UNIVERSIDADE FEDERAL DA PARAÍBA

Centro de Ciências e Tecnologia - CCT

Departamento de Sistemas e Computação - DSC

Coordenação de Pós-Graduação em Informática - COPIN

UMA LÓGICA PROPOSICIONAL DE RESOLUÇÃO DE PROBLEMAS

ANTONIO CARLOS DE ALBUQUERQUE

Campina Grande - PB
1998

Antonio Carlos de Albuquerque

Uma Lógica Proposicional de Resolução de Problemas

Dissertação submetida ao Curso de Pós-Graduação em Informática do Centro de Ciências e Tecnologia da Universidade Federal da Paraíba, como requisito parcial para obtenção do grau de Mestre em Informática.
Área de Concentração: Ciência da Computação
Sub-Área: Inteligência Artificial.

Orientador: Prof. Manoel Agamemnon Lopes

Campina Grande - PB
1998

DIGITALIZAÇÃO:
UFCG
SUPRA OMNES LUX LUCIS
SISTEMOTECA UFCG

A3451 Albuquerque, Antônio Carlos de.
Uma lógica proposicional de resolução de problemas / Antônio Carlos de Albuquerque. - Campina Grande, 1998.
91 f.

Dissertação (Mestrado em Informática) - Universidade Federal da Paraíba, Centro de Humanidades, 1998.
"Orientação : Prof. M.Sc. Manoel Agamenon Lopes".
Referências.

1. Inteligência Artificial. 2. Resolução de Problemas. 3. Lógica. 4. Dissertação - Informática. I. Lopes, Manoel Agamenon. II. Universidade Federal da Paraíba - Campina Grande (PB). III. Título

CDU 004.8(043)

UMA LÓGICA PROPOSICIONAL DE RESOLUÇÃO DE PROBLEMAS

ANTONIO CARLOS DE ALBUQUERQUE

DISSERTAÇÃO APROVADA EM 27.11.98

PROF. MANOEL AGAMEMNON LOPES, M.Sc
Presidente

Evandro de Barros Costa
PROF. EVANDRO DE BARROS COSTA, D.Sc
Examinador

PROF. RUY JOSÉ GUERRA BARRETO DE QUEIROZ, Ph.D
Examinador

CAMPINA GRANDE - PB

Dedicatória

Dedico este trabalho

A Edna Maria, minha esposa, pelo carinho e constante incentivo;

A Sara Elisabete, minha filhinha, porque a vida é um contínuo renovar-se. Sua graça e inocência nos faz encarar com entusiamo até o mais tenebroso dos teoremas;

A todos os filósofos e matemáticos cristãos, "que o Senhor os tenha em Sua Glória";

Aos demais filósofos e matemáticos, "que o Senhor tenha misericórdia desses infiéis";

Resumo

A Teoria Geral de Problemas (TGP) é um formalismo que representa qualquer problema através de uma estrutura algébrica, a partir da captação de elementos invariantes que caracterizam a natureza de um problema. A fim de facilitar o processo de resolução, a TGP também contempla a quebra de um problema em subproblemas, por meio de critérios bem definidos. A TGP contribui com a Inteligência Artificial no contexto de resolução de problemas. Nesse sentido, firmamos como objetivo principal desta dissertação desenvolver Uma Lógica Proposicional de Resolução de Problemas (LRP) como uma ferramenta correta e completa de inferência lógica que seja útil na apresentação, manipulação e formulação de conhecimento que envolva a entidade problema como vista pela TGP.

Abstract

General Theory of Problem (TGP) is a formalism that represents any problem through an algebraic structure, from captation of invariant elements that characterize the nature of a problem. To facilitate the process of resolution, TGP also contemplates the break of a problem in subproblems, by means of well defined criterions. TGP contributes to Artificial Intelligence in the context of problem solving. In this sense, we set as the main goal of this dissertation to develop A Proposicional Logic of Problem Solving (LRP) as a correct and complete tool of logic inference that is to be useful in the presentation, manipulation and formulation of knowledge that involves the entity problem as viewed by TGP.

Agradecimentos

A Deus, por razões totais de fé e consciência, e por razões parciais da Ciência;

A minha mãe, D. Chiquinha, cujas mãos minha vida ajudou a calejar;

A minha vó (in memoriam), D. Francisquinha, que me ensinou o gosto pela leitura;

A minha mãe do coração, Joselma Albuquerque, que projetou em seu coração que eu deveria ser um homem da Ciência, e ainda hoje investe nisso;

Ao restante de minha família que sempre acreditou em mim;

Ao meu orientador, Prof. Manoel Agamemnon, porque acredita e faz com que engenheiros se tornem lógicos; e, indo além, a orientação acadêmica gerou entre nós uma grande amizade, de modo que nutro profundo respeito pela sua pessoa;

Aos professores e funcionários do curso de pós-graduação;

Ao Prof. Edilson Ferneda porque nos recebeu calorosamente no mestrado; e esse calor nunca cedeu um grau sequer ao longo do tempo;

Ao Prof. Homero Cavalcanti, pela sua dedicação e paciência;

A Aninha e Vera que nunca nos dizem um "não", mesmo no meio do sufoco;

Aos colegas do mestrado, especialmente Mário Ernesto, pelas sessões de oráculo, e Aidre Cunha, pelos papos e "pegas".

Lista de Figuras

3.2.1 Visão Esquemática da Estrutura *Problema* 18

3.2.2 Visão Esquemática de *Solução de Problema* 21

6.2.1 Visão Esquemática da Máquina de Inferência da LRP 43

8.3.1 Visão Esquemática de Interpretação 75

Sumário

1 Introdução **1**

1.1 Motivação 1

1.2 Apresentação 3

2 Teoria Geral de Problemas (TGP) e IA **6**

2.1 A Inteligência Artificial como Ciência 6

2.2 Resolução de Problemas na IA 8

2.3 A TGP no contexto da IA 14

3 TGP - Noções Básicas **16**

3.1 Da Intuição para a Formalização 16

3.2 Problema e Solução de Problema 17

3.3 Construção de Problemas 22

4 A Questão da Negação na LRP **25**

4.1 Lógica 25

4.2 A Questão da Negação 27

4.2.1 Negação de Termos e Proposições 27

4.2.2 A Negação na LRP 28

5 Lógica de Resolução de Problemas **32**

5.1 Lógica de Esquemas de Problemas (LEP) 32

5.1.1 Sistema Formal Dedutivo na LEP 33

5.1.2 Prova Formal na LEP 34

5.2 Lógica de Esquemas de Solução (LES) 36

5.3 Lógica de Resolução de Problemas (LRP) 38

6 Sistema Formal Dedutivo e Atribuição de Verdade na LRP **40**

6.1 Sistema Formal Dedutivo para a LRP 40

6.2 Máquina de Inferência da LRP 43

6.3 Atribuição de Verdade na LRP 44

6.4 Equivalência de fbfs na LRP 47

6.5 Novo Teorema da Dedução para a LRP 47

6.6 Resultados do Teorema 4.4 (**[LOP 85]**) 53

7 Corretude, Consistência, Completude e Compacidade na LRP **55**

7.1 Corretude do Sistema Formal Dedutivo da LRP 55

7.2 Maximalidade Consistente de um Conjunto de Sentenças da LRP 58

7.3 Compacidade de um Sistema de Sentenças da LRP 61

7.4 Completude do Sistema Formal Dedutivo da LRP 64

8 Introdução a uma Lógica de Primeira Ordem para Resolução de Problemas (LPORP) **72**

8.1 As Linguagens de Primeira Ordem e a Formalização do Conhecimento Declarativo 73

8.2 Extensão da Linguagem da LRP 74

8.3 Interpretação e Verdade 75

8.4 Uma Interpretação para a LPORP 77

8.5 Considerações sobre um Sistema Formal Dedutivo para a LPORP 79

9 Conclusão **80**

Bibliografia 85

Apêndice A 89

Apêndice B 90

Capítulo 1

Introdução

Neste capitulo procuraremos explanar o mais breve possível os motivos que nos levaram a desenvolver o presente trabalho. Também é mostrada uma descrição sucinta dos capitulos 2 a 9.

1.1 Motivação

No ambiente de pós-graduação do Departamento de Sistemas e Computação (DSC) têm sido desenvolvidos alguns trabalhos envolvendo já os resultados obtidos da Teoria Geral dos Problemas (TGP). Esses trabalhos, até o momento, têm buscado a apresentação de metodologias para a resolução de determinadas classes de problemas; e tais metodologias são desenvolvidas como se se buscassem heurísticas[1]. A contribuição positiva a ser salientada nessa interação da TGP e da IA é que algumas heurísticas podem ser desenvolvidas baseadas na proposta da TGP de se tratar um problema como

---

[1] As metodologias visam sistematizações de solução ou até mesmo metodizações (computacionalidade = método de solução)

uma estrutura. Não uma estrutura simples como a proposta em **[BAN 80]**, mas uma estrutura muito mais rica do tipo apresentado em **[LOP 81]**.

O desenvolvimento dos resultados obtidos na dissertação de mestrado de Gilson Santos **[SAN 96]**, embora pudesse ser mostrado de um modo puramente algébrico, foi apresentado numa linguagem declarativa lembrando de certo modo o PROLOG. O ponto a ressaltar é de caráter semelhante ao do parágrafo anterior, isto é, se posteriormente alguém se propuser a desenvolver uma linguagem de programação para a metodologia de resolução proposta em **[SAN 96]**, um ponto de partida poderia ser o exame dessa "linguagem declarativa" usada ali.

O trabalho desenvolvido por Vera Prudência dos Santos **[PRU 97]** envolvendo o *Desenvolvimento de uma Metodologia para Resolução de Problemas do Tipo Determine*, é mais uma contribuição com o objetivo de fortalecer os laços entre a TGP e a Computabilidade.

Os trabalhos acima citados, na realidade, se constituem adendos a uma séries de outros (**[OLI 85]**,**[SIL 85]**,**[BED 87]**,**[ALM 90]**) que foram desenvolvidos tendo a TGP como uma base essencialmente matematica, ora explorando os aspectos formais da resolução de problemas, ora tratando das possibilidades de se computar problemas.

Sendo conhecido o potencial das lógicas para a tratativa de conhecimento declarativo, tanto na apresentação como na manipulação de tal conhecimento (**[KOW 79]** e **[RIN 88]**), nos propomos desenvolver, com a devida orientação, *Uma Lógica de Resolução de Problemas*.

O objetivo de nosso trabalho é ampliar o alcance da lógica de resolução de problemas como apresentada em **[LOP 85]**, de modo que, através da adição de novos elementos que caracterizem o potencial computacional dessa lógica, possamos utilizar a natureza do novo sistema formal na apresentação, manipulação e formulação de conhecimento envolvendo a entidade *problema* da maneira como é vista pela TGP.

Alimentamos o sentimento (baseado não em simples quimera) de que a LRP e o sistema formal dedutivo proposto para a mesma, se comportem nos aspectos principais, aos similares da lógica proposicional clássica. Assim, a LRP será a base para o desenvolvimento de uma lógica de ordem superior que trate de modo mais completo a expressão de idéias no mundo dos problemas.

Desde já sintonizados com o espirito de pesquisa e desenvolvimento de conhecimento cientifico que têm norteado o DSC, objetivamos desenvolver, como uma contribuição da TGP, um instrumento teórico que se mostre uma ferramenta útil no ambiente da Lógica, da Computabilidade, e da Inteligência Artificial.

Assim, faço minhas as palavras de Bernard Bolzano (1781-1844), em sua vasta obra sobre lógica, *Wissenschaftslehe*, "...tenho a esperança de que o pouco aqui apresentado possa agradar a alguém e encontrar mais tarde alguma aplicação" **[POL 95]**

1.2 Apresentação

O presente trabalho está disposto em 9 capítulos, sendo este o primeiro. O leitor que já tiver um conhecimento *a priori* da TGP e estiver perfeitamente familiarizado com a contextualização da TGP na IA pode, sem qualquer perda de continuidade, passar para o capitulo 4 e seguir em diante. A leitura a partir do quarto capitulo, entretanto, deve ser feita de modo seqüencial.

No capítulo 2 apresentamos um liame histórico do presente trabalho com a IA através da introdução da TGP como uma contribuição à resolução de problemas dentro da visão da Inteligência Artificial, buscando, entretanto, apresentar a TGP como uma alternativa à abordagem tradicional de *problemas* que a IA fazia até então.

No capítulo 3 introduzimos os conceitos para a formalização do tipo *problema*. É discutido que os aspectos sintáticos da questão *problema* são captados para formar a estrutura *problema*. Introduzimos também as noções básicas da Teoria Geral de

Problemas necessárias para a compreensão dos capítulos 4 em diante. São apresentadas a definição de problema e solução de problema; exemplos elucidativos são dados envolvendo a estrutura *problema*. Apresentamos as noções introdutórias de construção de problemas e a prova do teorema 2.3 de **[LOP 85]**.

No capítulo 4 apresentamos um brevíssimo conteúdo inicial introduzindo algo a respeito da filosofia da lógica. Tal breviário será de grande utilidade para se entender a natureza da negação na lógica, e particularmente a negação na LRP. Recomendamos aos menos familiarizados com a filosofia da lógica que leiam o quarto capítulo.

No capítulo 5 introduzimos os elementos formais da Lógica de Resolução de Problemas (LRP) como originalmente apresentada em **[LOP 85]** São apresentadas como construções independentes a Lógica de Esquemas de Problemas (LEP) e a Lógica de Esquemas de Solução (LES), bem como alguns dos respectivos resultados obtidos em cada uma delas. Finalmente, é apresentada a LRP como uma junção da LEP e da LES.

No capítulo 6 apresentamos um sistema formal dedutivo para a LRP e alguns resultados já apresentados em **[LOP 85]** Apresentamos um esboço da máquina de inferência da LRP e introduzimos formalmente um modelo de atribuição de verdade para a LRP. Neste capítulo, ressaltamos de forma especial, a apresentação do Novo Teorema da Dedução para a LRP. Tal teorema é completo ao contemplar todas as regras de inferência do sistema formal dedutivo da LRP.

No capítulo 7 apresentamos alguns resultados originais (para a lógica em discussão) tais como os teoremas da consistência, corretude, compactarão e completude da LRP, apontando assim para as possibilidades computacionais da lógica em estudo. O desenvolvimento de tais teoremas segue, sem traumas, as linhas gerais encontradas para os similares da lógica proposicional clássica. Os resultados iniciais se constituem nas evidências etimológicas que apontam para o bom comportamento da LRP nos moldes da lógica proposicional clássica.

No capitulo 8 discorremos sobre as possibilidades de a LRP vir a ser uma base para o desenvolvimento de uma lógica de primeira ordem (LPO) ainda visando a resolução de problemas. A intenção é apenas apresentar os elementos básicos para a definição da lógica. Mesmo assim, discorreremos um pouco a respeito das potencialidades de uma LPO. Também introduziremos algumas noções sobre uma interpretação para uma LPO para que haja uma rápida comparação com as funções de atribuição de verdade inerentes às lógicas sentenciais. O objetivo desse capitulo é apenas sugerir que as idéias sobre o mundo dos problemas de acordo com a TGP podem ser descritas de modo potencialmente mais rico através de uma LPO para problemas.

No capitulo 9 tecemos as considerações finais a respeito do presente trabalho, e apontamos algumas linhas de pesquisa para futuros trabalhos que permitam uma continuidade ao que foi feito nesta dissertação.

Capítulo 2

Teoria Geral de Problemas (TGP) e IA

Esta breve explanação tem como objetivo situar o nosso trabalho dentro do contexto da Inteligência Artificial no ambiente de Resolução de Problemas. Outrossim, também objetivamos situar o nosso trabalho como uma contribuição positiva para uma tratativa de problemas a nível macro dentro de uma Teoria Geral de Problemas, sob o ponto de vista da representação e manipulação de conhecimento sobre *problemas* através de *Uma Lógica Proposicional de Resolução de Problemas*.

