2$' representa a samblagem '$\neg(D_1 \equiv D_2)$'.

5.8 Definição. Se $D \approx_c E$, então $\vdash D \equiv E$.

Prova:
Seja $\Delta(D)$ a propriedade 'para cada E, se $D \approx_c E$, então $\vdash D \equiv E$'.
Suponha que $D \approx_c E$.

- Caso D é uma variável x
  Então E também é x.
  Mas, por RI, $\vdash x = x$, donde $\vdash D = E$.
- Caso D é uma constante c, então o raciocínio é análogo ao caso anterior.
- Caso D é da forma $f(t_1, \ldots, t_n)$

  Então E é da forma $f(u_1, \ldots, u_n)$, onde $\begin{cases} t_1 \approx_c u_1 \\ \ldots \\ t_n \approx_c u_n. \end{cases}$

[7] Se existem duas descrições impróprias, então elas são iguais.

Por HI, $\begin{cases} \vdash t_1 \approx_c u_1 \\ \dots \\ \vdash t_n \approx_c u_n, \end{cases}$ daí, por STF, $\vdash f(t_1, \dots, t_n) = f(u_1, \dots, u_n)$,
ou seja, $\vdash D \equiv E$.

- Caso D é da forma $p(t_1, \dots, t_n)$
  Então E é a forma $p(u_1, \dots, u_n)$, onde $\begin{cases} t_1 \approx_c u_1 \\ \dots \\ t_n \approx_c u_n. \end{cases}$
  Por HI, $\begin{cases} \vdash t_1 = u_1 \\ \dots \\ \vdash t_n = u_n, \end{cases}$ e daí, por SFA, $\vdash p(t_1, \dots, t_n) \leftrightarrow p(u_1, \dots, u_n)$,
  ou seja, $\vdash D \equiv E$.
- Caso D é da forma $\neg P$
  Então E é da forma $\neg Q$, onde $P \approx_c Q$.
  Por HI, $\vdash P \leftrightarrow Q$, daí $\vdash \neg P \leftrightarrow \neg Q$, logo $\vdash D \equiv E$.
- Caso D é da forma $P_1 \# P_1$, onde $\# \in \{\rightarrow, \wedge, \vee\}$
  Então E é da forma $Q_1 \# Q_2$ onde $\begin{cases} P_1 \approx_c Q_1 \\ \dots \\ P_2 \approx_c Q_2. \end{cases}$
  Por HI, $\begin{cases} \vdash P_1 \leftrightarrow Q_1 \\ \dots \\ \vdash P_2 \leftrightarrow Q_2, \end{cases}$ onde, por LSQ, $\vdash P_1 \# P_2 \leftrightarrow Q_1 \# Q_2$, ou seja,
  $\vdash D \equiv E$.
- Caso D é $\Psi x P$, onde $\Psi \in \{\forall, \exists, \tau\}$
  Então E é da forma $\Psi y Q$, onde y não é livre em $\Psi x P$ e
  $P(x|y) \approx_c Q$.
  Por HI, $\vdash P(x|y) \leftrightarrow Q$, daí, por Gen, $\vdash \forall y(P(x|y) \leftrightarrow Q)$, donde, por LSQ, temos que $\left\{ \begin{array}{l} \vdash \forall y P(x|y) \leftrightarrow \forall y Q, \\ \vdash \exists y P(x|y) \leftrightarrow \exists y Q, \end{array} \right\}$ (1) e também, por DE, temos que $\vdash \tau y P(x|y) = \tau y Q$.
  Temos que $\begin{cases} \vdash \forall x P \leftrightarrow \forall y P(x|y), \\ \vdash \exists x P \leftrightarrow \exists y P(x|y). \end{cases}$
  Por RI e DC, temos também que $\vdash \tau x P = \tau y P(x|y)$.
  De (1) e (2), temos que $\left\{ \begin{array}{l} \vdash \forall x P \leftrightarrow \forall y P, \\ \vdash \exists x P \leftrightarrow \exists y P, \\ \vdash \tau x P \leftrightarrow \tau y P. \end{array} \right\}$ (3)

De (3), temos que $\vdash \Psi x P \equiv \Psi y Q$, ou seja, $\vdash D \equiv E$.

$\square$

Antes da formulação das regras da substituição de LDC é necessária uma reformulação de mais alguns conceitos já presentes em LQC e em LEC, concernentes à *substituição de designador por designador em um designador* e *escopo de uma variável em um designador*.

Abaixo é estendida a definição 10.1.35, de *escopo de uma variável em uma fórmula*, para *escopo de uma variável em um designador*.

5.9 Definição.
Um designador D é dito *estar no escopo de uma variável x em um designador E* se E possuir um subdesignador de uma das formas $\forall x R$, $\exists x R$ ou $\tau x R$, tal que D é real em R.[8]

Devido à existência de variáveis ligadas em termos em LDC, o esquema e a regra da substituição da igualdade precisam ser reformulados. Faz-se necessária também a formulação de um esquema da instanciação da igualdade para termos, uma vez que, em LDC, a operação de instanciação de variáveis por termos em termos não é idêntica à operação de substituição. O esquema e a regra da substituição da igualdade para fórmulas, bem como o esquema da instanciação da igualdade para fórmulas de LEC são também válidos em LDC, e serão repetidos abaixo por comodidade.

5.10. **Esquema da Substituição da Igualdade**
Considere que D_1, D_2 e G são todos termos ou todos fórmulas.
Se $x_1, \ldots, x_n$ são as variáveis livres em $\{D_1, D_2\}$ tais que G está em E no seu escopo, então $\forall\, x_1 \ldots \forall\, x_n (D_1 \equiv D_2) \vdash E(G \| D_1) \equiv E(G \| D_2)$.

5.11. **Esquema da Substituição da Igualdade para Termos**
Se $x_1, \ldots, x_n$ são as variáveis livres em $\{t_1, t_2\}$ tais que v está em u no seu escopo, então $\forall\, x_1 \ldots \forall\, x_n (t_1 = t_2) \vdash u(v \| t_1) = u(v \| t_2)$.

5.12. **Regra da Substituição da Igualdade para Termos**
Se $\begin{cases} \Gamma \vdash t_1 = t_2, \\ v \text{ não está, em } u\text{, no escopo de nenhuma variável livre em } \Gamma \text{ e em } \{t_1, t_2\}, \end{cases}$
então $\Gamma \vdash u(v \| t_1) = u(v \| t_2)$.

Nas linguagens para LDC a instanciação de variáveis por termos em termos não é idêntica à substituição de variáveis por termos em termos, daí faz-se necessária uma formulação explícita de um esquema de instanciação da igualdade para termos, no qual não há restrições de aplicação.

[8] Se x está no escopo forte de y em D, então x está no escopo de y em D, mas a recíproca não é necessariamente verdadeira.

5.13. **Esquema da Instanciação da Igualdade para Termos**
$t_1 = t_2 \vdash u(x|t_1) = u(x|t_2)$.

5.14. **Esquema da Instanciação da Igualdade para Fórmulas**
$t_1 = t_2 \vdash Q(x|t_1) \leftrightarrow Q(x|t_2)$.