2.1 A Inteligência Artificial como Ciência

O campo da Inteligência Artificial é uma das mais fascinantes e desafiantes áreas da ciência da computação. A IA tem levantado algumas das mais profundas questões sobre a própria natureza da humanidade e aquilo que distingue os humanos das outras criaturas, sejam elas animadas ou inanimadas. O objetivo declarado da pesquisa em inteligência artificial é "ensinar" as máquinas a "pensar", isto é, a mostrar certas caracteristicas geralmente associadas com a inteligência humana. Nessa base, a Inteligência Artificial tem lutado para se tornar uma ciência no estrito termo da palavra.

Enquanto a IA gradualmente emergia como ciência, seu nascimento se processava de certo modo penoso. As razões para tão doloroso nascimento não são dificeis de serem entendidas: incerteza quanto ao que se constituia o objeto de interesse, incerteza sobre os objetivos da pesquisa, e incerteza sobre qual abordagem se mostraria mais efetiva. Deste ponto de vista a disciplina se parecia mais com a psicologia e as ciências sociais do que com as ciências exatas, tais como a matemática e a fisica.

A medida que os pioneiros na pesquisa da IA ajudavam a definir a disciplina, parecia existir tantas escolas de pensamento quanto era o número de pesquisadores. Isto levou a muitas e sérias controvérsias e confrontações. O que seria, ou o que deveria ser "a verdade" dentro da IA ? Essa não era uma pergunta fácil de responder, pois deveria se levar em consideração as dificuldades da IA quando comparada com as ciências exatas, nas quais "a verdade" tem uma existência relativamente objetiva determinada de maneira não ambigua pelo experimento. Entendendo-se "verdade"[2] neste contexto como sendo a formalização teórica da realidade observável.

Uma das definições da IA creditada a Marvin Minsky, um dos mais profícuos colaboradores da segunda geração de pesquisadores da IA, talvez resuma com clareza os objetivos da área: *"Inteligência Artificial é a ciência de fazer as máquinas fazerem coisas que requereriam inteligência se feitas pelos homens"* **[FIR 89]**. E segundo H. P. Winston, *"Inteligência Artificial é o estudo das idéias que habilitam os computadores a serem inteligentes"* **[WIN 84]**.

Como se pode perceber das idéias acima, o termo "Inteligência Artificial" foi cunhado para designar o ramo da ciência envolvido com o estudo de como fazer com que as tarefas normalmente executadas por seres humanos passassem a ser realizadas por

---

1 A "verdade" está sujeita aos vários graus de subjetivismo de quem a define, sendo, portanto, o resultado de uma atribuição pelos motivos mais diversos.

máquinas. A designação "inteligência artificial" é reputada como sendo criação de John McCarthy, um dos mais conhecidos cientistas da IA da segunda geração de pesquisadores, e um dos organizadores da famosa Conferência de Dartmouth, que em 1956 reuniu pela primeira vez cientistas da IA. O otimismo daqueles dias de 1956 com relação as possibilidades da IA pode ser sentido pelo pensamento que norteava o propósito do Dartmouth College, consolidado pela Fundação Rockfeller: *"A intenção é prosseguir sobre a base da conjectura de que todo aspecto da aprendizagem ou qualquer outra caracteristica da inteligência pode em principio ser tão precisamente descrita que uma máquina pode ser construida para simulá-los"* **[FIR 89]**.

Assim, sob os ambiciosos objetivos dos pioneiros, a IA consolidava-se como um promissor campo da pesquisa cientifica, e pela própria natureza e propósito, atraia e incorporava massivamente as áreas acadêmicas da filosofia, lingüistica, psicologia, matemática, fisica, engenharia elétrica, e ciência da computação, sem poder, no entanto, ser considerada como um ramo de qualquer um desses campos.

2.2 Resolução de Problemas na IA

Quando o termo "inteligência artificial" foi criado, já existia nos dominios da ciência recém nomeada o estudo a respeito de como resolver *problemas* num ambiente tipicamente caracterizado por máquinas computadoras. Aliás, não é estranho que no escopo da ciência recém criada tenha sido encontrado o envolvimento com a questão *problemas*, pois como é lembrado em **[LOP 81]**, num ambiente voltado para a investigação cientifica os problemas se constituem como os elementos básicos.

Uma das mais significativas contribuições neste periodo foi o *Logic Theorist* desenvolvido em 1956 por A. Newell, H. Simon e J. C. Shaw. O Logic Theorist conseguiu provar 38 dos 52 teoremas do *Principia Mathematica* de Whitehead e Russell. A prova do Teorema 2.85 (do Principia Mathematica) foi apresentada, de fato, numa forma mais concisa e elegante do que aquela apresentada nos Principia

Mathematica. Animados com o sucesso do Logic Theorist, e cientes de sua limitação, no ano seguinte, os três pesquisadores da RAND Corporation iniciaram o desenvolvimento do *General Problem Solver (GPS)*. Os objetivos dos autores do GPS eram basicamente dois:

a) Desenvolver um explícito e operacional paradigma (modelo) para a forma de como os seres humanos resolvem problemas; e b) Implementar este modelo num computador.

Como objetivo não declarado desta abordagem havia a tentativa de tentar separar os métodos gerais de resolver problemas dos dados específicos da tarefa que caracterizam o problema **[FIR 89]**. De certa forma, através desse método, os autores tencionavam abstrair algumas "heurísticas".

Os esforços da Inteligência Artificial na resolução de problemas não tem levado em consideração que um problema seja tratado como uma estrutura sintática passível de manipulação mecânica, isto é, não se identifica um problema com uma estrutura padronizada capaz de conter dentro de si elementos suficientes para descrever o problema e ainda prover informações necessárias para a solução do mesmo. Desse modo, o estudo de resolução de problemas tem se constituído em estudos de casos, ou seja, tais problemas com tais e tais características seriam resolvidos de uma maneira específica ou usando-se determinadas heurísticas identificadas com tal classe de problemas. Ainda hoje convivemos com tal tipo de abordagem em relação ao estudo de problemas dentro da IA.

A preocupação maior dos que abordam o tema tem sido as técnicas de como resolver os problemas e não o estudo do objeto problema em si. Em **[LOP 81]** encontramos a citação sobre a obra de N. J. Nilsson ([Nilsson 71]), *Problem Solving Methods in Artificial Intelligence*, que embora se constitua "num importante esforço auxiliar para uma abordagem mais ampla da questão *problemas*", trata de problemas "com o objetivo específico de estudar problemas computáveis e as especificações das estruturas de dados envolvidas em suas resoluções".

Em **[BAN 80]**, que se constitui uma versão estendida e modificada da obra do mesmo autor *Theory of Problem Solving: An Approach to Artificial Intelligence* **[BAN 69]**, encontramos uma abordagem nitidamente teórica sobre o que se constitui *problemas*. No trabalho de Banerji já se apresenta a preocupação com a necessidade de uma linguagem de discurso independente do dominio do problema que seja ainda suficientemente conveniente e precisa, de modo que se possa escrever algoritmos de resolução de problemas que sejam independentes dos problemas. Com isso o autor justifica a escolha da linguagem da matemática discreta.

Segundo Banerji, os problemas da vida real, na maioria das vezes, são percebidos de uma maneira tal que são dificeis de expressar de um modo aceitável para um computador. Isso se dá devido a natureza não-numérica de tais problemas, e por conseguinte as técnicas matemáticas que dependem dos números não podem expressar precisamente tais problemas. Assim o autor dirige a tratativa de problemas para "games e puzzles" "por causa de sua natureza bem formada", daí a observação em **[LOP 81]** de que a abordagem de Banerji esta "ainda presa ao enfoque *artificialista* da teoria dos jogos".

Ainda segundo Banerji, como os jogos e charadas têm também uma caracteristica não numérica, a esperança é que se pudesse transferir o entendimento na solução de tais problemas para os problemas da vida real de mesma natureza. Ele acrescenta que infelizmente a ponte pretendida não tem sido edificada, já que o entendimento abstraido dessa abordagem tem sido aplicado apenas de um jogo para outro. O problema reside no fato de que tem havido muito pouco entendimento da relação entre a estrutura de um problema e a estrutura dos bem sucedidos métodos de solução para o problema.

Banerji assevera que seu trabalho de pesquisa baseia-se, não em resolver problemas, mas principalmente em entender as técnicas de resolução de problemas e suas caracteristicas. De acordo com suas próprias palavras, "isso tem sido possivel... porque a pesquisa foi dirigida para o estudo de estruturas".

A estrutura *problema* proposta em [BAN 80] é um terno ordenado $\mathcal{P}$ = <S, W, R>, onde S é o conjunto de todos os estados possíveis, W é o conjunto dos possíveis estados finais, e R é uma relação que indica quais estados podem ser derivados de outros estados.

Como se pode perceber, tal estrutura tem alcance bastante limitado na intenção de representar *problema*, pois a definição de problema proposta por Banerji na realidade define *espaço de estados* [RIC 94], e ainda continua presa ao tradicional método da IA denominado "*state space approach*", que por sua vez se constitue um viés da teoria heurística da busca [SIL 85].

Em [KOW 79] encontramos uma outra abordagem da questão resolução de problemas. Nesta obra, o autor justifica o uso da lógica de primeira ordem, mais especificamente as sentenças na forma clausal (cláusulas de Horn), para a representação e manipulação de conhecimento relativo a resolução de problemas. Kowalski argumenta que a inferência lógica provê um modelo que é tanto mais simples e mais poderoso, já que a lógica é uma importante ferramenta na análise e apresentação de argumentos.

Ainda segundo Kowalski, a resolução de problemas pode ser classificada em três estágios principais:

a) O 1º estágio identifica o domínio do problema e formula os procedimentos para a resolução de problemas;

b) O 2º estágio aplica os procedimentos para a solução de problemas;

c) O 3º estágio melhora as estratégias de resolução de problemas e procedimentos do problema.

Nas próprias palavras de Kowalski, a abordagem centraliza-se no segundo estágio pois os outros dois estão envolvidos com questões de aprendizagem. Segundo ele, *McCarthy [1968]* e *Minsky [1968]* já haviam alertado sobre a necessidade de explorar a adequação da linguagem de representação antes de tratar com as questões de formular e melhorar a representação do domínio do problema.

Seja como for, este trabalho de Kowalski [**KOW 79**] além de se deter na questão de tratar essencialmente "dos procedimentos para a solução de problemas", levanta também o aspecto da adequação da linguagem já apontada por Banerji uma década antes dele, como também esbarra na questão de não avançar em aspectos que envolvam a aprendizagem.

Embora Kowalski não se detenha para especificar quais dos aspectos da formalização de *problema* são os responsáveis pelas dificuldades envolvendo a aprendizagem (e, na realidade não é esse o objetivo do seu estudo em questão), sabemos com certeza que a invariância da forma do objeto que está sendo manipulado contribui na facilitação ou não do processo de aquisição de conhecimento dos aspectos relativos ao manuseio de tal objeto.

O ideal seria que um ambiente de resolução de problemas proporcionasse uma abordagem completa sobre tudo que se relacionasse ao problema. Mas é de se supor que qualquer abordagem na resolução de problemas que não contemple uma maneira devida de representação do problema em si como um objeto manipulável apresente dificuldades operacionais em questões que envolvam a aprendizagem. Uma coisa é a manipulação de um objeto que tem uma forma padronizada, e outra é a manipulação de um objeto que sempre se apresenta amorfo.

O princípio da manipulação de objetos com forma invariável é aplicado com sucesso em qualquer processo mecânico. Como a forma é sempre a mesma aprende-se a manusear as peças com mais rapidez e destreza, e não é estranho que se encontre no ambiente de processo operadores que descobriram sua maneira própria de montar as peças utilizando um algoritmo mais eficiente do que o sugerido pela gerência do processo. Os programas de engenharia de produção e qualidade exploram exatamente este viés da questão.

A manipulação dos problemas numa estrutura padronizada, independente da forma como se apresentam, é semelhante a um recipiente utilizado para carregar objetos. Não interessa a forma do objeto colocado dentro do recipiente, a maneira de conduzi-lo será sempre a mesma. Pode ser que sobre espaços no recipiente que o objeto não conseguiu ocupar devido a singularidade de sua forma; mas vale observar que o eficiente acesso ao objeto dentro do recipiente, ou mesmo a eficiente manipulação do recipiente com o objeto, são permitidos envolvendo apenas uma quantidade limitada de graus de liberdade. Tais fatores tornam atraentes a *mecanização* do processo.

Ainda com relação aos três estágios, envolvendo a resolução de problemas, mencionados em **[KOW 79]**, a estruturação de *problema* proposta pela TGP **[LOP 81]** contribui significativamente para a abordagem, não apenas do 2º estágio, mas também do 1º e do 3º. Do 1º pois os domínios são claramente definidos na estrutura , e do 3º porque a TGP contempla a quebra do problema em subproblemas através de critérios bem definidos de redução e decomposição de problemas.

Embora tenhamos explorado o envolvimento dos cientistas da IA, vale salientar, entretanto, que alguns estudiosos da questão *problemas*, especificamente os matemáticos, já ensaiavam, lá pelo início dos anos 30, uma visão essencialmente sintática de problema. Em 1932, num artigo intitulado "Zur Deutung der Intuitionistischen Logik", Kolmogoroff apresenta um "cálculo para solução de problemas" como interpretação para um cálculo proposicional intuicionista **[LOP 81]** e **[LOP 85]**. Em [Suppes, 69] **[LOP 81]** o autor formula um modelo teórico para "decisões individuais" baseado na sugestão de um terno ordenado, semelhante ao que encontramos em **[LOP 81]** na definição de pré-problema, para representar uma "decisão individual".

O objetivo desta seção não é esgotar a lista dos cientistas da IA envolvidos com a resolução de problemas, nem tampouco traçar um histórico completo de todos que se envolveram com o tema. Como nosso enfoque é sempre do ponto de vista da TGP, o número de citações foram poucas porque poucos foram os que se debruçaram sobre a questão de formalizar matematicamente uma estrutura para *problema* (**[WID 64]**, [MAN 66],[BAN 69],[LOP 81]), pois a maioria se debruçou sobre a questão de formalizar a *resolução de problemas* ([NEW 59],[BEL 60],[MES 65],[QUL 68],[SAW 69] ,[NIL 71],[WIL 73]).

2.3 A TGP no contexto da IA

Pelos dados colhidos no levantamento histórico na parte introdutória da tese de doutoramento do Prof. M. A. Lopes, *Introdução a uma Teoria Geral de Problemas*, desde meados da década de 70 que alguns estudiosos brasileiros têm dedicado considerável atenção à questão *problemas*, contribuindo, assim, "efetivamente para a evolução de uma disciplina, que trata de problemas, desvinculada de um contexto específico". Citando ainda alguns trabalho dirigidos especificamente para problemas, [Veloso, 79] e [Veloso e Pequeno, 80], o Prof. Lopes observa que o Prof. Paulo Veloso "já canaliza suas atenções sobre a questão, com o objetivo de evidenciar propriedades de problemas independentemente dos ambientes em que eles estão inseridos", e os trabalhos conjuntos, [Veloso e Lopes, 79] e [Veloso e Lopes, 80], "foram dirigidos diretamente para essa abordagem" **[LOP 81]**.

Já tendo pleno conhecimento da abordagem adotada pela Inteligência Artificial em relação a resolução de problemas no tempo em que escreveu seu trabalho de doutoramento, o Prof. Lopes observa que o tipo de abordagem adotada não seguia a que se classificaria de "usual" dentro da IA, mas que a postura adotada "encarava um problema como algo abstrato e comum às situações que intuitivamente se chamam problemas, independentemente de suas naturezas e suas origens" **[LOP 81]**.

Finalmente o citado professor conclui em sua introdução que "uma Teoria Geral de Problemas não somente representa uma recompensa aos que trabalham em ciência e filosofia, como é uma contribuição efetiva no sentido de atender a um velho anseio de se conseguir indicações seguras para a resolução, crítica e análise de problemas" **[LOP 81]**.