O *Esquema* e a *Regra da Substituição da Igualdade para Fórmulas*, bem como o *Esquema da Instanciação da Igualdade para Fórmulas*, concernentes a LEC, valem da mesma forma em LDC. Os mesmos são novamente, por comodidade, enunciados a seguir.

5.15. **Esquema da Substituição da Igualdade para Fórmulas**
Se $x_1, \ldots, x_n$ são as variáveis livres em $\{t_1, t_2\}$ tais que v está em Q no seu escopo, então $\forall x_1 \ldots \forall x_n(t_1 = t_2) \vdash Q(v\|t_1) \leftrightarrow Q(v\|t_2)$.

5.16. **Regra da Substituição da Igualdade para Fórmulas**
Se $\begin{cases} \Gamma \vdash t_1 = t_2, \\ v \text{ não está, em } Q\text{, no escopo de nenhuma variável livre em } \Gamma \text{ e em } \{t_1, t_2\}, \end{cases}$
então $\Gamma \vdash Q(v\|t_1) \leftrightarrow Q(v\|t_2)$.

Em LDC é possível que fórmulas ocorram em termos, no caso das descrições. Por este motivo, formulamos a seguir o Esquema e a Regra da Substituição da Equivalência para Termos. São listados também o Esquema e Regra da Substituição da Equivalência para Fórmulas, os quais são idênticos à versão apresentada para LEC.

5.17. **Esquema da Substituição da Equivalência para Termos**
Se $x_1, \ldots, x_n$ são as variáveis livres em $\{P_1, P_2\}$ tais que S está em u no seu escopo, então $\forall x_1 \ldots \forall x_n(P_1 \leftrightarrow P_2) \vdash u(S\|P_1) = u(S\|P_2)$.

5.18. **Regra da Substituição da Equivalência para Termos**
Se $\begin{cases} \Gamma \vdash P_1 \leftrightarrow P_2, \\ S \text{ não está, em } u\text{, no escopo de nenhuma variável livre em } \Gamma \text{ e em } \{P_1, P_2\}, \end{cases}$
então $\Gamma \vdash u(S\|P_1) = u(S\|P_2)$.

O *Esquema* e a *Regra da Substituição da Equivalência para Fórmulas* são válidos da mesma forma tanto em LQC e LEC como em LDC[9]. Os mesmos são novamente enunciados a seguir.

5.19. **Esquema da Substituição da Equivalência para Fórmulas**
Se $x_1, \ldots, x_n$ são as variáveis livres em $\{P_1, P_2\}$ tais que S está em Q no seu escopo, então $\forall x_1 \ldots \forall x_n(P_1 \leftrightarrow P_2) \vdash Q(S\|P_1) \leftrightarrow Q(S\|P_2)$.

[9] Em LPC o *Esquema da Substituição da Equivalência para Fórmulas* é válido sem quaisquer restrições, simplesmente por não haver em suas linguagens variáveis ligadas em fórmulas.

5.20. **Regra da Substituição da Equivalência para Fórmulas**

Se $\begin{cases} \Gamma \vdash P_1 \leftrightarrow P_2, \\ S \text{ não está, em } Q, \text{ no escopo de nenhuma variável livre em } \Gamma \text{ e em } \{P_1, P_2\}, \end{cases}$

então $\Gamma \vdash Q(S\|P_1) \leftrightarrow Q(S\|P_2)$.

5.21. **Reflexividade da Equivalência** $\vdash P \leftrightarrow P$.

Os Esquemas da Substituição da Igualdade para Termos, da Igualdade para Fórmulas, da Equivalência para Termos e da Equivalência para Fórmulas podem ser expressos em uma só formulação.

5.22. **Esquema Geral da Substituição**

Se $x_1, \ldots, x_n$ são as variáveis livres em $\{D_1, D_2\}$ tais que G está em E no seu escopo, então $\forall x_1 \ldots \forall x_n (D_1 \equiv D_2) \vdash E(G\|D_1) \equiv E(G\|D_2)$.

As Regras da Substituição da Igualdade para Termos, da Igualdade para Fórmulas, da Equivalência para Termos e da Equivalência para Fórmulas também podem ser expressas em uma só formulação, como faremos logo a seguir.

5.23. **Regra Geral da Substituição**

Se $\begin{cases} \bullet\ \Gamma \vdash D_1 \equiv D_2, \\ \bullet\ G \text{ não está, em } E, \text{ no escopo de nenhuma variável livre em } \Gamma \text{ e em} \\ \quad \{D_1, D_2\}, \end{cases}$

então $\Gamma \vdash E(G\|D_1) \equiv E(G\|D_2)$.

Antes de provar a Regra Geral da Substituição considere o lema abaixo.

5.24 Lema.

Se $\begin{cases} \Gamma \vdash P, \\ \textit{para cada } i \in \{1, \ldots, n\}, x_i \textit{ não é livre em } \Gamma \textit{ ou } x_i \textit{ não é livre em } P, \end{cases}$

então, $\Gamma \vdash \forall \overrightarrow{x} P$.

Prova da Regra Geral da Substituição: Assuma a hipótese.

Seja $\overrightarrow{x}$ a lista de variáveis tal que G está em E no seu escopo.

Temos que, para cada $i \in \{1, \ldots, n\}$, x_i não é livre em Γ ou x_i não é livre em $D_1 \equiv D_2$.

De HGb e lema 5.24, temos que $\Gamma \vdash \forall \overrightarrow{x} (D_1 \equiv D_2)$.

Mas, pelo EGS, $\forall \overrightarrow{x} (D_1 \equiv D_2) \vdash E(G\|D_1) \equiv E(G\|D_2)$.

Portanto $\Gamma \vdash E(G\|D_1) \equiv E(G\|D_2)$. $\square$

Finalmente, os Esquemas da Instanciação da Igualdade para Termos e para Fórmulas também admitem uma única formulação.

5.25. **Esquema Geral da Instanciação para a Igualdade**

$t_1 = t_2 \vdash E(x|t_1) \equiv E(x|t_2)$.

§6. Correção e Completude do cálculo de sequentes de LDC com respeito à semântica de LDC

Tudo que foi provado para LEC, com algumas alterações, também é válido para LDC.

6.1 Correção do cálculo de sequentes de LDC com respeito à semântica de LDC

6.1 Teorema. ***(Correção)***

- *Se* $\Gamma \vdash_{\mathrm{LDC}} P$*, então* $\Gamma \models_{\mathrm{LDC}} P$.

Prova:

- Queremos mostrar que o esquema da Descrição Própria é correto.
  Temos $\exists! xP \vdash_{\mathrm{LDC}} P(x|\tau xP)$ é um exemplar de DP.
  Suponha que $I_V(\exists! xP) = \mathrm{v}$.
  Então, pela proposição 11.2.7, existe um único $d \in \Delta$ tal que $I(x|d)_V(P) = \mathrm{v}$.
  Daí $I_D(\tau xP) = d$.
  $I_V(P(x|\tau xP)) = I(x|I_D(\tau xP))_V(P) = I(x|d)_V(P) = \mathrm{v}$.
- Queremos mostrar que o esquema das Descrições Congruentes é correto.
  Temos $\tau xP = \tau yP(x|y)$ é um exemplar de DC.
  Seja y não livre em P.
  Seja $I = \langle \Delta, w, s, d_0 \rangle$.
  Pela propriedade 5.2.18, $I(x|d)_V(P) = I(y|d)_V(P(x|y))$ donde $I(x|d)_V(P) = \mathrm{v}$ sss $I(y|d)_V(P(x|y)) = \mathrm{v}$. (1)
  $*$ Caso existe um único d tal que $I(x|d)_V(P) = \mathrm{v}$.
  Temos que $I_D(\tau xP) = d$, pela semântica de LDC.
  De (1), existe um único $d \in \Delta$ tal que $I(y|d)_V(P(x|y)) = \mathrm{v}$, donde $I_D(\tau yP(x|y)) = d$, portanto $I_V(\tau xP = \tau yP(x|y)) = \mathrm{v}$.
  $*$ Caso não existe um único d tal que $I(x|d)_V(P) = \mathrm{v}$.
  Daí não existe um único d tal que $I(y|d)_V(P(x|y)) = \mathrm{v}$.
  Então $I_D(\tau xP) = do$ e $I_D(\tau yP(x|y)) = d_0$, logo $I_V(\tau xP = \tau yP(x|y)) = \mathrm{v}$.
- Queremos mostrar que o esquema das Descrições Impróprias é correto.
  Seja DI igual a $\neg\exists!\, xP,\ \neg\exists!\, yQ \vdash \tau\, xP = \tau\, yQ$.
  Seja $I = \langle \Delta, w, s, d_0 \rangle$ uma LDC-interpretação para P e Q, e suponha que $I_V(\neg\exists! xP) = \mathrm{v}$, $I_V(\neg\exists! yQ) = \mathrm{v}$, daí, pela propriedade 11.2.7, não existe um único $d \in \Delta$ tal que $I(x|d)_V(P) = \mathrm{v}$ e não existe um único $d \in \Delta$

tal que $I(y|d)_V(Q) = \text{v}$. Logo $I_D(\tau xP) = d_0$ e $I_D(\tau yQ) = d_0$, portanto $I_V(\tau xP = \tau yQ) = \text{v}$

$\square$

6.2 Completude do cálculo de sequentes de LDC com respeito à semântica de LDC

6.2 Definição. φ é P-coleção de Henkin em L com respeito a LDC

$$\rightleftharpoons$$

- φ é P-saturado em L com respeito a LDC,
- para toda fórmula Q de L e para toda variável x, se $\exists xQ \in \varphi$, então existe uma constante c em L, tal que $Q(x|c) \in \varphi$,
- para toda fórmula Q de L, para toda variável x e para toda constante c em L, se para cada constante c em L, $Q(x|c) \in \varphi$, então $\forall xQ \in \varphi$,
- existe uma constante a_0 fixa em L tal que, para toda fórmula Q de L e para toda variável x, se $\exists!xQ \notin \varphi$, então '$\tau xQ = a_0$'$\in \varphi$.

6.3 Fato.
Seja P-coleção de Henkin para L tal que $\tau x(x \neq x) = c_0 \in \varphi$.
Para qualquer termo, temos que este fato acontece.

6.4 Definição.
Seja $\begin{cases} \varphi \text{ uma } P\text{-coleção de Henkin para } L \text{ com respeito a LDC,} \\ I = \langle \Delta, w, s, d_0 \rangle \text{ uma LDC-interpretação.} \end{cases}$
I é uma LDC-interpretação canônica para φ se
(i) $\Delta = \{[t]_\sim | t \text{ é termo em } L\}$.
(ii) para qualquer constante c em L, $w(c) = [c]_{\sim_\varphi}$.
(iii) para qualquer $n > 0$, para qualquer sinal funcional n-ário em L, $w(f) = f_\varphi$.
(iv) para qualquer $n \geq 0$, para qualquer sinal predicativo n-ário em L, $w(p) = p_\varphi$.
(v) para qualquer variável, $s(x) = [x]_{\sim_\varphi}$.
(vi) $d_0 = [c_0]$.

6.5 Proposição.
Seja I uma LDC-interpretação canônica e Δ um universo de discurso.
Dado $d \in \Delta$, existe uma constante c em L tal que $I_D(c) = d$.
- existe um termo t em L tal que $d = [t]$.
- existe uma constante c em L tal que $t = c \in \varphi$
- existe uma constante c em L tal que $t \approx c \in \varphi$.
- existe uma constante c em L tal que $d = [c]$.
- existe uma constante c em L tal que $d = I_D(c)$.

6.6 Teorema.

Se $\begin{cases} \varphi \textit{ é } P\textit{-coleção de Henkin para } L \textit{ com respeito a } \text{LDC}, \\ I \textit{ é uma } \text{LDC}\textit{-interpretação canônica para } \varphi, \end{cases}$

então, para cada fórmula Q em L, $Q \in \varphi$ sss $I_V(Q) = \text{v}$.

Prova:

Assuma a hipótese.

Seja $I = \langle \Delta, w, s, d_0 \rangle$.

Para cada designador E em L, seja $\Lambda(E)$ a propriedade

$\begin{cases} \text{(i)} & \text{se } E \text{ é termo em } L\text{, então } I_D(E) = [E]. \\ \text{(ii)} & \text{se } E \text{ é fórmula em } L\text{, então } E \in \varphi \text{ sss } I_V(E) = \text{v}. \end{cases}$

Se $d \in \Delta$, então existe um termo t em L, tal que $d = [t]$, daí, dado uma variável x não livre em t, temos que $\exists x(x = t) \in \varphi$, daí existe uma constante c em L tal que $c = t \in \varphi$, donde $\varphi \vdash_{\text{LDC}} c = t$, logo $c \approx_\varphi t$, ou seja, $[c]_\approx = [t]_\approx$, daí $[c]_\approx = d$, portanto acabamos de provar que, para cada $d \in \Delta$, existe uma constante c em L tal que $I_D(c) = d$. (1)