Embora a abordagem da questão em **[LOP 81]** não tenha sido conduzida de acordo com o "costume da época", pode-se perceber nas intenções do autor, com relação a resolução de problemas, os mesmos interesses da IA e até outros mais além, os quais, se encampados pela Inteligência Artificial, só resultaria em ganho.

A propósito do que se mencionou no parágrafo anterior, Silva **[SIL 85]** em sua dissertação de mestrado compõe uma análise da teoria de Banerji **[BAN 69]** (baseada em estruturas de M-situações e W-problemas anteriormente apresentados por Marino **[MAN 66]** e Windeknecht **[WID 64]**) através da TGP, e conclui que a TGP mapeia completamente a teoria de Banerji. O nosso ponto de vista é que a TGP o faz com uma estrutura mais compacta e menos complexa para representar problema.

Avançando na questão, no sentindo de criar um liame mais forte entre a computação e a Teoria Geral de Problemas, os professores Lopes e Acioly, então desenvolvendo trabalhos no Depto. de Informática da UFPE, apresentaram num artigo, em 1985, uma introdução a *Uma Lógica de Resolução de Problemas* **[LOP 85]**.

Além dos trabalhos mencionados na seção 1.1(**[SAN 96]** e **[PRU 97]**), calcados sobre a TGP, ainda apontamos os trabalhos de Almeida, Bedregal, Oliveira Jr., e Silva (**[ALM 90]**, **[BED 87]**, **[OLI 85]**, **[SIL 85]**, respectivamente), calcados sobre a mesma base. Assim, a TGP, uma teoria de natureza essencialmente matemática, e os trabalhos derivados da mesma, ora tratando dos aspectos formais de resolução de problemas, ora tratando das possibilidades computacionais de problemas, se inserem no contexto de resolução de problemas da IA, enriquecendo-a.

Capítulo 3

TGP - Noções Básicas

Neste capitulo introduzimos as noções básicas da Teoria Geral de Problemas necessárias para a compreensão dos capitulos 4 em diante.

3.1 Da Intuição para a Formalização

No nossa vivência e convivência diária freqüentemente nos deparamos com alguma situação a qual coloca diante de nós algum tipo de dificuldade que, geralmente, requer uma *solução*. Quando nos encontramos em tal situação, automaticamente vem à nossa mente *que estamos com um problema*. Desse modo, vai se formando, intuitivamente, a idéia de *problema*. A formulação intuitiva da noção de *problema* é tão marcante que assumimos saber *o que é problema*. Ouvimos e dizemos com tanta freqüência "*Qual é o problema ?*" que dificilmente algum de nós já se perguntou "*O que é problema ?*". O aspecto pragmático de *problema* é tão mais dominante que sua formulação semântica quase que perde completamente a importância, e a respeito da existência de um possível aspecto sintático nem sequer cogitamos. É exatamente sobre o "esquecido" aspecto sintático da questão que a TGP presta sua principal contribuição. A estruturação de um *problema* sai do limbo para a pristina representação algébrica, abrindo, assim, uma gama de possibilidades para a sua manipulação matemática e computacional.

Problema-financeiro, problema-de-saúde, problema-de-matemática, etc., todos se apresentam diante de nós com algumas características em comum que nos trazem indicações seguras de que estamos diante de um *problema*, "muito embora seja extremamente difícil se atribuir a um ou mais elementos a qualidade responsável pela identificação" de *problema*. "Entretanto, a abstração *problema* pode ser feita através da captação de alguns elementos estruturais, invariantes e relevantes, que ocorrem no que identificamos intuitivamente como sendo um *problema*, ou seja, o "Tipo Problema" **[LOP 81]**.

A "tipagem" de *problema* proposta pela TGP abrange conceitos da Álgebra Universal e da Teoria dos Modelos, e a estrutura lógico-matemática resultante, apontando para o *tipo problema*, cria um liame com a Computação ao herdar aspectos de Tipos Abstratos de Dados, quais sejam: domínios de objetos munidos de operações e relações.

3.2 Problema e Solução de Problema

Como será observado na seção 6.4, a noção de satisfação na Lógica de Resolução de Problemas envolve o conceito de satisfação de um esquema de solução **a** nos domínios do problema **P**. A definição de problema e solução que será apresentada a seguir tem por finalidade introduzir o leitor no entendimento da estrutura usada na TGP para representar um *problema*. Os elementos componentes de tal estrutura são os domínios do problema, além dos predicados envolvendo os elementos do domínio e a possível solução. Todas as noções apresentadas nesta seção se acham originalmente explanadas em **[LOP 81]** e **[LOP 85]**.

Definição 3.2.1 - Um problema é uma estrutura do tipo $\mathcal{P} = \langle \mathbf{D}, \mathbf{R}, \mathbf{d}, \mathbf{r}, \mathbf{A} \rangle$, onde **D** e **R** são conjuntos não-vazios, denominados respectivamente *domínio de dados* e *domínio de resultados*, $\mathbf{d} \subseteq \mathbf{D}$ é um predicado unário sobre **D**, $\mathbf{r} \subseteq \mathbf{D} \times \mathbf{R}$, um predicado binário sobre $\mathbf{D} \times \mathbf{R}$, e $\mathbf{A} \subseteq \mathbf{R}^{\mathbf{D}}$ um predicado unário sobre $\mathbf{R}^{\mathbf{D}}$.

Os conjuntos não-vazios **D** e **R** representam a *parte substantiva* do problema $\mathcal{P}$, e **d**, **r** e **A** representam a *parte intencional* e são denominados *predicado fonte*, *predicado intencional* e *predicado admissibilidade*, respectivamente. O objetivo é que **r** reflita a *condição principal* ou *intencionalidade principal*, segundo a qual os pares (f,s), $f \in \mathbf{D}$ e $s \in \mathbf{R}$, são formados, isto é, $(f,s) \in \mathbf{r}$; **d**, uma *restrição sobre* **D**, reflita a parte de **D** relevante ao problema, segundo **r**; e **A** seja a restrição às possíveis funções de **D** em **R** que satisfazem **r**, e que não serão consideradas como admissíveis.

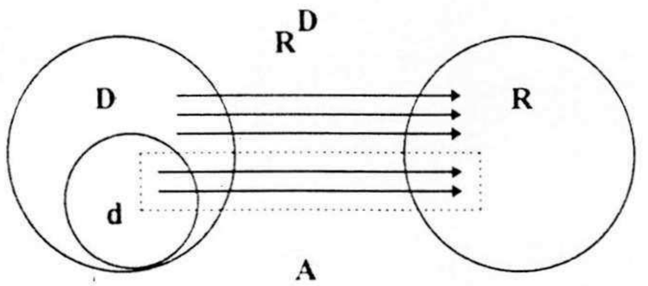

Fig. 3.2.1 - Visão esquemática da estrutura *problema*.

De modo a fixar o significado e função de cada elemento da estrutura *problema*, apresentamos abaixo alguns exemplos.

Exemplos 3.2.1 - Problemas:

a) Seja $\mathcal{P}$ o seguinte problema: "Se α é um teorema de uma teoria $\mathcal{T}$, demonstre-o, caso contrário, apresente um contra-exemplo para α".

Uma estruturação para $\mathcal{P}$ poderia ser a seguinte:

$$\mathcal{P} = \langle \textbf{Form}, \textbf{Demo}, \textbf{Sent}, \textbf{Pred}, \textbf{A} \rangle$$

onde:

Form - conjunto das fórmulas da linguagem $\mathcal{L}$ de $\mathcal{T}$

Demo - conjunto de demonstrações $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ para sentenças de $\mathcal{T}$ unido com o conjunto dos modelos de $\mathcal{T}$

Sent - conjunto das sentenças de $\mathcal{L}$

$(\alpha, x) \in$ **Pred** $\Leftrightarrow$ x é uma demonstração para α ou x é um modelo para $\neg\alpha$

A - livre de restrições.

b) Seja $\mathcal{P}$ um problema envolvendo funções reais e suas respectivas aproximações polinomiais, descrito como segue: "Seja f uma função contínua definida no intervalo [a,b] da reta. Determine uma aproximação polinomial p cujo erro em qualquer ponto do intervalo [a,b] não exceda um certo ε dado".

$\mathcal{P}$ pode ser representado pela seguinte estrutura:

$$\mathcal{P} = \langle \textbf{Func}, \textbf{Pol}, \textbf{Fcont}, \textbf{Pred}, \textbf{Met} \rangle$$

onde:

Func - conjunto de funções reais definidas no intervalo [a,b]

Pol - conjunto e polinômios de grau n para $n \geq 0$.

Fcont - conjunto de funções reais contínuas no intervalo [a,b]

$(f(x), p(x)) \in$ **Pred** $\Leftrightarrow |f(x) - p(x)| \leq \varepsilon$ para todo $a \leq x \leq b$

Met - conjuntos de métodos que para cada função definida em [a,b] associa um polinômio de grau n.

c) Considere $\mathcal{P}$ o seguinte problema: "Escreva algoritmos escritos em FORTRAN que efetuem a multiplicação de pares de inteiros"

$\mathcal{P}$ pode ser estruturado do seguinte modo:

$$\mathcal{P} = \langle \mathbf{Z}^2, \mathbf{Alg}, \mathbf{Z}^2, \mathbf{Pred}, \mathbf{FOR} \rangle$$

onde:

$\mathbf{Z}^2$ - conjunto de pares de inteiros

Alg - conjunto de algoritmos capazes de efetuar a multiplicação de inteiros

$((a,b), a) \in$ **Pred** $\Leftrightarrow$ para todo $(a,b) \in \mathbf{Z}^2$, o algoritmo a produz na saída um número z que é o produto de a vezes b ($z = a \bullet b$)

FOR - conjunto de algoritmos escritos em FORTRAN

A partir do entendimento dos papéis desempenhados pelos domínios e predicados intencionais dentro da estrutura problema, podemos então introduzir o entendimento do conceito de solução de problema.

Definição 3.2.2 - Seja $\mathcal{P} = \langle \mathbf{D}, \mathbf{R}, \mathbf{d}, \mathbf{r}, \mathbf{A} \rangle$ um problema. Uma *solução* para $\mathcal{P}$ é uma função a definida de **D** em **R**, tal que $a \in \mathbf{A}$ e para todo $f \in \mathbf{d}$, $(f, a(f)) \in \mathbf{r}$; ou ainda, $\mathcal{P}$ é solúvel se e somente se existe um a cumprindo tais condições. Indica-se que "a resolve $\mathcal{P}$" ou que "a é solução para $\mathcal{P}$" pelo símbolo $a \mid\mid\!- \mathcal{P}$.

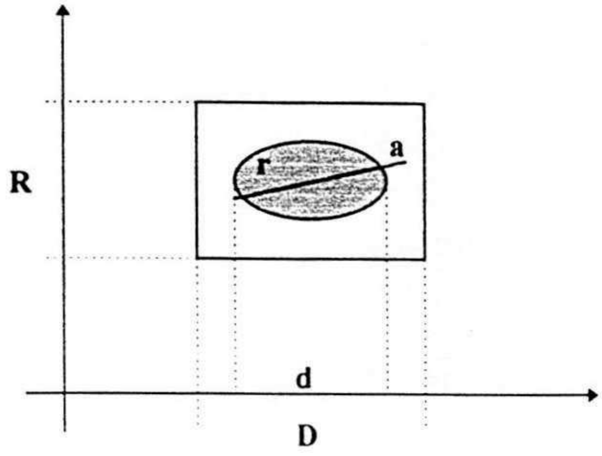

Fig. 3.2.2 - Visão esquemática de *solução de problema*.

Assim, compreende-se mais facilmente porque uma solução a para um problema $\mathcal{P}$ é necessariamente um elemento de **A**, e não de **R**. Desse modo, um elemento de **R** que satisfaz a condição imposta por **r** mas que não é selecionável por nenhum $a \in \mathbf{A}$ não serve como resultado em $\mathcal{P}$.

Aqui vale introduzir um breve parênteses a respeito da diferença entre *resultado* e *solução*, levando em conta que para o bom entendimento dos conceitos de problema e solução de problema da TGP, *resultado* e *solução* não devem ser confundidos.

Resultado é um elemento de **R**, e de acordo com a Definição 3.2.2, *solução* é um elemento de $\mathbf{R}^{\mathbf{D}}$. Apenas pelos domínios já se tem uma idéia da diferença de natureza que envolve os dois termos. *Resultado* pode ser um número, um literal, um funcional, etc.., enquanto que *solução* é uma função. O *resultado* pode advir quando da aplicação da *solução* ao problema em questão. Quando existente, a *solução* é o método que ao ser aplicado ao problema conduz a um *resultado*. O *resultado* de um problema que pede a determinação das raizes de uma equação do segundo grau pode ser conseguido através da aplicação da *solução* conhecido como Método de Bhaskara.

3.3 Construção de Problemas

De acordo com **[LOP 81]** e **[BED 87]** podemos, a partir de problemas já existentes, construir outros através de soma, união, composição e combinação de problemas. Em **[LOP 85]** são apresentados três novos tipos de problemas que serão exaustivamente utilizados nas sentenças da LRP.

Definição 3.3.1 - A partir dos problemas $\mathcal{P} = \langle D_1, R_1, d_1, r_1, A_1 \rangle$ e $\mathcal{Q} = \langle D_2, R_2, d_2, r_2, A_2 \rangle$ podemos construir os seguintes novos problemas:

$\mathcal{P} \wedge \mathcal{Q} = \langle D_1 \times D_2, R_1 \times R_2, d_1 \times d_2, r_1 \times r_2, A_1 \times A_2 \rangle$, onde "$\times$" indica o produto cartesiano, e lemos "problema $\mathcal{P}$ e $\mathcal{Q}$".

$\mathcal{P} \vee \mathcal{Q} = \langle D_1 \times D_2, R_1 \times R_2, d_1 \times d_2, r_1 \cup r_2, A_1 \times A_2 \rangle$, onde $r_1 \cup r_2 = \{(x,y) / x \in r_1$ ou $y \in r_2\}$ indica a união cartesiana). $\mathcal{P} \vee \mathcal{Q}$ será lido "problema $\mathcal{P}$ ou $\mathcal{Q}$".

$\mathcal{P} \rightarrow \mathcal{Q} = \langle D_1 \times D_2, R_1 \times R_2, d_1 \times d_2, \overline{r}_1 \cup r_2, A_1 \times A_2 \rangle$, onde $\overline{r}_1$ é o complemento de r_1 em relação a $D_1 \times R_1$. O problema $\mathcal{P} \rightarrow \mathcal{Q}$ será lido "$\mathcal{P}$ implica $\mathcal{Q}$".

Como se pode observar, os novos problemas criados ($\mathcal{P} \wedge \mathcal{Q}$, $\mathcal{P} \vee \mathcal{Q}$ e $\mathcal{P} \rightarrow \mathcal{Q}$), têm a forma e a leitura de expressões proposicionais na forma clássica. Na realidade, tal forma foi buscada de modo intencional, pois é objetivo da LRP tratar de expressões envolvendo as entidades problemas ligadas por conectivos do tipo "e", "ou" e "implica" como vistos acima.

Um outro fato importante que será explorado pela LRP é a ligação entre soluções e problemas. A intenção é tratar da possibilidade de que certos esquemas funcionais venham a ser ou não soluções para determinados problemas. Devido aos entraves epistemológicos que serão estudados no capítulo 4, a LRP será apresentada como uma junção de duas lógicas - uma de esquemas de solução e outra de esquemas de problemas - a priori independentes. Será preciso ter uma idéia do tratamento que deve ser dado aos esquemas de solução, dissociados dos esquemas de problemas, e será também necessário ter uma visão do tratamento de esquemas de problemas como entidades dissociadas e independentes de quaisquer esquemas de solução. Na realidade, tais tratamentos são contemplados pelas respectivas lógicas de esquemas de solução e esquemas de problemas.