- Caso E é da forma $\tau x Q$ e existe um único $d \in \Delta$ tal que $I(x|d)_D(Q) = \text{v}$.
  Seja d este único d tal que $I(x|d)_D(Q) = \text{v}$.
  Por HI, $I_D(c) = [c]$, daí existe uma constante c em L tal que $[c] = d$.
  $I_V(Q(x|c)) = I(x|I_D(c))_V(Q) = I(x|[c])_D(Q) = I(x|d)_D(Q) = \text{v}$, daí, por HI, $Q(x|c) \in \varphi$ e daí $\varphi \vdash_{\text{LDC}} Q(x|c)$. (2)
  Suponha por absurdo que $\neg\bar{\exists} x Q \in \varphi$, daí $\varphi \vdash_{\text{LDC}} \neg\bar{\exists} x Q$.
  Sejam x_1, x_2 variáveis distintas não livres em Q.
  Temos que $\vdash_{\text{LDC}} \neg\bar{\exists} x Q \leftrightarrow \exists x_1 \exists x_2 (Q(x|x_1) \wedge Q(x|x_2) \wedge x_1 \neq x_2)$, daí $\varphi \vdash_{\text{LDC}} \exists x_1 \exists x_2 (Q(x|x_1) \wedge Q(x|x_2) \wedge x_1 \neq x_2)$, donde $\exists x_1 \exists x_2 (Q(x|x_1) \wedge Q(x|x_2) \wedge x_1 \neq x_2) \in \varphi$, e daí existem constantes c_1 e c_2 tais que $(Q(x|x_1) \wedge Q(x|x_2) \wedge x_1 \neq x_2)(x|c_1)(x_2|c_2) \in \varphi$, daí $Q(x|x_1)(x_1|c_1) \wedge Q(x|x_2)(x_2|c_2) \wedge c_1 \neq c_2 \in \varphi$, logo $Q(x|c_1) \wedge Q(x|c_2) \wedge c_1 \neq c_2 \in \varphi$, donde $\begin{cases} Q(x|c_1) \in \varphi, \text{(1)} \\ Q(x|c_2) \in \varphi, \text{(2)} \\ c_1 \neq c_2 \in \varphi. \end{cases}$
  Como $c_1 \neq c_2 \in \varphi$, daí $c_1 = c_2 \notin \varphi$. (3)
  De (1), (2), (3) e HI, $\begin{cases} I_V(Q(x|c_1)) = \text{v}, \\ I_V(Q(x|c_2)) = \text{v}, \\ I_V(c_1 = c_2) = \text{f}. \end{cases}$
  Donde $\begin{cases} I(x|I_D(c_1)_V(Q) = \text{v}, \\ I(x|I_D(c_2)_V(Q) = \text{v}, \\ I_D(c_1) \neq I_D(c_2) = \text{f}, \end{cases}$
  o que é absurdo, logo $\neg\bar{\exists} x Q \notin \varphi$, portanto $\bar{\exists} x Q \in \varphi$. (4)
  Se $\neg\exists x Q \in \varphi$, $\varphi \vdash_{\text{LDC}} \neg\exists x Q$, daí $\varphi \vdash_{\text{LDC}} \forall x \neg Q$, donde

$\varphi \vdash_{\text{LDC}} \neg Q(x|c)$, logo $\neg Q(x|c) \in \varphi$, donde $Q(x|c) \notin \varphi$, o que é absurdo.
Logo $\neg\exists xQ \notin \varphi$, donde $\exists xQ \in \varphi$. (5)
De (4) e (5), $\exists! xQ \in \varphi$, daí $\varphi \vdash_{\text{LDC}} \exists! xQ$, daí $\varphi \vdash_{\text{LDC}} \tau xP = c$, logo $\{t|t$ é termo em$L$ e $\varphi \vdash_{\text{LDC}} c = t\} = \{t|t$ é termo em L e $\varphi \vdash_{\text{LDC}} \tau xP = t\}$, ou seja, $[c]_{\approx} = [\tau xP]_{\approx}$, portanto $I_D(\tau xP) = [\tau xP]_{\approx}$.

- Caso $E = \tau xQ$ e não exista um único $d \in \Delta$ tal que $I(x|d)_V(Q) = \text{v}$.
Suponha por absurdo que $\exists! xQ \in \varphi$.
Então $\exists Q \in \varphi$ (1) e $\overline{\overline{\exists}} xQ \in \varphi$ (2).
De (1) temos que existe uma constante c tal que $Q(x|c) \in \varphi$, daí, por HI, $I_V(Q(x|c)) = \text{v}$, daí $I(x|I_D(c))_V(Q) = \text{v}$, ou seja, existe $d \in \Delta$ tal que $I(x|d)_V(Q) = \text{v}$. (3)
De (2), temos que, dadas duas variáveis distintas x_1 e x_2 não livres em Q, $\forall x_1 \forall x_2 (Q(x|x_1) \wedge Q(x|x_2) \to x_1 = x_2) \in \varphi$. (4)
Suponha que $d_1, d_2 \in \Delta$ e $I(x|d_1)_V(Q) = \text{v}$ e $I(x|d_2)_V(Q) = \text{v}$.
De 6.5, existem constantes c_1 e c_2 em L tal que $I_D(c_1) = d_1$ e $I_D(c_2) = d_2$, donde $\begin{cases} I(x|I_D(c_1))_V(Q) = \text{v}, \\ I(x|I_D(c_2))_V(Q) = \text{v}, \end{cases}$ logo, $\begin{cases} I_V(Q(x|c_1)) = \text{v}, \\ I_V(Q(x|c_2)) = \text{v}, \end{cases}$ e daí, por HI, $\begin{cases} Q(x|c_1) \in \varphi, \\ Q(x|c_2) \in \varphi, \end{cases}$ donde, por (4), $c_1 = c_2 \in \varphi$, daí $c_1 \sim c_2$, donde $[c_1] = [c_2]$, mas $[c_1] = I_d(c_1)$ e $[c_2] = I_d(c_2)$, daí $I_D(c_1) = I_D(c_2)$, logo $d_1 = d_2$, ou seja, existe um máximo d tal que $I(x|d)_V(Q) = \text{v}$. (5)
De (3) e (5), temos que existe um único $d \in \Delta$ tal que $I(x|d)_V(Q) = \text{v}$, o que é absurdo, logo $\exists! xQ \notin \varphi$, daí $\neg\exists! xQ \in \varphi$, portanto $\tau xQ = \tau x(x \neq x) \in \varphi$ e daí, de 6.3, como $\tau x(x \neq x) = c_0 \in \varphi$, temos que $\tau x(Q) = c_0 \in \varphi$, daí $\tau x(Q) \sim c_0$, daí $[\tau xQ] = [c_0]$. (6)
Por outro lado, $I_D(\tau xQ) = d_0 = [c_0]$. (7)
De (6) e (7), concluímos que $I_D(\tau xQ) = [\tau xQ]$.

□

6.7 Teorema.
Se $\Gamma \nvdash_{\text{LDC}} P$*, então existe uma linguagem* L *para* LDC *tal que*
$\begin{cases} P \textit{ é fórmula de } L, \\ \varphi \textit{ é coleção de fórmulas de } L, \end{cases}$
e $\begin{cases} \Gamma \in \varphi, \\ \varphi \textit{ é } P\textit{-coleção de Henkin para } L \textit{ com respeito a } \text{LDC} \end{cases}$

Prova:
O raciocínio é análogo à prova feita para LQC em 10.3.7. □

6.8 Teorema. *(Completude do Cálculo de Sequentes para* LDC*)*

• *Se* $\Gamma \models_{\mathrm{LDC}} P$*, então* $\Gamma \vdash_{\mathrm{LDC}} P$*.*

Prova:

O cálculo de sequentes de LDC satisfaz as três condições gerais de completude definidas em 7.3.2.

Se φ é P-coleção de Henkin em L com respeito a LDC, então φ é P-coleção saturada em L com respeito a LDC, donde, pelo lema 7.4.19 e a definição 6.2, $P \notin \varphi$. Ou seja, o cálculo de sequentes para LDC satisfaz a primeira condição do teorema geral da completude. (1)

Pelas definições 11.4.16 e 6.6, temos que o cálculo de sequentes para LDC satisfaz a segunda condição do teorema geral da completude. (2)

Pelo pelo teorema 6.7, temos que o cálculo de sequentes para LDC satisfaz a terceira condição do teorema geral da completude. (3)

De (1), (2) e (3) e 7.3.2, temos que $\Gamma \models_{\mathrm{LDC}} P$, então $\Gamma \vdash_{\mathrm{LDC}} P$. □

Capítulo 13

Lógica das Descrições Indefinidas

Este capítulo apresenta a segunda versão de uma Lógica Descritiva, construída sobre a Lógica Equacional Clássica, e que trata das descrições indefinidas. Chamaremos este sistema de *Lógica das Descrições Indefinidas* ou simplesmente LDI.

Esta lógica estuda conjuntamente as propriedades dos conectivos, dos quantificadores, da igualdade e das *descrições indefinidas*.

§1. Uma Linguagem para LDI

1.1 Definição.
Um *alfabeto para* LDI contém todos os sinais de um *alfabeto* para LEC, mais um qualificador, o 'ε'. O qualificador 'ε' é chamado de *artigo indefinido*.

Segundo [55], o símbolo de Hilbert é a letra grega épsilon, ε. Ele também é denominado *descritor indefinido*, pois permite que nos refiramos a um objeto do domínio de indivíduos que possuem uma propriedade, mesmo que não se saiba exatamente qual é esse objeto.

1.2 Definição.
Termos e fórmulas em LDI são todos os termos e fórmulas obtidos pelas regras de formação de LEC, mais os termos da forma $\varepsilon\, xP$, onde x é uma variável e P é uma fórmula em LDI.
Os termos $\varepsilon\, xP$ são ditos *descrições em* LDI e a fórmula P é chamada de *corpo da descrição* $\varepsilon\, xP$.

1.3 Notação. Adotaremos nesta seção as notações especificadas em 3.1.8.

1.4 Exemplo.
- Termos em LDI:
  - x,
  - c,
  - $f(t_1, \ldots, t_n)$,
  - εxP.
- Fórmulas em LDI:
  - $p(t_1, \ldots, t_n)$,
  - $t = u$,
  - $\neg P$,
  - $P \to Q$,
  - $P \wedge Q$,
  - $P \vee Q$,
  - $\forall\, xP$,
  - $\exists\, xP$.

1.5 Definição.
Termos e fórmulas são ditos *Designadores* em LDI.

1.6 Leitura. O termo $\varepsilon\, xP$ pode ser lido como "*um x tal que P*".[1]

1.7. **Interpretações para** $\varepsilon\, xP$:
O termo $\varepsilon\, xP$ pode ser interpretado de duas formas distintas:
- Nos contextos em que a fórmula $\exists\, xP$ for verdadeira, $\varepsilon\, xP$ denota um objeto do universo de discurso que satisfaz P. Neste caso, a descrição $\varepsilon\, xP$ é uma descrição própria.
- Nos contextos em que a fórmula $\exists\, xP$ for falsa, $\varepsilon\, xP$ denota um objeto do universo de discurso escolhido arbitrariamente para corresponder a todas as descrições deste tipo; tais descrições são ditas *descrições impróprias*.

1.8 Exemplo.
Considerando a interpretação usual para os sinais contidos no termo $\varepsilon x(x \in \mathbb{N} \wedge 2 < x)$, temos que o mesmo pode significar qualquer número natural maior que 2. Tal termo é uma descrição indefinida, pois não possui um significado determinado.

1.9 Exemplo.
Considerando os significados usuais atribuídos aos símbolos contidos nas descrições abaixo, temos:
- Descrição Própria: $\varepsilon\, x(x \in \mathbb{N} \wedge x > 5 \wedge x < 10)$.

[1] Tal leitura não reflete precisamente o significado de $\varepsilon\, xP$, como será visto a seguir.

- Descrição Imprópria: $\varepsilon\, x(x \neq x)$.

1.10 Definição.
Definimos em LDI, de um modo análogo ao realizado para LDC, com as devidas adaptações, *designador em* LDI, *ocorrência de variável em um designador*, *ocorrência ligada de uma variável*, *ocorrência livre de uma variável*, *variável ligada em um designador*, *variável livre em um designador*, *instanciação de variável por termo*, *escopo forte de uma variável* e *aceitação de um termo por uma variável*.

1.11 Convenção. Na formulação das leis lógicas de LDI, só consideraremos doravante instanciações de variáveis por termos em designadores nas quais cada termo seja aceito pela variável correspondente.

§2. Uma Semântica para LDI

As definições referentes à semântica de LEC, enunciadas na seção 11.2, continuam válidas em LDI, com exceção da definição de LEC-interpretação, reformulada abaixo, juntamente com a definição de função escolha.

2.1 Definição.
Considere a definição de *função escolha* definida em 2.3.84.
Podemos ainda definir uma *função escolha* da seguinte forma:

Seja Δ uma coleção não vazia.
Dizemos que f é uma função escolha para Δ se as seguintes condições forem satisfeitas:
(i) f é uma função;
(ii) $\mathcal{D}(f) = \mathcal{P}(\Delta) - \{\emptyset\}$;
(iii) para cada $A \in \mathcal{D}(f), f(A) \in A$.

2.2 Exemplo.
Considere o seguinte domínio: $\Delta = \{3, 7, 9\}$.
Uma função f irá associar, para cada conjunto não vazio, um elemento desse universo de discurso,
$f : P(\{3, 7, 9\}) - \{\emptyset\} \to \{3, 7, 9\}$.
Sendo assim, todas as associações a seguir são resultados da função escolha sobre o este domínio Δ:
- $\{3, 7, 9\} \mapsto 9$,
- $\{3, 7\} \mapsto 3$,
- $\{3, 9\} \mapsto 9$,

- $\{7,9\} \mapsto 7$,
- $\{3\} \mapsto 3$, $\{7\} \mapsto 7$, $\{9\} \mapsto 9$.

2.3 Definição.
Uma LDI-interpretação é uma quíntupla $I = \langle \Delta, w, s, f, d_0 \rangle$,

onde $\begin{cases} \langle \Delta, w, s \rangle \text{ é uma LEC-interpretação,} \\ f \text{ é uma função escolha para } \Delta, \\ d_0 \in \Delta. \end{cases}$

Deste modo, as seguintes condições são satisfeitas:

(i) se existe pelo menos um $d \in \Delta$ tal que $I(x|d)_V(P) = v$, então $I_D(\varepsilon\, xP) = f(\{d|d \in \Delta \wedge I(x|d)_V(P) = v\})$,

(ii) se não existe $d \in \Delta$ tal que $I(x|d)_V(P) = v$, então $I_D(\varepsilon\, xP) = d_0$.

2.4 Definição.

$I_t(\varepsilon xP)$=$\begin{cases} \bullet\ f(\{d|d \in \Delta \text{ e } I(x|d)_f(P) = v\}), \text{ se existe } d \in \Delta \text{ tal que} \\ \quad I(x|d)_f(P) = v, \\ \bullet\ d_0, \text{ caso contrário.} \end{cases}$

§3. Um Cálculo de Sequentes para LDI

Esta seção apresenta um cálculo de sequentes para LDI.

Este se constitui de todas as leis primitivas de LEC, mais os esquemas primitivos da descrição indefinida: descrição própria, das descrições equivalentes e das descrições congruentes.