O teorema seguinte será a base para o estabelecimento da ligação biunívoca (associação e dissociação) que deverá ser estabelecida na relação existente entre uma expressão envolvendo esquemas de solução e outra envolvendo esquemas de problemas. De modo particular, a associação entre esquemas de solução e esquemas de problemas será futuramente explorada de maneira especial, pois será sobre tal relação que se estabelecerá um dos pontos fortes da LRP - uma função de atribuição de verdade que capta o grau de envolvimento entre um esquema de solução e um esquema de problema.

Teorema 3.3.1 - (Teorema 2.3 de **[LOP 85]**) Sejam a, a_1 e a_2 soluções, e $\mathcal{P}$ e $\boldsymbol{Q}$ problemas. Então,

(i) $a = (a_1, a_2) \Vdash \mathcal{P} \wedge \boldsymbol{Q} \Leftrightarrow a_1 \Vdash \mathcal{P}$ e $a_2 \Vdash \boldsymbol{Q}$.

(ii) $a = (a_1, a_2) \Vdash \mathcal{P} \vee \boldsymbol{Q} \Leftrightarrow a_1 \Vdash \mathcal{P}$ ou $a_2 \Vdash \boldsymbol{Q}$.

(iii) $a = (a_1, a_2) \Vdash \mathcal{P} \rightarrow \boldsymbol{Q} \Leftrightarrow a_1 \nVdash \mathcal{P}$ ou $a_2 \Vdash \boldsymbol{Q}$.

Demonstração: (i) $(\Rightarrow)$ $a = (a_1, a_2) \Vdash \mathcal{P} \wedge \mathcal{Q}$. Pela Definição 3.3.1, temos

$\mathcal{P} \wedge \mathcal{Q} = \langle \mathbf{D_1} \times \mathbf{D_2}, \mathbf{R_1} \times \mathbf{R_2}, \mathbf{d_1} \times \mathbf{d_2}, \mathbf{r_1} \times \mathbf{r_2}, \mathbf{A_1} \times \mathbf{A_2} \rangle$, onde

$\mathbf{r_1} \times \mathbf{r_2} = \{(x,y)/\ x \in \mathbf{r_1} \text{ e } y \in \mathbf{r_2}\} \Rightarrow \forall (f_1, f_2) \in \mathbf{d_1} \times \mathbf{d_2}$, daí

$\mathbf{r_1} \times \mathbf{r_2} = ((f_1, a_1(f_1)), (f_2, a_2(f_2))) \Rightarrow a_1 \Vdash \mathcal{P}$ e $a_2 \Vdash \mathcal{Q}$.

$(\Leftarrow)$ $a_1 \Vdash \mathcal{P}$ e $a_2 \Vdash \mathcal{Q} \Rightarrow \forall f_1 \in \mathbf{d_1}, (f_1, a_1(f_1)) \in \mathbf{r_1}$ e $\forall f_2 \in \mathbf{d_2}, (f_2, a_2(f_2)) \in \mathbf{r_2}$.

$\forall (f_1, f_2) \in \mathbf{d_1} \times \mathbf{d_2} \Rightarrow ((f_1, f_2), a_1 \times a_2(f_1, f_2)) \in \mathbf{r_1} \times \mathbf{r_2}$

$\Rightarrow \forall f \in \mathbf{d_1} \times \mathbf{d_2}\ (f, a(f)) \in \mathbf{r_1} \times \mathbf{r_2} \Rightarrow a = (a_1, a_2) \Vdash \mathcal{P} \wedge \mathcal{Q}$.

Para (ii) e (iii) o raciocínio é análogo.

O conteúdo do Teorema 3.3.1 será plenamente utilizado na seção 6.5 quando do estabelecimento de equivalências entre expressões da LRP.

Capítulo 4

A Questão da Negação na LRP

Como este trabalho trata do desenvolvimento de uma lógica, achamos por bem incluir alguns poucos comentários que discorresse, mesmo que superficialmente, alguma coisa a respeito da filosofia da Lógica.

Gostaríamos de salientar que o tema mais importante deste capítulo é a parte que trata da questão da negação na lógica em geral, e sua particular extensão para a LRP. Embora tenhamos vasculhado em mais de três dezenas de textos, menos de um terço se preocupava em discorrer sobre a filosofia da Lógica, e, uma quantidade menor ainda em tratar sobre os aspectos da negação.

4.1 Lógica

A lógica é tida como a "ciência das formas de pensamento" **[COP 72] [LIA 68]**, ou a ciência do raciocínio **[COP 72]**, e pelo menos dois principais objetivos são perseguidos: estabelecer as leis do pensamento e determinar as diferentes aplicações das mesmas **[LIA 68]**. E o estudo da lógica é o estudo dos métodos e princípios usados para distinguir o raciocínio correto do incorreto **[COP 72]**. Assim, a lógica estuda a razão como instrumento da ciência, ou o meio de adquirir e possuir a *verdade* **[MAR 62]**.

Por meio da intuição ou apreensão de fatos que *estão em nós* ou *fora de nós*, nossa mente vai captando o seu material de trabalho. A função da lógica começa quando por meio das operações e mecanismos mentais, formamos a *associação* desses materiais que nos são fornecidos. Tais associações são, basicamente, de três espécies: *as noções*, *os juizos*, e *as inferências*; e para expressá-las fazemos uso da *linguagem*. As noções são expressas por meio dos *termos* (ou *nomes*), os juizos pelas *proposições* (ou *sentenças*), e as inferências pelos *raciocínios*. **[LIA 68] [MAR 62]**

As noções não passam de materiais do pensamento; ainda não há pensamento quando apenas nos limitamos a considerar idéias isoladas umas das outras, sem ligações entre si. O pensamento começa com a afirmação, pois, é através dela que se dá o efeito da união das idéias que se apresentam separadamente à mente. A afirmação é o próprio ato do juizo; e a proposição é o *enunciado* do juizo. **[LIA 68]**

Os termos, ou nomes, são os limites da proposição **[MAR 62]**, isto é, o sujeito com o qual ela começa, e o predicado pelo qual termina. Assim, toda proposição se compõe de dois termos: o sujeito, aquele de quem se afirma; o atributo ou predicado, aquilo que é afirmado; e uma cópula, o verbo _é_, que une o sujeito e o predicado. Mas a lógica vê no verbo unicamente a ligação do sujeito e do predicado, salientando-se que *todo verbo* pode ser *reduzido* ao verbo *ser*.

Quando as proposições bem construídas são associadas coerentemente umas às outras, conclusões podem ser estabelecidas. A partir dessa junção, os mecanismos da mente conseguem abstrair idéias que antes não apareciam em cada proposição isolada. Dizemos, então, que há inferência.

A lógica trata, portanto, de argumentos e inferências. Um de seus propósitos básicos é apresentar métodos capazes de identificar argumentos logicamente *válidos*, distinguindo-os dos que não são logicamente válidos. **[SAL 69]**

4.2 A Questão da Negação

As idéias que serão apresentadas nas seções 4.2.1 e 4.2.2 seguintes são basilares para a concepção das lógicas de esquemas de problema e de esquemas de solução, e conseqüentemente, da lógica de resolução de problema.

4.2.1 Negação de Termos e Proposições

"Todo termo pressupõe a idéia por ele expressa" **[LIA 68]**. Os termos podem ser *positivos* ou *negativos*. Quando significam a presença de uma determinada qualidade, são positivos; negativos, se significarem a ausência dessa mesma qualidade. Todo termo positivo é acompanhado no pensamento, e geralmente também no vocabulário, de um termo negativo correspondente.

Do ponto de vista da lógica sentencial clássica, podemos afirmar que quando dois *termos opostos* admitem uma *terceira alternativa* é porque está havendo confusão entre a *presença* ou *ausência* da qualidade, com o *grau* dessa mesma qualidade. Quando se trata da presença ou ausência de uma qualidade *não existe* meio termo.

A questão da *negação do termo* está relacionada com a *natureza* do conceito que ele representa. Esta idéia pode ser entendida da seguinte maneira: Os termos *homem*, *Maria* e *casa*, designando um ser, uma pessoa e um objeto, respectivamente, não admitem um termo negativo correspondente, pois, a idéia de *não-homem*, *não-Maria* e *não-casa* não está associada a qualquer coisa identificada individualmente. Por outro lado, os termos *bem* e *mau*, designando qualidade, admitem um termo negativo correspondente, pois, a idéia de *não-bem* e *não-mau* encontra plena identidade nos termos *mal* e *bom*, respectivamente.

"A qualidade de uma proposição é o seu caráter afirmativo ou negativo" **[LIA 68]**. Negação e afirmação não diferem em essência. Percebe-se sem esforço que negar que certo atributo é conveniente a um sujeito é afirmar que não lhe convém. Pela afirmação, com efeito, a idéia é declarar que certo predicado está incluído em dado sujeito; enquanto que pela negação, ao contrário, a idéia é declarar que certo atributo não se acha contido em determinado sujeito.

Quando se trata da inclusão do sujeito na *extensão* do predicado, toda negação se reduz a uma afirmação, e nesse caso o predicado é qualificador do sujeito. Quando se trata da inclusão do predicado para a *compreensão* do sujeito, *não se pode mais* sustentar a *identidade lógica* entre a negação e a afirmação. Nesse caso o predicado ajuda a definir o que significa o sujeito.

Podemos ilustrar a idéia acima através de exemplos. Suponha que tenhamos a seguinte negação: "O homem NÃO é quadrúpede". Tal sentença é claramente entendida como afirmando que o homem não pertence à classe dos quadrúpedes, o que pode muito bem ser declarado da seguinte maneira: "O homem É não quadrúpede". A mesma "identidade lógica" entre negação e afirmação não pode ser mantida na seguinte expressão: "Bola NÃO é um corpo quadrado". Quando dizemos que "Bola É não um corpo quadrado", afirmamos algo que não sabemos o que significa. Essa dificuldade se estabelece porque no primeiro caso o sujeito está recebendo uma *qualificação*, enquanto que no segundo ele está recebendo uma *definição*.

4.2.2 A Negação na LRP

Lopes e Acioly **[LOP 85]** observam que da maneira como o objeto *problema* é apresentado em **[LOP 81]**, isto é, como uma estrutura relacional, "não é possível encontrar o similar à interpretação de Kolmogoroff para $\neg\mathcal{P}$". Como para Kolmogoroff a idéia de problema é pragmática, "sem estruturação prévia", uma questão do tipo "dada uma solução para o problema $\mathcal{P}$, obtenha uma contradição" é plenamente aceita na interpretação do seu cálculo proposicional intuicionista.

Em **[LOP 81]** a negação de um problema, "do ponto de vista de sua solubilidade, não é definível univocamente, *uma vez que envolve simultaneamente a noção de intencionalidade e admissibilidade*"[1]. Assim, a sintaxe da lógica intuicionista de Kolmogoroff não se presta para "uma interpretação de problemas, no sentido de Lopes... porque a interpretação da negação está prejudicada" **[LOP 85]**.

Para Lopes e Acioly "é natural que *a idéia de negação* de um problema seja obtida através da *negação de uma intenção*"[2]. Através dessa idéia resolve-se um sério problema, qual seja, o livrar-se da negação da intenção do problema na descrição, passando-se a negá-la na estrutura do problema. Esse salto epistemológico resolve questões sem solução do tipo, por exemplo, um problema cuja intenção é descrita pelo termo "determine" tem a negação da descrição da intenção como sendo "não-determine"; nesses termos não é possível estruturar o novo problema. Enquanto que, se a negação for feita a nível de estrutura, estaremos, na realidade, diante de um novo e distinto problema.

Sendo um problema $\mathcal{P}$ = <**D**, **R**, **d**, **r**, **A**>, sua negação se daria negando-se **r**. Convencionando-se que a negação de **r** é $\bar{\mathbf{r}}$, a negação de $\mathcal{P}$ seria o problema $\neg\mathcal{P}$ = <**D**, **R**, **d**, $\bar{\mathbf{r}}$, **A**>. Nesse caso, **r** e $\bar{\mathbf{r}}$ são disjuntas, isto é, independentes.

Em **[LOP 85]** um problema proposto a título de exemplo tem sua negação abordada segundo duas análises distintas. A conclusão a que os autores chegam é

---

[1] Grifo acrescentado

[2] Grifo acrescentado

bastante interessante do ponto de vista epistemológico, e mostra a complexidade envolvida com a idéia de negação de problema.

O problema proposto é o seguinte: "Dada uma seqüência de números naturais sem repetição, ordene-a". Claramente, a intenção que transparece no problema é a de "ordenação". A princípio, a negação dessa intenção seria não ordene ou desordene. Como a negação lógica consiste em acrescentar o conectivo NÃO antes do que se pretende negar, a alternativa lógica seria não ordene. Mas, como defendem Lopes e Acioly, o termo não ordene não caracteriza em si uma ação, 'uma vez que "não aja" não tem valor de procedimento'. Assim, a negação da intenção de ordenação seria desordene, mesmo fugindo da idéia a priori da negação lógica.

A escolha da negação apropriada, como no caso acima, pode conduzir a alguns incovenientes técnicos. No caso do problema acima, Lopes e Acioly sugerem, por exemplo, que descrevamos ordene como a conjunção de dois predicados: *mesma* e *boa*, onde $mesma(s,s')$ significa que as seqüências s e s' possuem os mesmos elementos, e $boa(s')$ significa que s' está ordenada. Assim, significando que s' é a seqüência s ordenada, teremos $ordene(s,s') = mesma(s,s') \wedge boa(s')$. Por conseguinte, a negação lógica, $desordene(s,s')$, seria $\neg ordene(s,s') = \neg(mesma(s,s') \wedge boa(s'))$, que é equivalente a $\neg mesma(s,s') \vee \neg boa(s')$ [1], mas que não corresponde a idéia intuitiva de desordene. A fórmula predicativa capaz de refletir a noção intuitiva de desordene seria $mesma(s,s') \wedge \neg boa(s')$, não sendo, entretanto, equivalente a [1].

Esse contraste mostra que no esforço para se conseguir a correta noção intuitiva da negação lança-se mão de "considerações extralógicas", requerendo-se, portanto, "uma análise semântica a priori da estrutura sintática da intenção do problema" **[LOP 85]**. Tal abordagem constitui-se num sério impecilho para um sistema formal dedutivo, isto é, a manipulação de sentenças de um modo puramente mecânico.

Os resultados da análise a que chegaram os citados pesquisadores podem ser sumariados em dois pontos principais:

i) A negação de uma ação com valor de procedimento parece adquirir em si própria o *status* de uma ação. Independente da ação positiva, a negação dessa ação passa a ser também uma ação positiva. Desse modo, os problemas $\mathcal{P}$ e $\neg\mathcal{P}$ são problemas diferentes e independentes;
ii) A negação de um problema, do ponto de vista da intenção, não dá a essa negação o *status* de conectivo lógico;

Desse modo, concluem que um sistema formal de problemas parece servir adequadamente de interpretação para o cálculo proposicional positivo clássico.

No caso do sistema formal de esquemas de solução, embora a análise seja menos complicada, as considerações não são menos importantes. Os parâmetros da Lógica de Esquemas de Solução são esquemas *funcionais*, por conseguinte, "a negação de esquemas de solução não tem sentido lógico". Se o termo *função* identifica apropriadamente um ente matemático individual, ou mesmo uma classe, o mesmo não se pode dizer de *não-função*, que ao invés de identificar univocamente o conceito, na realidade dispersa-o. Assim, se a é um esquema de solução, a idéia intuitiva de $\neg a$ seria melhor interpretada como $-a$ ou $a^{\prime}$, "e isso tira a condição de conectivo lógico que gostaríamos de dar à negação ($\neg$)".

Assim, como observam os autores, "um sistema formal de esquemas de solução foi utilizado para interpretar também o cálculo proposicional positivo clássico" **[LOP 85]**.

Capítulo 5

Lógica de Resolução de Problemas

Neste capítulo serão apresentados os elementos e as idéias que definiram o desenvolvimento da parte introdutória da Lógica de Resolução de Problemas (LRP) de acordo com o que se acha exposto em **[LOP 85]**.