Analogamente a LDC, todas as leis de LEC, devidamente traduzidas para a linguagem de LDI, também são válidas em LDI, com exceção do esquema da substituição da igualdade para termos.

Nesta seção nós falaremos somente da Lógica das Descrições Indefinidas; assim, para dizer que P é consequência de Γ em LDI, notaremos isto por $\Gamma \vdash P$.

Esquemas Primitivos da Descrição Indefinida

3.1. **Descrição Própria** $\exists\, xP \vdash P(x|\varepsilon\, xP)$.

3.2. **Descrições Equivalentes**[2] $\vdash \forall\, x(P \leftrightarrow Q) \rightarrow \varepsilon\, xP = \varepsilon\, xQ$.

3.3. **Descrições Congruentes**
Se y não é livre em P, então $\vdash \varepsilon\, xP = \varepsilon\, yP(x|y)$.

3.4. **Descrições Impróprias** $\neg\exists\, xP, \neg\exists\, yQ \vdash \varepsilon\, xP = \varepsilon\, yQ$.

[2]Este esquema serve para fazer com que as fórmulas equivalentes P e Q correspondam ao mesmo objeto denotado por $\varepsilon\, xP$, segundo consta em Carrion;Da Costa,1988.

Antes de provar a lei das descrições impróprias vamos apresentar e provar o seguinte lema:

3.5 Lema. $\neg\exists\, xP \vdash \varepsilon\, xP = \varepsilon\, x(x \neq x)$.

Prova:

1	$\neg\exists xP$	pr
2	$\forall x\neg P$	1, NE
3	$\neg P$	2, $\forall$-el
4	$x = x$	RI
5	$\neg\neg(x = x)$	4, DN
6	$\neg(x \neq x)$	5, def
7	$P \leftrightarrow x \neq x$	3, 6, ME
8	$\forall x(P \leftrightarrow x \neq x)$	7, Gen
9	$\varepsilon xP = \varepsilon x(x \neq x)$	8, DE

$\square$

Prova da lei das descrições impróprias:

1	$\neg\exists xP$	pr
2	$\neg\exists yQ$	pr
3	$\varepsilon xP = \varepsilon x(x \neq x)$	1, lema
4	$\varepsilon yQ = \varepsilon y(y \neq y)$	2, lema
5	$\varepsilon x(x \neq x) = \varepsilon y(y \neq y)$	RI, DCg
6	$\varepsilon xP = \varepsilon yQ$	3, 4, 5, STI
7	$P \leftrightarrow x \neq x$	3, 6, ME
8	$\forall x(P \leftrightarrow x \neq x)$	7, Gen
9	$\varepsilon xP = \varepsilon x(x \neq x)$	8, DE

$\square$

Abaixo são listados alguns resultados sintáticos de LDI.

Leis Básicas da Descrição Indefinida

3.6. **Fórmula Existencial** $\vdash \exists\, xP \leftrightarrow P(x|\varepsilon\, xP)$.

1	$\exists xP \to P(x\|\varepsilon xP)$	DP
2	$P(x\|\varepsilon xP) \to \exists xP$	$\exists$-int
3	$\exists xP \leftrightarrow P(x\|\varepsilon xP)$	1, 2, $\wedge$-int

3.7. **Fórmula Universal** $\vdash \forall\, xP \leftrightarrow P(x|\varepsilon\, x\neg P)$.

1	$\exists x\neg P \leftrightarrow \neg P(x\|\varepsilon x\neg P)$	FE
2	$\neg\exists x\neg P \leftrightarrow \neg\neg P(x\|\varepsilon x\neg P)$	1, LSC
3	$\forall x\neg\neg P \leftrightarrow P(x\|\varepsilon x\neg P)$	2, NE, DN
4	$\forall xP \leftrightarrow P(x\|\varepsilon x\neg P)$	3, DN

3.8 Definição.
A *congruência de designadores em* LDI é especificada de um modo análogo ao realizado para LDC.

3.9 Congruência. *As proposições concernentes à congruência, válidas em* LDC*, também valem em* LDI*, com as devidas adaptações para esta.*

3.10 Definição.
Definimos em LDI, de um modo análogo ao realizado para LDC, com as adaptações óbvias, *ocorrência real de um designador*, *designador real*, *substituição de um designador* e *escopo de uma variável*.

3.11 Escólio.

Situamos que
- *o esquema da substituição da igualdade para termos,*
- *a regra da substituição da igualdade para termos,*
- *o esquema da instanciação da igualdade para termos,*
- *o esquema da substituição da igualdade para fórmulas,*
- *a regra da substituição da igualdade para fórmulas,*
- *o esquema da instanciação da igualdade para fórmulas,*
- *o esquema da substituição da equivalência para termos,*
- *a regra da substituição da equivalência para termos,*
- *o esquema da substituição da equivalência para fórmulas,*
- *a regra da substituição da equivalência para fórmulas,*

valem igualmente em LDI.

3.12 Escólio. *O esquema geral da substituição, a regra geral da substituição e o esquema geral da instanciação para a igualdade valem também em* LDI.

Capítulo 14

Conclusões

§1. Considerações Finais

O objetivo primordial desta dissertação de mestrado foi apresentar os principais sistemas da Lógica Clássica.

Especificamente, focamos na Lógica Proposicional Clássica, Lógica Quantificacional Clássica, Lógica Equacional Clássica, Lógica Descritiva Clássica e Lógica das Descrições Indefinidas.

O trabalho se fundamenta no seu aspecto elucidativo, e não em seu caráter totalmente inédito. Ainda assim, apresentamos neste trabalho dois tipos de contribuição.

Do ponto de vista da pura especulação formal no campo das lógicas clássicas, contribuímos na modelagem e no aperfeiçoamento de alguns aspectos semânticos e sintáticos.

No capítulo 8 apresentamos uma abordagem genérica para analisar a validade de fórmulas de uma dada lógica através do método dos tablôs por confutação, desde que algumas condições gerais de correção e completude sejam satisfeitas. Tais condições foram apresentadas em 8.3.2 e 8.3.3. Anteriormente, na seção §3.3, através da definição 3.3.1, foram dadas condições essenciais para que uma lógica seja dotada de uma semântica de valorações, através da qual são definidos, de um modo geral, os conceitos de satisfabilidade, validade e relação de consequência. No capítulo 7, apresentamos as condições gerais de correção e completude de um cálculo de sequentes com respeito a uma semântica de valorações. Os resultados para lógicas específicas foram aplicados nas seções §9.3, para LPC, em §10.3, para LQC, e em 11.4, para LEC. Com isto obtivemos uma estrutura conceitual adequada a diversas lógicas, e daí, sempre que um cálculo de sequentes e uma semântica de valorações preencherem as condições acima indicadas, podemos concluir a correção e completude deste cálculo com respeito a esta semântica, sem necessidade de considerações adicionais, o que proporciona uma grande economia de pensamento.

Outro aspecto tratado neste trabalho destina-se a aperfeiçoar a abordagem de alguns autores, tais como [5], [6] e [2], a qual fixa uma linguagem quando define conceitos úteis para a apresentação de uma semântica de uma certa lógica. Alguns exemplos[1]: Γ é $\mathcal{L}$-satisfatível, se toda interpretação para Γ, satisfaz Γ em L com respeito à lógica $\mathcal{L}$; P é válido, se toda $\mathcal{L}$-interpretação para P satisfaz P em L; $\Gamma \models_{\mathcal{L}} P$, se uma $\mathcal{L}$-interpretação para $\Gamma \cup \{P\}$ que satisfaz Γ, também satisfaz P, em L. Neste último caso, se existir uma interpretação I, ela é obrigada a interpretar todos os sinais não lógicos da linguagem L, mesmo se estiver definindo uma interpretação I para $\Gamma \cup \{P\}$. Um problema é que, se o alfabeto da linguagem L possuir sinais não lógicos que não ocorrem em Γ e P, então será preciso lidar com sinais desnecessários, em vez de lidar apenas com sinais em Γ e em P. Ou seja, esta abordagem obriga que todos os sinais não lógicos que estejam no alfabeto da linguagem L sejam interpretados. A nossa alternativa para tais definições apóia-se nos preceitos do lógico franciscano *William of Ockham*[2]. Ela enuncia tais definições semânticas de forma desvinculada da linguagem[3]. Assim, se queremos definir quando Γ acarreta P em uma dada lógica, basta considerarmos interpretações para $\Gamma \cup \{P\}$, as quais não precisam interpretar sinais não lógicos que não figurem em Γ e P. Outros exemplos: Γ é $\mathcal{L}$-satisfatível, se toda interpretação para Γ satisfaz Γ; P é válido, se toda $\mathcal{L}$-interpretação para P satisfaz P.

Do ponto de vista didático, advogamos para o nosso trabalho o mérito de representar um passo proeminente para o entendimento da formalização intermediária de sistemas lógicos pelos estudantes das Ciências Informáticas e de áreas interdisciplinares, tais como a Filosofia, a Matemática, o Direito, a Linguística, entre outras.

---

[1]Exemplos relacionados com satisfabilidade, validade e relação de consequência.

[2]Ele enuncia que a explicação para um fato deve assumir o menor número de pressupostos possíveis, eliminando aqueles que não contribuem para tal explicação. Dessa forma, a chamada 'Navalha de Ockham' sugere que os excessos desnecessários numa definição, argumentação ou explicação sejam cortados, preferindo a solução mais simples. Mais detalhes sobre este pensamento podem ser encontrados em [58] e [59].

[3]Alguns conceitos vinculados a uma dada linguagem são os de coleção P-saturada, P-coleção de Henkin e coleção maximal não trivial.

§2. Trabalhos futuros

Nos capítulos 11, 12 e 13, uma linguagem, uma semântica e um cálculo de sequentes foram apresentados para a Lógica Equacional Clássica, a Lógica Descritiva Clássica e a Lógica das Descrições Indefinidas, respectivamente. Uma complementação deste trabalho está na apresentação de sistemas de tablôs para estas lógicas. Estes resultados também podem ser aplicados de forma prática, através da automatização do raciocínio, tratada nos derradeiros capítulos.

Um outro trabalho futuro é estender os resultados desta dissertação para algumas lógicas deviantes, tais como a Lógica Paraconsistente, a Lógica Paracompleta, a Lógica Intuicionista, a Lógica Relevante, dentre outras.

A partir desta dissertação, temos a intenção de publicar um livro texto contendo os resultados apresentados neste trabalho, e outros materiais apresentados em trabalhos anteriores a nossa linha de pesquisa.

Referências Bibliográficas

1 DA COSTA, N. C. A.; KRAUSE, D. **Lógica**. 2005. Disponível em: <http://www.cfh.ufsc.br/~dkrause/pg/cursos/Novo5.pdf>. Acesso em: 22 dez. 2009.

2 BELL, J. L.; MACHOVER, M. **A Course in Mathematical Logic**. Amsterdam: North Holland, 1977.

3 MENDELSON, E. **Introduction to Mathematical Logic**. 3. ed. New York: Wadsworth and Brooks, 1987.

4 NERODE, A.; SHORE, R. A. **Logic for Applications**. D Ner: Springer-Verlag, 1993.

5 ENDERTON, H. B. **A Mathematical Introduction to Logic**. 2. ed. San Diego, California: Academic Press, 1972.

6 MACHOVER, M. **Set theory, Logic and Their Limitations**. Cambridge, England: Cambridge University Press, 1995.

7 NOLT, J.; ROHATYN, D. **Lógica**. São Paulo: Makron Books, McGraw-Hill, 1991. (Coleção Schaum).

8 POGORZELSKI, W. A. **Notions and Theorems of elementary formal logic**. Poland: Warsaw University - Bialystok Branch, 1994.

9 COPI, I. M. **Introdução à Lógica**. São Paulo: Mestre Jou, 1978.

10 FILHO, C. F. **História da Computação: O Caminho do Pensamento e da Tecnologia**. Porto Alegre: EDIPUCRS, 2007.

11 EBBINGHAUS, H. D.; FLUM, J.; THOMAS, W. **Mathematical Logic**. 2. ed. New York: Springer–Verlag, 1984.

12 MORTARI, C. A. **Introdução à Lógica**. São Paulo: Editora UNESP, 2001.

13 SUPPES, P. **Axiomatic Set Theory**. New York: Dover Publications, 1972.

14 HALMOS, P. R. **Naive Set Theory**. New York: Springer, 1974.

15 ENDERTON, H. B. **Elements of Set Theory**. New York: Academic Press, 1977.

16 DA COSTA, N. C. A. On the theory of inconsistent formal systems. **Notre Dame Journal of Symbolic Logic**, v. 11, p. 497–510, 1974.

17 DA COSTA, N. C. A.; BÉZIAU, J.-Y.; BUENO, O. A. S. Aspects of paraconsistent logic. **Logic Journal of the IGPL**, v. 3, n. 4, p. 597–614, 1995.

18 BUCHSBAUM, A.; PEQUENO, T. H. C. Uma família de lógicas paraconsistentes e/ou paracompletas com semânticas recursivas. **Coleção Documentos - Série Lógica e Teoria da Ciência**, v. 14, p. 1–59, 1993.

19 DA COSTA, N. C. A.; MARCONI, D. A note on paracomplete logic. **Rendiconti Dell'Accademia Nazionale Dei Lincei**, v. 80, p. 504–509, 1986.

20 MARGARIS, A. **First Order Mathematical Logic**. New York: Dover Publications, Inc., 1990.

21 FILHO, E. A. **Iniciação à Lógica Matemática**. São Paulo: Nobel, 2002.

22 WIKIPEDIA. **Nou**. Disponível em: <http://pt.wikipedia.org/wiki/NOU>. Acesso em: 14 jan. 2010.

23 WIKIPEDIA. **Nem**. Disponível em: <http://pt.wikipedia.org/wiki/NEM>. Acesso em: 14 jan. 2010.

24 SMULLYAN, R. M. **First Order Logic**. New York: Springer-Verlag, 1968.

25 BETH, E. W. Semantic entailment and formal derivability. **Medlingen der Koninklijke Nederlandse Akademie van Wetenschapen**, v. 18, n. 13, p. 309–342, 1955.