Nesse trabalho, os autores primeiramente desenvolvem uma Lógica de Esquemas de Problemas (LEP) e uma Lógica de Esquemas de Solução (LES). Nesses sistemas formais, esquemas de problemas e esquemas de solução são tratados como entidades sintaticamente distintas, sendo, por conseguinte, elementos de lógicas distintas. Em cada uma delas são desenvolvidos resultados cujos aspectos sintáticos são semelhantes aos da lógica sentencial clássica (**[COS 79]**, **[MOR 67]**, **[MAT 68]**, **[ROB 65]**), através de um sistema formal dedutivo baseado em axiomas e regras de inferência.

Posteriormente os dois sistemas formais, independentes e distintos, se juntam para formar o que os autores denominam de Lógica de Resolução de Problemas. A ligação entre os dois sistemas lógicos é feita através de uma noção de satisfação não-tarskiana, onde um esquema de solução $\mathbf{a}$ satisfaz um esquema de problema $\mathbf{P}$, se e somente se, a *realizado* em P implica em que a *resolve* P. A noção de realização proposta pelos autores é que o esquema de solução *é realizável* no problema, *se* o esquema de solução *pode ser interpretado como uma certa função do* domínio de dados *no* domínio de resultados desse problema.

Novamente é desenvolvido um sistema formal dedutivo para a Lógica de Resolução de Problemas baseado nos novos axiomas e regras de inferência deste novo sistema formal. Esse cálculo contempla a manipulação das sentenças dos esquemas de solução e de problemas, interligadas pela relação a "é esquema para" P, de modo que as novas deduções que vão aparecendo em ambos os lados da relação continuem sendo preservadas dentro da relação.

5.1 Lógica de Esquemas de Problemas (LEP)

A linguagem $\mathcal{L}_1$ com a qual vamos expressar os esquemas de problemas é baseada no seguinte alfabeto $\mathcal{A}_1$: Os *símbolos lógicos* serão compostos de parênteses, à esquerda e à direita, como *símbolos de pontuação*: (,); além dos *símbolos conectivos sentenciais*: $\wedge$, $\vee$ e $\rightarrow$, para expressar, respectivamente, o sentido de "e", "ou" e "implica" lógicos. Os *símbolos não-lógicos* ou *parâmetros* serão as letras maiúsculas A, B, C, ..., Y, Z.

Na linguagem $\mathcal{L}_1$ as fórmulas bem formadas (fbfs), chamadas de *esquemas de problemas*, são obtidas de acordo com as seguintes regras:

> **i) Os parâmetros A, B, C, ..., Y, Z, são esquemas de problemas;**
>
> **ii) Se A e B são esquemas de problemas, então (A $\wedge$ B), (A $\vee$ B), e (A $\rightarrow$ B) também o são;**
>
> **iii) Os esquemas de problemas são somente as sentenças obtidas de i e ii acima.**

A igualdade entre dois esquemas de problemas A e B será aqui introduzida por definição:

$$A = B \underset{def}{\equiv} (A \rightarrow B) \wedge (B \rightarrow A)$$

5.1.1 Sistema Formal Dedutivo na LEP

O sistema formal dedutivo da LEP será definido através do seguinte conjunto Λ de axiomas e uma regra de inferência:

A_1 - $A \rightarrow (B \rightarrow A)$

A_2 - $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$

A_3 - $(A \rightarrow (B \rightarrow A)) \rightarrow A$

A_4 - $A \rightarrow (A \vee B)$

A_5 - $B \rightarrow (A \vee B)$

A_6 - $(A \rightarrow C) \rightarrow ((B \rightarrow C) \rightarrow ((A \vee B) \rightarrow C))$

A_7 - $(A \wedge B) \rightarrow A$

A_8 - $(A \wedge B) \rightarrow B$

A_9 - $A \rightarrow (B \rightarrow (A \wedge B))$

R_1 - Se A e $A \rightarrow B$, então B. (*Modus Ponens*)

5.1.2 Prova Formal na LEP

Uma prova é um argumento que se dá a alguém e que o convence completamente da corretude da afirmação que requereu a prova. Deve ser finita, e deve possibilitar que qualquer outra pessoa seja capaz de checá-la para verificar que ela não contém quaisquer falácias, isto é, argumentos inválidos (**[END 72]**). Para o caso do sistema formal dedutivo da LEP que faz uso de apenas uma regra de inferência, cabem as seguintes definições:

Definição 5.1.1 - Seja A um esquema de problema e Γ um conjunto de esquemas de problemas. *Uma dedução (ou prova formal) de* A *a partir de* Γ é uma seqüência $<A_0, \ldots, A_n>$ de esquemas de problemas tal que $A_n = A$, e para cada $i \leq n$ ou

i) A_i está em $\Gamma \cup \Lambda$, ou

ii) Para algum j e k menor do que i, A_i é obtido de A_j e A_k através de modus ponens ($A_k = A_j \rightarrow A_i$)(**[END 72]**).

Aquelas fbfs da LEP cujas existências podem ser comprovadas através de um processo de dedução recebem a denominação especial de teoremas.

Definição 5.1.2 - Os teoremas de um conjunto Γ de esquemas de problemas são as fórmulas bem formadas obtidas de $\Gamma \cup \Lambda$ através do uso de modus ponens num número finito de vezes, isto é, *se existe uma dedução de* A *a partir de* Γ, *então* A *é um teorema de* Γ. Nesse caso, diz-se que A é *dedutível* de Γ e simbolizamos por $\Gamma \vdash A$ (**[END 72]**).

Assumindo as definições de *prova* e *teorema*, temos os seguintes resultados que são conseqüências diretas da aplicação dos axiomas de Λ e da regra de inferência R_1.

1 - Seja Γ um conjunto de esquemas de problemas e **A** e **B** problemas. Se $\Gamma; A \vdash B$, então $\Gamma \vdash A \rightarrow B$ (Teorema da Dedução). $[\Gamma; A = \Gamma \cup A]$

2 - $\mathbf{A \rightarrow B, B \rightarrow C \vdash A \rightarrow C}$

3 - $\vdash \mathbf{A \rightarrow (A \rightarrow B) \rightarrow B}$

4 - $\vdash (A \rightarrow (A \rightarrow B)) \rightarrow (A \rightarrow B)$

5 - $(A \rightarrow B) \rightarrow B, A \rightarrow C \vdash (C \rightarrow B) \rightarrow B$

6 - $\vdash A \vee (A \rightarrow B)$

7 - $A \vee B, A \rightarrow C, B \rightarrow D \vdash C \vee D$

8 - $((A_1 \vee \ldots\ldots \vee A_n) \vee B) = (A_1 \vee \ldots\ldots \vee A_n \vee B)$

9 - $A_1 \vee \ldots\ldots \vee A_n \vdash B_1 \vee \ldots\ldots \vee B_K$ se cada um $A_1, \ldots\ldots, A_n$ é um $B_1, \ldots\ldots, B_K$

10 - $A \vee B, C \rightarrow A \vdash (B \rightarrow C) \rightarrow A$

11 - $A \vee B, A \rightarrow B \vdash B$

12 - $\vdash (A \rightarrow (B \wedge C) = (A \rightarrow B) \wedge (A \rightarrow C)$

13 - $\vdash ((A \wedge B) \rightarrow C) = (A \rightarrow (B \rightarrow C))$

14 - $\vdash (C \vee (A \wedge B)) = (C \vee A) \wedge (C \vee B)$

5.2 Lógica de Esquemas de Solução (LES)

De modo semelhante faremos a introdução da linguagem $\mathcal{L}_2$ e do alfabeto $\mathcal{A}_2$ utilizados na LES.

No alfabeto $\mathcal{A}_2$, os *símbolos de pontuação* e os *símbolos conectivos sentenciais* são iguais aos utilizados em $\mathcal{A}_1$; e com o mesmo significado. Os *símbolos não-lógicos* são as letras minúsculas a, b, c, ..., y, z.

Em $\mathcal{L}_2$ as fórmulas bem formadas (fbfs) são chamadas de *esquemas de solução*. Tais fbfs são obtidas de acordo com as seguintes regras:

i) Os parâmetros a, b, c, ..., y, z, são esquemas de solução;

ii) Se a e b são esquemas de solução, então $(a \wedge b)$, $(a \vee b)$, e $(a \rightarrow b)$ também o são;

iii) Os esquemas de solução são somente as sentenças obtidas de i e ii acima.

Semelhantemente ao que foi feito na seção anterior, introduziremos o cálculo dedutivo de *esquemas de solução* que será definido através do seguinte conjunto Δ de axiomas e uma regra de inferência:

S_1 - $a \rightarrow (b \rightarrow a)$

S_2 - $(a \rightarrow (b \rightarrow c)) \rightarrow ((a \rightarrow b) \rightarrow (a \rightarrow c))$

S_3 - $(a \rightarrow (b \rightarrow a)) \rightarrow a$

S_4 - $a \rightarrow (a \vee b)$

S_5 - $b \rightarrow (a \vee b)$

S_6 - $(a \rightarrow c) \rightarrow ((b \rightarrow c) \rightarrow ((a \vee b) \rightarrow c))$

S_7 - $(a \wedge b) \rightarrow a$

S_8 - $(a \wedge b) \rightarrow b$

S_9 - $a \rightarrow (b \rightarrow (a \wedge b))$

I_1 - Se a e $a \rightarrow b$, então b. (*Modus Ponens*)

O conceito de igualdade entre dois esquemas de solução a e b é introduzido de modo semelhante ao que foi feito para esquemas de problema, a saber

$$\boxed{a = b \underset{\text{def}}{\equiv} (a \rightarrow b) \wedge (b \rightarrow a)}$$

Obtemos para o cálculo dedutivo de esquemas de solução, pela aplicação dos axiomas de Δ e regra de inferência I_1, resultados idênticos aos obtidos no cálculo de esquemas de problema, bem entendido, que no lugar dos esquemas de problemas estão os esquemas de solução.

As definições de prova e teorema na LES seguem o mesmo raciocínio da LEP.

De posse dos sistemas formais de esquemas de problemas e de esquemas de solução, passaremos à formalização da Lógica de Resolução de Problemas.

5.3 Lógica de Resolução de Problemas (LRP)

Numa visão sintática, a princípio, a Lógica de Resolução de Problemas se apresenta como um meio termo entre uma lógica sentencial e uma lógica de primeira ordem. Vai além de uma lógica sentencial ao contemplar a existência de uma relação predicativa entre seus termos, mas não generaliza proposições como numa lógica de primeira ordem. Esse modo de apresentação da LRP foge, sem dúvida, ao padrão canônico; mas como existia a idéia de se continuar a partir da LRP na concepção de uma lógica de ordem mais elevada para a tratativa de problemas, optou-se por incluir essa relação predicativa. Mas a presença da relação predicativa numa expressão da LRP, nunca aparece como um elemento independente, pois tal expressão sempre é lida e considerada como uma *proposição*.

Seja $\mathcal{L}_3$ a linguagem na qual expressaremos as idéias da LRP. O alfabeto $\mathcal{A}_3$ no qual se baseia esta linguagem é apresentado como segue:

1) *Símbolos lógicos*:

- i) *Símbolos de pontuação*: Parênteses, à esquerda e à direita: (,);
- ii) *Símbolos conectivos sentenciais*: $\neg$, $\mathbf{\wedge}$, $\mathbf{\vee}$ e $\Rightarrow$, para expressar, respectivamente, o sentido de "não", "e", "ou" e "implica" lógicos;
- iii) *Variáveis*: As letras maiúsculas **A, B, C, ..., Y, Z** para expressar as sentenças da LEP, e as minúsculas **a, b, c, ..., y, z** para expressar as sentenças da LES.

2) *Parâmetros*:

- i) *Símbolo predicativo*: $\|\!\rightarrow$, para indicar "é esquema para".

As fórmulas atômicas da LRP são expressões do tipo a ||→ A e sua negação, significando, respectivamente, "a é esquema para A", e "a não é esquema para A".

As fórmulas bem formadas nesta linguagem são definidas como segue:

i) As fórmulas atômicas são fórmulas bem formadas;
ii) Se α e β são fórmulas bem formadas, então $(\neg\alpha)$, $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$ e $(\alpha \Rightarrow \beta)$ também o são;
iii) As fórmulas bem formadas são somente obtidas de i e ii acima.

Com relação aos parênteses, os mesmos só serão utilizados quando forem estritamente necessários. No caso de uma expressão $(\alpha \wedge \beta)$, a mesma pode ser reescrita $\alpha \wedge \beta$, de modo mais simples, sem qualquer perda de significado.

A partir dos elementos desta lógica criam-se condições para o enunciado, a nível de proposições, das expressões envolvendo a relação estabelecida pelo simbolo "||→" existente entre *esquemas de solução* e *esquemas de problemas*. Tais expressões deverão ser obtidas através do sistema formal dedutivo para a LRP.

Capítulo 6

Sistema Formal Dedutivo e Atribuição de Verdade na LRP

Neste capítulo apresentamos um sistema formal dedutivo para a LRP e alguns resultados já apresentados em **[LOP 85]**. Apresentamos um esboço da máquina de inferência da LRP e introduzimos formalmente um modelo de atribuição de verdade para a LRP nos moldes originalmente propostos por Lopes e Acioly no trabalho acima citado.

6.1 Sistema Formal Dedutivo para a LRP

Um sistema formal dedutivo é composto de axiomas e regras de inferências. Axiomas são declarações primitivas (também chamadas de postulados); são aceitos como verdadeiros sem no entanto ser necessário estabelecer suas respectivas validades. As regras de inferência - ou regras de prova - servem para estabelecer como podem ser transformadas as expressões já tidas como válidas, como também gerar novas expressões válidas **[TAR 76]**.

A base para o cálculo dedutivo de resolução de problemas será o seguinte conjunto de axiomas Ω e regras de inferência:

Ax_1 - $\alpha \Rightarrow (\beta \Rightarrow \alpha)$

Ax_2 - $(\alpha \Rightarrow (\beta \Rightarrow \gamma)) \Rightarrow ((\alpha \Rightarrow \beta) \Rightarrow (\alpha \Rightarrow \gamma))$

Ax_3 - $(\neg\beta \Rightarrow \neg\alpha) \Rightarrow (\alpha \Rightarrow \beta)$ *(Contraposição)*

Rf_{MP} - $\dfrac{\alpha\ ,\ \alpha \Rightarrow \beta}{\beta}$ *(Modus Ponens I)*

Rf_{MPII} - $\dfrac{a \,\|\!\rightarrow A\ ,\ a \rightarrow b \,\|\!\rightarrow A \rightarrow B}{b \,\|\!\rightarrow B}$ *(Modus Ponens II)*

$Rf_{\rightarrow}$ - $\dfrac{a \,\|\!\rightarrow A\ ,\ b \,\|\!\rightarrow B}{a \rightarrow b \,\|\!\rightarrow A \rightarrow B}$ *(Regra Implicativa)*

$Rf_{\wedge}$ - $\dfrac{a \,\|\!\rightarrow A\ ,\ b \,\|\!\rightarrow B}{a \wedge b \,\|\!\rightarrow A \wedge B}$ *(Regra Conjuntiva)*

$Rf_{\vee}$ - $\dfrac{a \,\|\!\rightarrow A\ ,\ b \,\|\!\rightarrow B}{a \vee b \,\|\!\rightarrow A \vee B}$ *(Regra Disjuntiva)*

A composição de um sistema formal dedutivo é importante no sentido de que, como veremos no capítulo 7, se o mesmo for *correto*, há a garantia de que quando tal sistema de regras de inferência é aplicado a um determinado conjunto de fbfs (hipóteses), o que é gerado em decorrência desse processo (as conclusões) sempre será conseqüência do conjunto de hipóteses. Tais conseqüências são chamadas de *teoremas*.

Muito embora só seja apresentada a idéia de conseqüência (tauto)lógica na seção 6.3, mesmo assim deixamos aqui a noção de que os teoremas são exatamente as conseqüências tautológicas geradas a partir de um conjunto de hipóteses.