26 HINTIKKA, K. J. J. Form and content in quantification theory. **Acta Philosohica Fennica**, v. 8, p. 7–55, 1955.

27 SMULLYAN, R. M. **Lógica de Primeira Ordem**. São Paulo: Editora UNESP, 2009.

28 NETO, M. C. L. **Um Método dos Tablôs por Prova Direta para a Lógica Clássica**. 77 p. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, 2004.

29 HEGENBERG, L.; ANDRADE E SILVA, M. F. **Novo Dicionário de Lógica**. Rio de Janeiro: Pós-Moderno, 2005.

30 BRANQUINHO, J.; MURCHO, D.; GOMES, N. G. **Enciclopédia de termos lógico-filosóficos**. São Paulo: Martins Fontes, 2006.

31 RUSSELL, B. **The Analysis of Matter**. New York: Routledge, 2001.

32 WAZLAWICK, R. S. **Metodologia de Pesquisa para Ciência da Computação**. Rio de Janeiro: Elsevier, 2008.

33 RUGABER, S. **Thoughts on the Structure of CS Dissertations**. 1963. Disponível em: <http://www.cc.gatech.edu/fac/Spencer.Rugaber/txt/thesis.html>. Acesso em: 15 dez. 2009.

34 DA SILVA, F. S. C.; FINGER, M.; DE MELO, A. C. V. **Lógica para Computação**. São Paulo: Thomson Learning, 2006.

35 GOLDBALTT, R. **Topoi: The Categorial Analysis of Logic**. Amsterdam: North Holland, 1984.

36 LAMBEK, J.; SCOTT, P. J. **Introduction to Higher-Order Categorical Logic**. Cambridge, U.K: Cambridge University Press, 1986.

37 MENEZES, P. B.; HAEUSLER, E. H. **Teoria das Categorias para Ciência da Computação**. Porto Alegre, RS: Artmed, 2009.

38 CANTOR, G. **Contributions to the founding of the theory of transfinite numbers**. New York: Dover Publications, 1915.

39 DE CARVALHO, R. L.; DE OLIVEIRA, C. M. G. M. **Modelos de Computação e Sistemas Formais**. Rio de Janeiro: DCC/IMC, COPPE/Sistemas, NCE/UFRJ, 1998.

40 ROGERS, H. J. **Theory of recursive functions and effective computability**. New York-St. Louis-San Francisco-Toronto-London-Sydney-Hamburg: McGraw-Hill Book Company, 1967. 482 p. (McGraw-Hill Series in Higher Mathematics).

41 GENTZEN, G. **The Collected Papers of Gerhard Gentzen**. Amsterdam/London: North Holland, 1969. Edited by M. E. Szabo.

42 FITCH, F. B. **Symbolic Logic**. New York: The Ronald Press Company, 1952.

43 SOLOW, D. **How to read and do proofs: and introduction to mathematical thought process**. New York: John Wiley & Sons, 2001.

44 BACKX, A. C.; TAVARES, R. N. O. **Indução Matemática**. Niterói: Instituto de Lógica, Filosofia e Teoria da Ciência da Universidade Federal Fluminense, 1984. 94 p.

45 BONEVAC, D.; ASHER, N. M.; KOONS, R. C. **Logic, Sets and Functions**. EUA: Kendall/Hunt Publishing Company, 1999.

46 SOMINSKII, I. S. **The method of mathematical induction**. New York/London: Blaisdell Publishing Company, 1961.

47 BUCHSBAUM, A. B.; PEQUENO, T. O método dos tableaux generalizado e sua aplicação ao raciocínio automático em lógicas não clássicas. **O que nos faz pensar — Cadernos do Departamento de Filosofia da PUC-Rio**, v. 3, p. 81–96, set. 1990.

48 BUCHSBAUM, A.; GARCIA, A. Reasoning by tableaux in direct form. Unpublished.

49 GRIES, D.; SCHEIDER, F. B. **A Logical Approach to Discrete Math**. Berlin: Springer-Verlag, 1993. ISBN 3-540-94115-0.

50 SHOENFIELD, J. **Mathematical Logic**. [S.l.]: Addison-Wesley, 1967.

51 RUSSEL, B. On denoting. **Mind – New Series**, v. 14, p. 479–493, oct. 1905.

52 RUSSEL, B. **Introduction to Mathematical Philosophy**. London: George Allen and Unwin, 1993. First published in 1919.

53 RUSSEL, B.; WHITEHEAD, A. N. **Principia Mathematica**. Cambridge: Cambridge University Press, 1910.

54 ROSSER, J. B. **Logic for Mathematicians**. EUA: McGraw-Hill Book Company, 1953.

55 CARRION, R.; COSTA, N. C. A. D. **Introdução à Lógica Elementar com o símbolo de Hilbert**. Porto Alegre: Ed. da Universidade UFRGS, 1988.

56 BOSTOCK, D. **Intermediate Logic**. EUA: Oxford University Press, 1997.

57 NEWTON-SMITH, W. H. **Lógica um curso introdutório**. Lisboa: Gradiva, 1998.

58 THORBURN, W. M. The myth of Occam's razor. **Mind**, v. 27, p. 345–353, 1918. A detailed study of what Ockham actually wrote and what others wrote after him. Disponível em: <http://www.allisons.org/ll/Images/People/Ockham/>.

59 BOEHNER, P. **Philosophical Writings - Ockham - A selection**. New York: Bobbs-Merrill Company, Inc., 1964.

Publicação de Artigos e Apresentação de Trabalhos

Resumos publicados em anais de eventos

BUCHSBAUM, A.; GARCIA, A. **On a Direct Method of Generating Tableau Proofs**. In: Science, Truth and Consistency CLE/AIPS - Dedicated to Newton da Costa's 80th Anniversary, 2009, Campinas. Annals of the Science, Truth and Consistency CLE/AIPS - Dedicated to Newton da Costa's 80th Anniversary, v. 1, p. 79-79, Aug., 2009.

Apresentação de trabalho

GARCIA, A. **Introdução à Lógica Formal e ao Raciocínio Lógico**. 8ª Semana de Ensino, Pesquisa e Extensão, Universidade Federal de Santa Catarina, UFSC, Florianópolis, Out., 2009.

Apresentação de trabalho

GARCIA, A. **On a Direct Method of Generating Tableau Proofs**. Oral presentation in English in: Science, Truth and Consistency CLE/AIPS - Dedicated to Newton da Costa's 80th Anniversary, Universidade Estadual de Campinas, UNICAMP, Campinas, SP, Aug., 2009.

Artigo completo publicado em periódico

GARCIA, A.; BUCHSBAUM, A. **About LaTeX tools that students of Logic should know**. Practex Journal, Issue 2010-1. Issue theme: "LaTeX Academic Work Bench". Disponível em: <http://www.tug.org/pracjourn/2010-1/garcia/>

Artigo completo aceito para publicação

BUCHSBAUM, A.; GARCIA, A. **O Método da Confutação Generalizado**. In: 9a. Conferencia Iberoamericana en Sistemas, Cibernética e Informática (CISCI 2010), 2010, Orlando, Florida, EE.UU.
Qualis Internacional nível C.

Índice Remissivo