A partir do conceito de *atribuição de verdade* que também será apresentado na seção 6.3 - o qual é essencial para o estabelecimento do conceito de conseqüência tautológica - se verificará que o processo de "procura de teoremas" via idéia de *conseqüência* pode se tornar completamente inviável na prática. Por outro lado, a *dedução* de teoremas via sistema formal dedutivo é um processo essencialmente mecânico e perfeitamente exeqüível desde que o sistema de regras seja *correto*.

A fim de estabelecer quais são os teoremas na LRP, apresentamos uma definição de *prova* para esta lógica baseada na definição encontrada em **[LOP 85]**, a saber:

Definição 6.1.1 - Seja Σ um conjunto de fbfs da LRP, e α uma fbf. Uma *prova* (ou dedução) de α na LRP é uma seqüência de fbfs $\alpha_0, \ldots, \alpha_n$, $\alpha_n = \alpha$ tal que

i) α_i é um axioma de Ω, ou

ii) α_i é uma fbf de Σ, ou

iii) Para algum j e k menor do que i, α_i é obtido de α_j e α_k pela aplicação de alguma regra de inferência.

Observamos a partir da definição acima que *dedução* é um processo; e processo puramente mecânico. Tal processo mecânico pode ser utilizado como peça fundamental em uma ação de *nomeação*, isto é, uma ação de se *associar significado* a algo.

A partir da definição de teorema que apresentaremos a seguir, fica clara a idéia expressa no parágrafo anterior de que, embora haja uma ligação direta entre o conceito de dedução e a idéia de teorema, dedução é um processo e teorema é um nome.

Definição 6.1.2 - Os teoremas de um conjunto Σ de esquemas de problemas são as fórmulas bem formadas obtidas de $\Sigma \cup \Omega$ através do uso de regras de inferência um número finito de vezes, isto é, *se existe uma dedução de α a partir de Σ, então α é um teorema de Σ*. Nesse caso, diz-se que α é *dedutível* de Σ e simbolizamos por $\Sigma \vdash \alpha$.

O resultado da definição acima é utilizado para o desenvolvimento de um teorema da dedução para a LRP (Teorema 4.3 **[LOP 85]**). Seguindo os moldes da lógica proposicional clássica, o teorema da dedução estabelece que se temos um conjunto de sentenças Σ e sentenças γ e φ tal que tenhamos $\Sigma \cup \gamma \vdash \varphi$, então podemos transformar a dedução de φ a partir de $\Sigma \cup \gamma$ numa dedução de $(\gamma \rightarrow \varphi)$ a partir de Σ. O teorema da dedução se constitui o inverso da regra *Modus Ponens*. Na seção 6.5 apresentaremos um novo teorema da dedução para a LRP.

6.2 Máquina de Inferência da LRP

Como resumo esquemático do sistema formal dedutivo da LRP, apresentamos o seguinte diagrama:

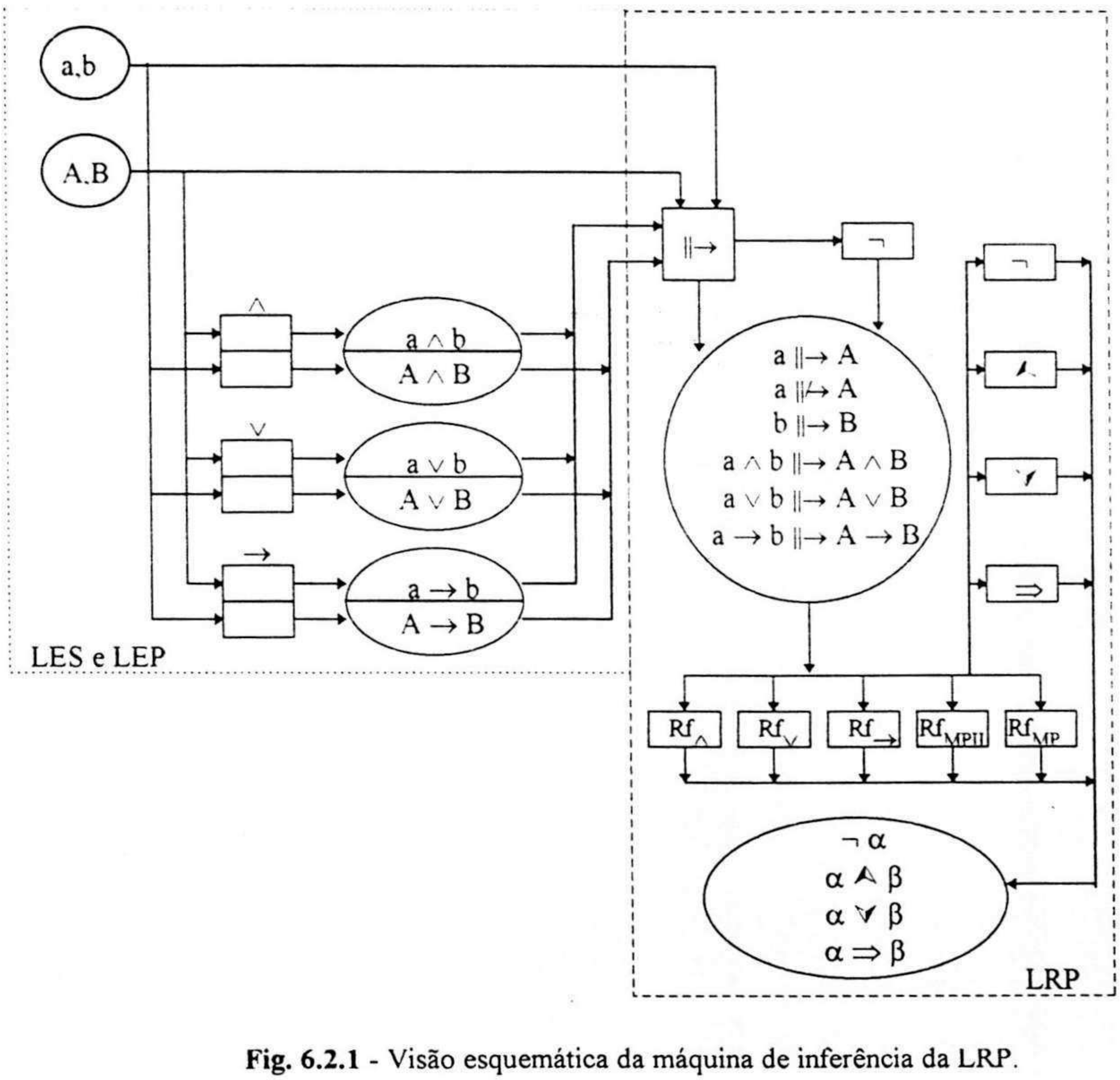

Fig. 6.2.1 - Visão esquemática da máquina de inferência da LRP.

6.3 Atribuição de Verdade na LRP

Posteriormente, para chegarmos a alguns resultados importantes na LRP, iremos necessitar do conceito de conseqüência lógica; mais precisamente, entender o que significa para uma fbf da LRP seguir logicamente de outras fbfs. Por exemplo, considere as fbfs α e $(\alpha \wedge \beta)$. α deve seguir de $(\alpha \wedge \beta)$. Não importando realmente o que os parâmetros α e β signifiquem, se a tradução de $(\alpha \wedge \beta)$ é verdadeira, então a tradução de α deve ser verdadeira. Mas a idéia de se verificar a implicação lógica amarrada a todas as possíveis traduções é laboravelmente vaga. Daí a necessidade de expressar a noção de conseqüência lógica de uma maneira simples e precisa. A definição de uma "atribuição de verdade" é de fundamental importância para atingirmos o objetivo de tratar conseqüência lógica de uma maneira puramente mecânica, não interessando as possíveis traduções das fbfs envolvidas **[END 72] [DAV 83] [TAR 76]**.

Mas a atribuição de verdade é uma atribuição de significado. Em algum momento pode haver a necessidade de se saber o significado de uma expressão escrita numa determinada linguagem; há de se saber se tal expressão é *válida* ou não de acordo com os critérios da realidade definida ou arbitrada - "a verdade depende da realidade" **[QUI 72]**. Por isso, Quine define a lógica como sendo a resultante de dois componentes: gramática e verdade **[QUI 72] [BAR 75]**.

Atribuição de Verdade na LRP

Seja $\mathcal{S}$ o conjunto de fórmulas atômicas da LRP e $\bar{\mathcal{S}}$ o conjunto de fbfs geradas pelas três regras de formação de fbfs dadas na seção 5.3. Seja também um conjunto $\{$**V**,**F**$\}$ de *valores verdade* consistindo de dois valores distintos:

V, chamado *verdadeiro*,

F, chamado *falso*.

Uma *atribuição de verdade* $\boldsymbol{v}$ para $\mathcal{S}$ é uma função

$$\boldsymbol{v}: \quad \mathcal{S} \longrightarrow \{\mathbf{V}, \mathbf{F}\}$$

$$s \,\|\!\rightarrow P \longrightarrow \begin{cases} \mathbf{V}, \text{ se } s \in \mathbf{R}^{\mathbf{D}} \equiv> s \,\|\!- P, \\ \mathbf{F}, \text{ senão.} \end{cases}$$

atribuindo **V** ou **F** a cada fórmula atômica em $\mathcal{S}$. Onde **D** e **R** são, respectivamente, os domínios de dados e de resultados do problema P.

Queremos uma extensão $\bar{v}$ de v, $\quad \bar{v} : \bar{\mathcal{S}} \longrightarrow \{\mathbf{V}, \mathbf{F}\}$,

que atribua o valor verdade correto a cada fbf em $\bar{\mathcal{S}}$. Tal extensão deve obedecer as seguintes condições:

a) Para qualquer **A** em $\mathcal{S}$, $\bar{v}(\mathbf{A}) = v(\mathbf{A})$.

Para qualquer α, β em $\bar{\mathcal{S}}$:

b) $\bar{v}((\neg\alpha)) = \begin{cases} \mathbf{V}, \text{ se } \bar{v}(\alpha) = \mathbf{F}, \\ \mathbf{F}, \text{ senão.} \end{cases}$

c) $\bar{v}((\alpha \wedge \beta)) = \begin{cases} \mathbf{V}, \text{ se } \bar{v}(\alpha) = \mathbf{V} \text{ e } \bar{v}(\beta) = \mathbf{V} \\ \mathbf{F}, \text{ senão.} \end{cases}$

d) $\bar{v}((\alpha \vee \beta)) = \begin{cases} \mathbf{V}, \text{ se } \bar{v}(\alpha) = \mathbf{V} \text{ ou } \bar{v}(\beta) = \mathbf{V} \text{ (ou ambos)} \\ \mathbf{F}, \text{ senão.} \end{cases}$

e) $\bar{v}((\alpha \Rightarrow \beta)) = \begin{cases} \mathbf{F}, \text{ se } \bar{v}(\alpha) = \mathbf{V} \text{ e } \bar{v}(\beta) = \mathbf{F} \\ \mathbf{V}, \text{ senão.} \end{cases}$

Definição 6.3.1 - Uma atribuição de verdade v *satisfaz* uma fbf ϕ, ou v é um *modelo* para ϕ, se e somente se $\bar{v}(\phi) = V$. **[CAS 87] [DAV 83]**

Definição 6.3.2 - Uma fbf ϕ é uma *tautologia* ($\models \phi$) se e somente se toda atribuição de verdade v é modelo para ϕ. **[CAS 87] [DAV 83]**

Finalmente, de posse do conceito de *satisfação*, passamos a definir o conceito de implicação (tauto)lógica. Considere Σ um conjunto de fbfs da LRP (pense como sendo as hipóteses) e uma outra fbf ϕ (pense como sendo uma possivel conclusão).

Definição 6.3.3 - Σ implica tautologicamente ϕ ($\Sigma \models \phi$) se e somente se toda atribuição verdade para os símbolos sentenciais em Σ e ϕ que é modelo todo membro de Σ também é modelo ϕ.

"Esta definição reflete nosso sentimento intuitivo de que uma conclusão segue de um conjunto de hipóteses se a suposição de que as hipóteses são verdadeiras resulta em que a conclusão é verdadeira" **[END 72]**.

Na verdade a implicação tautológica é mais forte do que a implicação lógica. A implicação lógica depende de interpretações das expressões no mundo real, enquanto que a implicação tautológica depende exclusivamente da sintaxe das expressões. "A validade dos argumentos dedutivos é determinada pela forma lógica e não pelo conteúdo das proposições que os compõem **[SAL 69]**.

Teorema 6.3.1 - Se $\alpha_1 \wedge \alpha_2 \wedge \cdots \wedge \alpha_n \models \phi$, então $\models (\alpha_1 \wedge \alpha_2 \wedge \cdots \wedge \alpha_n \Rightarrow \phi)$.

Demonstração: $(\alpha_1 \wedge \alpha_2 \wedge \cdots \wedge \alpha_n \Rightarrow \phi)$ não é uma tautologia no caso em que para alguma atribuição de verdade v, $\bar{v}(\alpha_1 \wedge \alpha_2 \wedge \cdots \wedge \alpha_n) = V$ e $\bar{v}(\phi) = F$. Isto é, exatamente no caso em que para alguma atribuição de verdade ν,

$\bar{\nu}(\alpha_1) = \bar{\nu}(\alpha_2) = \cdots = \bar{\nu}(\alpha_n) = \mathbf{V}$ e $\bar{\nu}(\phi) = \mathbf{F}$, o que significa simplesmente que não é o caso no qual $\alpha_1 \curlywedge \alpha_2 \curlywedge \cdots \curlywedge \alpha_n \models \phi$.

6.4 Equivalência de fbfs na LRP

De acordo com a definição 3.3.1 e o Teorema 3.3.1 estabelecemos as seguintes equivalências entre fbfs da LRP:

Definição 6.4.1 - Sejam a e b esquemas de solução e A e B esquemas de problemas, e (s_1, s_2) um par ordenado de esquemas de solução s_1 e s_2. Então,

a) Para $(s_1, s_2) = a \wedge b$ e $(s_1, s_2) \Vdash A \wedge B$, temos

$$a \wedge b \; \| \!\rightarrow A \wedge B \;\equiv\; a \; \| \!\rightarrow A \curlywedge b \; \| \!\rightarrow B.$$

b) Para $(s_1, s_2) = a \vee b$ e $(s_1, s_2) \Vdash A \vee B$, temos

$$a \vee b \; \| \!\rightarrow A \vee B \;\equiv\; a \; \| \!\rightarrow A \curlyvee b \; \| \!\rightarrow B.$$

c) Para $(s_1, s_2) = a \rightarrow b$ e $(s_1, s_2) \Vdash A \rightarrow B$, temos

$$a \rightarrow b \; \| \!\rightarrow A \rightarrow B \;\equiv\; a \; \| \!\rightarrow A \Rightarrow b \; \| \!\rightarrow B.$$

6.5 Novo Teorema da Dedução para a LRP

Na parte dedicada à conclusão em **[LOP 85]**, os autores sugerem que um novo teorema da dedução seja elaborado de modo que o novo teorema se constitua não só como o inverso da regra *Modus Ponens* (Rf_{MP}) mas também de todas as outras regras de inferência que compõe o sistema formal dedutivo da LRP. Seguindo a sugestão dos autores desenvolvemos um teorema da dedução completo que apresentaremos nesta seção.

Teorema 6.5.1 - Seja Σ um conjunto de fbfs da LRP e α, β, $a \Vert\!\rightarrow A$, $b \Vert\!\rightarrow B$ fbfs.

a) Se $\Sigma \cup \{\alpha\} \vdash \beta$, então $\Sigma \vdash \alpha \Rightarrow \beta$.

b) Se $\Sigma \cup \{ a \Vert\!\rightarrow A \} \vdash b \Vert\!\rightarrow B$, então $\Sigma \vdash a \rightarrow b \Vert\!\rightarrow A \rightarrow B$.

c) Se $\Sigma \cup \{ a \Vert\!\rightarrow A \} \vdash a \rightarrow b \Vert\!\rightarrow A \rightarrow B$, então $\Sigma \vdash b \Vert\!\rightarrow B$.

d) Se $\Sigma \cup \{ a \Vert\!\rightarrow A \} \vdash a \wedge b \Vert\!\rightarrow A \wedge B$, então $\Sigma \vdash b \Vert\!\rightarrow B$.

e) Se $\Sigma \cup \{ a \Vert\!\rightarrow A \} \vdash a \vee b \Vert\!\rightarrow A \vee B$, então $\Sigma \vdash b \Vert\!\rightarrow B$.

Demonstração: a) Mostramos por indução sobre todo teorema β de $\Sigma \cup \{\alpha\}$ que a fórmula $\alpha \Rightarrow \beta$ é teorema de Σ.

Caso1: $\beta = \alpha$. Então, obviamente $\vdash \alpha \Rightarrow \beta$ (pois $\vdash \alpha \Rightarrow \alpha$), e forçosamente $\underline{\Sigma \vdash \alpha \Rightarrow \beta}$.

Caso 2: $\beta \in \Sigma \cup \Omega$. Daí, $\Sigma \vdash \beta$.

(1)	β	Hipótese
(2)	$\beta \Rightarrow (\alpha \Rightarrow \beta)$	$[Ax_1]$
(3)	$\alpha \Rightarrow \beta$	(1), (2), Rf_{MP}

Então, $\underline{\Sigma \vdash \alpha \Rightarrow \beta}$.

Caso 3: β é obtido por Modus Ponens de γ e $\gamma \Rightarrow \beta$. Pela hipótese da indução, $\Sigma \vdash \alpha \Rightarrow \gamma$ [Pois $\Sigma \cup \{\alpha\} \vdash \gamma$] e $\Sigma \vdash \alpha \Rightarrow (\gamma \Rightarrow \beta)$ [Pois $\Sigma \cup \{\alpha\} \vdash \gamma \Rightarrow \beta$].

(1)	$\alpha \Rightarrow \gamma$	Hipótese
(2)	$\alpha \Rightarrow (\gamma \Rightarrow \beta)$	Hipótese
(3)	$(\alpha \Rightarrow (\gamma \Rightarrow \beta)) \Rightarrow ((\alpha \Rightarrow \gamma) \Rightarrow (\alpha \Rightarrow \beta))$	$[Ax_2]$
(4)	$(\alpha \Rightarrow \gamma) \Rightarrow (\alpha \Rightarrow \beta)$	(2), (3), Rf_{MP}

(5) $\alpha \Rightarrow \beta$ (1), (4), Rf_{MP}

Então, $\underline{\Sigma \vdash \alpha \Rightarrow \beta}$.

b) Mostramos por indução sobre o teorema b $\|\rightarrow$ B de $\Sigma \cup \{a \|\rightarrow A\}$ que a fórmula a $\rightarrow$ b $\|\rightarrow$ A $\rightarrow$ B é teorema de Σ.

<u>Caso 1</u>: b $\|\rightarrow B \in \Sigma \cup \Omega$. Daí, $\Sigma \vdash$ b $\|\rightarrow$ B.

(1) $b \|\rightarrow B$ Hipótese

(2) $b \|\rightarrow B \Rightarrow (a \|\rightarrow A \Rightarrow b \|\rightarrow B)$ $[Ax_1]$

(2') $b \|\rightarrow B \Rightarrow (a \rightarrow b \|\rightarrow A \rightarrow B)$ (2), item c) seção 6.4

(3) $a \rightarrow b \|\rightarrow A \rightarrow B$ (1), (2'), Rf_{MP}

Então, $\underline{\Sigma \vdash a \rightarrow b \|\rightarrow A \rightarrow B}$.

<u>Caso 2</u>: b $\|\rightarrow$ B é obtido por Modus Ponens de γ e $\gamma \Rightarrow b \|\rightarrow B$. Pela hipótese da indução, $\Sigma \vdash a \|\rightarrow A \Rightarrow \gamma$ e $\Sigma \vdash a \|\rightarrow A \Rightarrow (\gamma \Rightarrow b \|\rightarrow B)$.

(1) $a \|\rightarrow A \Rightarrow \gamma$ Hipótese

(2) $a \|\rightarrow A \Rightarrow (\gamma \Rightarrow b \|\rightarrow B)$ Hipótese

(3) $(a \|\rightarrow A \Rightarrow (\gamma \Rightarrow b \|\rightarrow B)) \Rightarrow$

$(a \|\rightarrow A \Rightarrow \gamma) \Rightarrow (a \|\rightarrow A \Rightarrow b \|\rightarrow B))$ $[Ax_2]$

(3') $(a \|\rightarrow A \Rightarrow (\gamma \Rightarrow b \|\rightarrow B)) \Rightarrow$ (3), item c) seção 6.4

$((a \|\rightarrow A \Rightarrow \gamma) \Rightarrow (a \rightarrow b \|\rightarrow A \rightarrow B))$

(4) $(a \|\rightarrow A \Rightarrow \gamma) \Rightarrow (a \rightarrow b \|\rightarrow A \rightarrow B)$ (2), (3'), Rf_{MP}

(5) $a \rightarrow b \|\rightarrow A \rightarrow B$ (1), (4), Rf_{MP}

Então, $\underline{\Sigma \vdash a \rightarrow b \|\rightarrow A \rightarrow B}$.

<u>Caso 3</u>: $b \|\!\!\rightarrow B$ é obtido por Modus Ponens II de $c \|\!\!\rightarrow C$ e $c \rightarrow b \|\!\!\rightarrow C \rightarrow B$. Pela hipótese da indução, $\Sigma \vdash a \|\!\!\rightarrow A \Rightarrow c \|\!\!\rightarrow C$ e $\Sigma \vdash a \|\!\!\rightarrow A \Rightarrow (c \rightarrow b \|\!\!\rightarrow C \rightarrow B)$.

(1)	$a \|\!\!\rightarrow A \Rightarrow c \|\!\!\rightarrow C$	Hipótese
(1')	$a \rightarrow c \|\!\!\rightarrow A \rightarrow C$	(1), ítem c) seção 6.4
(1'')	$y \|\!\!\rightarrow Y$	(1') desde que $y \|\!\!\rightarrow Y \equiv a \rightarrow c \|\!\!\rightarrow A \rightarrow C$
(2)	$a \|\!\!\rightarrow A \Rightarrow (c \rightarrow b \|\!\!\rightarrow C \rightarrow B)$	Hipótese
(2')	$a \|\!\!\rightarrow A \Rightarrow x \|\!\!\rightarrow X$	(2) desde que $x \|\!\!\rightarrow X \equiv c \rightarrow b \|\!\!\rightarrow C \rightarrow B$
(2'')	$a \rightarrow x \|\!\!\rightarrow A \rightarrow X$	(2'), ítem c) seção 6.4
(2''')	$u \|\!\!\rightarrow U$	(2'') desde que $u \|\!\!\rightarrow U \equiv a \rightarrow x \|\!\!\rightarrow A \rightarrow X$
(3)	$(a \|\!\!\rightarrow A \Rightarrow (c \|\!\!\rightarrow C \Rightarrow b \|\!\!\rightarrow B)) \Rightarrow$ $((a \|\!\!\rightarrow A \Rightarrow c \|\!\!\rightarrow C) \Rightarrow a \|\!\!\rightarrow b \Rightarrow A \|\!\!\rightarrow B)$	$[Ax_2]$
(3')	$(a \|\!\!\rightarrow A \Rightarrow (c \rightarrow b \|\!\!\rightarrow C \rightarrow B)) \Rightarrow$ $((a \|\!\!\rightarrow A \Rightarrow c \|\!\!\rightarrow C) \Rightarrow a \rightarrow b \|\!\!\rightarrow A \rightarrow B)$	(3), ítem c) seção 6.4
(3'')	$u \|\!\!\rightarrow U \Rightarrow (y \|\!\!\rightarrow Y \Rightarrow t \|\!\!\rightarrow T))$	(3'), (1''), (2''') desde que $t \|\!\!\rightarrow T \equiv a \rightarrow b \|\!\!\rightarrow A \rightarrow B$
(3''')	$u \|\!\!\rightarrow U \Rightarrow z \|\!\!\rightarrow Z$	(3'') desde que $z \|\!\!\rightarrow Z \equiv y \|\!\!\rightarrow Y \Rightarrow t \|\!\!\rightarrow T$
(3'''')	$u \rightarrow z \|\!\!\rightarrow U \rightarrow Z$	(3'''), ítem c) seção 6.4
(4)	$z \|\!\!\rightarrow Z$	(2'''), (3''''), Rf_{MPII}
(4')	$y \|\!\!\rightarrow Y \Rightarrow t \|\!\!\rightarrow T$	(3''')
(4'')	$y \rightarrow t \|\!\!\rightarrow Y \rightarrow T$	(4'), ítem c) seção 6.4
(5)	$t \|\!\!\rightarrow T$	(1''), (4''), Rf_{MPII}

(5') $a \to b \Vert\!\!\to A \to B$ (3'')

Então, $\underline{\Sigma \vdash a \to b \Vert\!\!\to A \to B}$.

Aqui abriremos um parênteses para comentar a respeito da natureza de $Rf_{\to}$, $Rf_{\wedge}$ e $Rf_{\vee}$ que difere da natureza de Rf_{MP} e Rf_{MPII}. Enquanto Rf_{MP} e Rf_{MPII} têm claramente uma natureza de *separação*, $Rf_{\to}$, $Rf_{\wedge}$ e $Rf_{\vee}$ têm características de *aglutinação*. Desse modo, a natureza das regras de aglutinação influem diretamente no processo de prova do teorema da dedução que se relaciona com elas.

c) Mostramos por indução sobre o teorema $a \to b \Vert\!\!\to A \to B$ de $\Sigma \cup \{a \Vert\!\!\to A\}$ que a fórmula $b \Vert\!\!\to B$ é teorema de Σ. $a \to b \Vert\!\!\to A \to B$ é obtido a partir de $\Sigma \cup \{ a \Vert\!\!\to A \}$. Pela hipótese da indução temos $\Sigma \vdash a \Vert\!\!\to A \Rightarrow (a \to b \Vert\!\!\to A \to B)$.

(1) $a \Vert\!\!\to A \Rightarrow (a \to b \Vert\!\!\to A \to B)$ Hipótese

(1') $a \Vert\!\!\to A \Rightarrow (a \Vert\!\!\to A \Rightarrow b \Vert\!\!\to B)$ (1), ítem c) seção 6.4

(2) $a \Vert\!\!\to A \Rightarrow a \Vert\!\!\to A$ $[\Vdash\alpha \Rightarrow \alpha]$

(3) $(a \Vert\!\!\to A \Rightarrow (a \Vert\!\!\to A \Rightarrow b \Vert\!\!\to B)) \Rightarrow$ $[Ax_2]$
$((a \Vert\!\!\to A \Rightarrow a \Vert\!\!\to A) \Rightarrow (a \Vert\!\!\to A \Rightarrow b \Vert\!\!\to B))$

(4) $(a \Vert\!\!\to A \Rightarrow a \Vert\!\!\to A) \Rightarrow (a \Vert\!\!\to A \Rightarrow b \Vert\!\!\to B))$ (1'), (3), Rf_{MP}

(5) $a \Vert\!\!\to A \Rightarrow b \Vert\!\!\to B$ (2), (4), Rf_{MP}

(6) $\neg(a \Vert\!\!\to A) \curlyvee \underline{b \Vert\!\!\to B} \curlyvee (\underline{a \Vert\!\!\to A \curlywedge b \Vert\!\!\to B})$ (5)

(6') $b \Vert\!\!\to B$ A partir de (6)

Então, $\underline{\Sigma \vdash b \Vert\!\!\to B}$. Ou ainda

(6'') $a \Vert\!\!\to A \curlywedge b \Vert\!\!\to B$ A partir de (6)

(7) $(a \Vert\!\!\to A \curlywedge b \Vert\!\!\to B) \Rightarrow b \Vert\!\!\to B$ [Apêndice A]

(8) $b \| \!\rightarrow B$ (6'),(7), Rf_{MP}

Então, $\underline{\Sigma \vdash b \| \!\rightarrow B}$.

d) Mostramos por indução sobre o teorema $a \wedge b \| \!\rightarrow A \wedge B$ de $\Sigma \cup \{a \| \!\rightarrow A\}$ que a fórmula $b \| \!\rightarrow B$ é teorema de Σ. $a \wedge b \| \!\rightarrow A \wedge B$ é obtido a partir de $\Sigma \cup \{ a \| \!\rightarrow A \}$. Pela hipótese da indução temos $\Sigma \vdash a \| \!\rightarrow A \Rightarrow (a \wedge b \| \!\rightarrow A \wedge B)$.

(1) $a \| \!\rightarrow A \Rightarrow (a \wedge b \| \!\rightarrow A \wedge B)$ Hipótese

(1') $a \| \!\rightarrow A \Rightarrow (a \| \!\rightarrow A \curlywedge b \| \!\rightarrow B)$ (1), ítem a) seção 6.4

(2) $\neg(a \| \!\rightarrow A) \curlyvee (a \| \!\rightarrow A \curlywedge b \| \!\rightarrow B)$ (1')

(3) $\neg(a \| \!\rightarrow A) \curlyvee \neg(a \| \!\rightarrow A) \curlyvee \neg(b \| \!\rightarrow B)$ (2)

(4) $\neg(a \| \!\rightarrow A) \curlyvee \neg(b \| \!\rightarrow B)$ (3)

(5) $a \| \!\rightarrow A \curlywedge b \| \!\rightarrow B$ (4)

(6) $(a \| \!\rightarrow A \curlywedge b \| \!\rightarrow B) \Rightarrow b \| \!\rightarrow B$ [Apêndice A]

(7) $b \| \!\rightarrow B$ (5), (6), Rf_{MP}

Então, $\underline{\Sigma \vdash b \| \!\rightarrow B}$.

e) Mostramos por indução sobre o teorema $a \vee b \| \!\rightarrow A \vee B$ de $\Sigma \cup \{a \| \!\rightarrow A\}$ que a fórmula $b \| \!\rightarrow B$ é teorema de Σ. $a \vee b \| \!\rightarrow A \vee B$ é obtido a partir de $\Sigma \cup \{ a \| \!\rightarrow A \}$. Pela hipótese da indução temos $\Sigma \vdash a \| \!\rightarrow A \Rightarrow (a \vee b \| \!\rightarrow A \vee B)$.

(1) $a \| \!\rightarrow A \Rightarrow (a \vee b \| \!\rightarrow A \vee B)$ Hipótese

(1') $a \| \!\rightarrow A \Rightarrow (a \| \!\rightarrow A \curlyvee b \| \!\rightarrow B)$ (1), ítem b) seção 6.4

(2) $\neg(a \Vdash\!\rightarrow A) \curlyvee a \Vdash\!\rightarrow A \curlyvee b \Vdash\!\rightarrow B$ (1')

(3) $b \Vdash\!\rightarrow B$ (2)

Então, $\underline{\Sigma \vdash b \Vdash\!\rightarrow B}$.

O Teorema 4.4 de **[LOP 85]** "estabelece de modo mais completo" os resultados pretendidos na ligação entre o cálculo de esquemas de solução e o cálculo de esquemas de problema. Os resultados, em número de nove, são devidamente amparados pelas respectivas provas baseadas em Ω e nas regras de inferência mostrados anteriormente. Como as provas são extensas e já se acham expostas em **[LOP 85]**, a seguir listaremos apenas os resultados do teorema.

6.6 Resultados do Teorema 4.4 (**[LOP 85]**)

Para $a \Vdash\!\rightarrow A$, $b \Vdash\!\rightarrow B$ e $c \Vdash\!\rightarrow C$, temos:

1 - $a \rightarrow (b \rightarrow a) \Vdash\!\rightarrow A \rightarrow (B \rightarrow A)$

2 - $(a \rightarrow (b \rightarrow c)) \rightarrow ((a \rightarrow b) \rightarrow (a \rightarrow c)) \Vdash\!\rightarrow (A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$

3 - $(a \rightarrow (b \rightarrow a)) \rightarrow a \Vdash\!\rightarrow (A \rightarrow (B \rightarrow A)) \rightarrow A$

4 - $a \rightarrow (a \vee b) \Vdash\!\rightarrow A \rightarrow (A \vee B)$

5 - $b \rightarrow (a \vee b) \Vdash\!\rightarrow B \rightarrow (A \vee B)$

6 - $(a \rightarrow c) \rightarrow ((b \rightarrow c) \rightarrow ((a \vee b) \rightarrow c)) \Vdash\!\rightarrow (A \rightarrow C) \rightarrow ((B \rightarrow C) \rightarrow ((A \vee B) \rightarrow C))$

7 - $(a \wedge b) \rightarrow a \Vdash\!\rightarrow (A \wedge B) \rightarrow A$

8 - $(a \wedge b) \rightarrow b \Vdash\!\rightarrow (A \wedge B) \rightarrow B$

9 - $a \rightarrow (b \rightarrow (a \wedge b)) \Vdash\!\rightarrow A \rightarrow (B \rightarrow (A \wedge B))$

O sistema formal dedutivo da LRP tem a característica de fazer transparecer, a princípio, que esquemas de solução e esquemas de problemas, quando ligados pela relação "é esquema para", tenham a mesma natureza de formação, isto é, a natureza de formação do esquema de solução herda, a princípio, a natureza de formação do esquema de problema. A "genética" de formação da expressão envolvendo os esquemas de problemas é preservada na formação da expressão envolvendo os esquemas de solução.

Esse fato interessante levou a formulação do Teorema 4.5 **[LOP 85]**. Esse teorema mostra que se na LES existe uma prova $a_0, \ldots, a_n$ para a e na LEP existe uma prova $A_0, \ldots, A_n$ para A, e cada a_i, A_i estão relacionados de modo que $a_0 \| \!\!\rightarrow A_0, \ldots, a_n \| \!\!\rightarrow A_n$, então $a \| \!\!\rightarrow A$ se constitui num teorema da LRP. Em outras palavras, a partir de um mecanismo de dedução que trabalhe em paralelo as sentenças da LEP e da LES, mantendo as sentenças de cada sistema formal referenciadas uma a uma na LRP, se pode deduzir teoremas da LRP.

Com base no importante resultado do Teorema 4.5 acima citado, Lopes e Acioly criam o conceito de *adequação*, definido a seguir:

Definição 6.6.1 - Um esquema de solução a é dito *adequado* a um esquema de problema A, denotado por $a \mid\!\!\rightarrow A$, se e somente se, $a \| \!\!\rightarrow A$ e a e A são, respectivamente, teoremas da Lógica de Esquemas de Solução e da Lógica de Esquemas de Problemas.

Portanto, o conceito de *adequado* quando aplicado a um teorema da LRP é o certificado de garantia de que houve corretude no processo de dedução de tal teorema, ou ainda, que há legitimidade no processo de ligação do esquema de solução com o esquema de problemas através da relação "é esquema para".

Capítulo 7

Corretude, Consistência, Compacidade e Completude na LRP

Neste capítulo procuraremos desenvolver os teoremas da consistência, corretude, compacidade e completude para a LRP nos moldes como foi apresentada até aqui. Esses quatro teoremas se constituem em indicativos chaves do bom comportamento de qualquer lógica como instrumento legitimo e completo de inferência lógica.

7.1 Corretude do Sistema Formal Dedutivo da LRP

Dizer que um sistema de regras de inferência é *correto* (ou *legitimo*) é dizer que qualquer conclusão deduzida com seu auxilio é conseqüência das premissas a partir das quais foi obtida. Assim, para mostrar que o Sistema Formal Dedutivo da LRP (baseado no conjunto de axiomas Ω e nas regras de inferência Rf_{MP}, Rf_{MPII}, $Rf_{\rightarrow}$, $Rf_{\wedge}$ e $Rf_{\vee}$) é correto, deveremos mostrar que para qualquer fbf ϕ e conjunto de fbfs Γ, se ϕ for dedutível de Γ, então ϕ será conseqüência de Γ.

Teorema 7.1.1 - [Corretude] O sistema formal dedutivo da LRP é correto.

$(\Gamma \vdash \phi \implies \Gamma \models \phi)$.

A idéia da prova é que os axiomas lógicos são logicamente implicados por alguma coisa, e que as regras de inferência preservam as implicações lógicas.

Lema 7.1.1 - Os axiomas lógicos de Ω são tautologias.

Assumindo o lema mostraremos por indução que qualquer fórmula ϕ dedutível de Γ é logicamente implicada por Γ.

Caso 1: ϕ é axioma lógico. Então pelo lema $\models \phi$, se $\emptyset \models \phi$, então forçosamente, $\Gamma \models \phi$.

Caso 2: $\phi \in \Gamma$. Então, claramente $\Gamma \models \phi$.

Caso 3: ϕ é obtido de Γ por uma das cinco regras de inferência.

3.a) ϕ é obtida de ψ e $\psi \Rightarrow \phi$ por *modus ponens*. Pela hipótese da indução $\Gamma \models \psi$ e $\Gamma \models (\psi \Rightarrow \phi)$, então segue-se que $\Gamma \models \phi$. Isto é, $(\psi \wedge (\psi \Rightarrow \phi)) \models \phi$, pois $\models \psi \wedge (\psi \Rightarrow \phi)) \Rightarrow \phi$ (De acordo com o Teorema 6.3.1).

3.b) ϕ é obtida de $a \Vert\!\rightarrow \mathrm{A}$ e $a \rightarrow b \Vert\!\rightarrow \mathrm{A} \rightarrow \mathrm{B}$ por *modus ponens II*. $\phi = b \Vert\!\rightarrow \mathrm{B}$ [vide Rf_{MPII}]. Então, Pela hipótese da indução temos $\{a \Vert\!\rightarrow \mathrm{A}, (a \rightarrow b \Vert\!\rightarrow \mathrm{A} \rightarrow \mathrm{B})\} \models b \Vert\!\rightarrow \mathrm{B}$.

(1)	$(a \Vert\!\rightarrow \mathrm{A} \wedge (a \rightarrow b \Vert\!\rightarrow \mathrm{A} \rightarrow \mathrm{B})) \models b \Vert\!\rightarrow \mathrm{B}$	Hipótese
(2)	$(a \Vert\!\rightarrow \mathrm{A} \wedge (a \Vert\!\rightarrow \mathrm{A} \Rightarrow b \Vert\!\rightarrow \mathrm{B})) \models b \Vert\!\rightarrow \mathrm{B}$	(1), ítem c) seção 6.4
(3)	$\models (a \Vert\!\rightarrow \mathrm{A} \wedge (a \Vert\!\rightarrow \mathrm{A} \Rightarrow b \Vert\!\rightarrow \mathrm{B})) \Rightarrow b \Vert\!\rightarrow \mathrm{B}$	Teorema 6.3.1

3.c) ϕ é obtida de $a \parallel\!\rightarrow A$ e $b \parallel\!\rightarrow B$ por *regra implicativa*. Então, pela hipótese da indução temos $\{a \parallel\!\rightarrow A,\ b \parallel\!\rightarrow B\} \models (a \rightarrow b \parallel\!\rightarrow A \rightarrow B)$.

(1)	$(a \parallel\!\rightarrow A \curlywedge b \parallel\!\rightarrow B) \models (a \rightarrow b \parallel\!\rightarrow A \rightarrow B)$	Hipótese
(2)	$(a \parallel\!\rightarrow A \curlywedge b \parallel\!\rightarrow B) \models (a \parallel\!\rightarrow A \Rightarrow b \parallel\!\rightarrow B)$	(1), ítem c) seção 6.4
(3)	$\models (a \parallel\!\rightarrow A \curlywedge b \parallel\!\rightarrow B) \Rightarrow (a \parallel\!\rightarrow A \Rightarrow b \parallel\!\rightarrow B)$	Teorema 6.3.1

3.d) ϕ é obtida de $a \parallel\!\rightarrow A$ e $b \parallel\!\rightarrow B$ por *regra conjuntiva*. Então, pela hipótese da indução temos $\{a \parallel\!\rightarrow A,\ b \parallel\!\rightarrow B\} \models (a \wedge b \parallel\!\rightarrow A \wedge B)$.

(1)	$(a \parallel\!\rightarrow A \curlywedge b \parallel\!\rightarrow B) \models (a \wedge b \parallel\!\rightarrow A \wedge B)$	Hipótese
(2)	$(a \parallel\!\rightarrow A \curlywedge b \parallel\!\rightarrow B) \models (a \parallel\!\rightarrow A \curlywedge b \parallel\!\rightarrow B)$	(1), ítem a) seção 6.4
(3)	$\models (a \parallel\!\rightarrow A \curlywedge b \parallel\!\rightarrow B) \Rightarrow (a \parallel\!\rightarrow A \curlywedge b \parallel\!\rightarrow B)$	Teorema 6.3.1

3.e) ϕ é obtida de $a \parallel\!\rightarrow A$ e $b \parallel\!\rightarrow B$ por *regra disjuntiva*. Então, pela hipótese da indução temos $\{a \parallel\!\rightarrow A,\ b \parallel\!\rightarrow B\} \models (a \vee b \parallel\!\rightarrow A \vee B)$.

(1)	$(a \parallel\!\rightarrow A \curlywedge b \parallel\!\rightarrow B) \models (a \vee b \parallel\!\rightarrow A \vee B)$	Hipótese
(2)	$(a \parallel\!\rightarrow A \curlywedge b \parallel\!\rightarrow B) \models (a \parallel\!\rightarrow A \curlyvee b \parallel\!\rightarrow B)$	(1), ítem b) seção 6.4
(3)	$\models (a \parallel\!\rightarrow A \curlywedge b \parallel\!\rightarrow B) \Rightarrow (a \parallel\!\rightarrow A \curlyvee b \parallel\!\rightarrow B)$	Teorema 6.3.1

Resta agora, portanto, provar o lema 7.1.1. Ao se examinar a sintaxe de Ax_1, Ax_2 e Ax_3, percebe-se que tais axiomas são tautologias da lógica proposicional clássica (**[COS 79]**,**[MOR 67]**,**[MAT 68]**, **[ROB 65]**).[Vide Apêndice B].

O resultado mais importante advindo do teorema da corretude é que um sistema formal dedutivo correto jamais vai derivar uma fbf que esteja dissociada das fbfs originais que a geraram. Daí porque a descrição do teorema explora a questão do *significado* envolvendo tudo aquilo que está sendo *deduzido*.

Um sistema formal dedutivo é *consistente* se não há fbf ϕ tal que tanto ϕ como $\neg\phi$ sejam dedutiveis de um conjunto de fbfs Λ **[MAT 68] [ROB 65]**. Portanto, claro está que se o sistema de regras de inferência é correto, também é consistente, pois nenhuma fbf e sua respectiva negação são ambas conseqüências de Λ. Por outro lado, consistência não é garantia de legitimidade. Não é difícil imaginar uma regra de inferência consistente, mas que não seja correta. Por exemplo, uma regra que estabelecesse o seguinte: "deduza a fbf S (e nenhuma outra fbf) de qualquer conjunto de fbfs". Com tal regra de inferência não se pode garantir a legitimidade de tal sistema formal quando aplicado a qualquer conjunto de fbfs.

Do exposto acima concluímos que as regras de inferência (Rf_{MP}, Rf_{MPII}, $Rf_{\rightarrow}$, $Rf_{\wedge}$ e $Rf_{\vee}$) foram convenientemente escolhidas de modo a tornar correto o sistema formal dedutivo da LRP. Ou seja, as regras de inferência escolhidas para compor um sistema formal dedutivo correto devem ser de tal modo que estejam em consonância com o conjunto de fbfs Γ sobre as quais as mesmas operarão. O sistema formal dedutivo apresentado para a LRP não tornou-se correto por acaso, mas foi planejado para ser correto.

7.2 Maximalidade Consistente de um Conjunto de Fbfs da LRP.

Definição 7.2.1 - Um conjunto Γ de fbfs da LRP é dito *consistente* se e somente se não existe qualquer fbf ϕ, tal que ϕ e sua negação pertençam a Γ. **[CHA 77]** (Se $\phi \in \Gamma$ então $\neg\phi \notin \Gamma$, ou se $\neg\phi \in \Gamma$ então $\phi \notin \Gamma$).

Podemos perceber do resultado do teorema 7.1.1, que se o nosso sistema de regras de inferência for sucessivamente aplicado sobre um conjunto consistente Σ de fbfs da LRP, as fbfs derivadas dessas sucessivas aplicações podem ir sendo acrescentadas ao conjunto original Σ, de modo que o mesmo vai sendo expandido, mas esta expansão continuará sempre consistente. Desse modo, qualquer fbf α passível de dedução a partir de Σ, certamente pertencerá a algum subconjunto da expansão de Σ. A intuição acima é garantida ser verdade pelo teorema da consistência maximal.

Definição 7.2.2 - Um conjunto Γ de fbfs da LRP é dito *maximal* se para qualquer fbf ϕ, ϕ pertence a Γ ou $\neg\phi$ pertence a Γ. ($\phi \in \Gamma$ ou $\neg\phi \in \Gamma$). Além disso, se $\phi \in \Lambda$, então $\Lambda \subseteq \Gamma$.

Definição 7.2.3 - Um conjunto Γ de fbfs da LRP é dito *maximal consistente* se e só se Γ é consistente e o único conjunto consistente de fbfs que contém Γ é o próprio Γ **[CHA 77]** [($\phi \in \Gamma$ e $\neg\phi \notin \Gamma$) ou ($\neg\phi \in \Gamma$ e $\phi \notin \Gamma$)].

Teorema 7.2.1 - [Consistência Maximal] Qualquer conjunto consistente Σ de fbfs da LRP pode ser expandido até um conjunto maximal consistente Γ de fbfs.

Demonstração: Construamos uma lista $\phi_0, \phi_1, \ldots, \phi_n, \ldots$ das fbfs da LRP. A ordem em que as fbfs ϕ_i são listadas não importa, contanto que a lista associe um número ordinal a cada fbf.

Seja $\Sigma = \Sigma_0 \subset \Sigma_1 \subset \ldots \subset \Sigma_n \subset \ldots$ um encadeamento crescente de conjuntos consistentes de fbfs da LRP. Σ será construído do seguinte modo: se $\Sigma_0 \cup \{\phi_0\}$ é consistente, defina $\Sigma_1 = \Sigma_0 \cup \{\phi_0\}$. Senão, defina $\Sigma_1 = \Sigma_0$. E assim por diante, de modo que no n-ésimo estágio, definimos $\Sigma_{n+1} = \Sigma_n \cup \{\phi_n\}$ se $\Sigma_n \cup \{\phi_n\}$ é consistente, e caso contrário, definimos $\Sigma_{n+1} = \Sigma_n$. Assim, no limite do ordinal n temos,

$\Sigma_n = \cup \Sigma_m$, $m < n$. Agora, considere Γ como sendo a união de todos os conjuntos Σ_n.

1. Nossa afirmação é: Γ é consistente. Suponha que Γ não é consistente. Então existe uma dedução $\alpha_1, \alpha_2, \ldots, \alpha_j$ da fbf $\sigma \wedge \neg\sigma$ a partir de Γ. Seja $\theta_1, \theta_2, \ldots, \theta_k$ todas as fbfs em Γ que foram usadas nessa dedução. Podemos escolher um n de modo que $\theta_1, \theta_2, \ldots, \theta_k$ pertençam a Σ_n. Nesse caso Σ_n seria inconsistente, o que é uma contradição. [*O teorema não estabelece a natureza da origem das fbfs ϕ_i ou a natureza da fbf $\sigma \wedge \neg\sigma$. As fbfs ϕ_i tanto podem já existir num "banco de fbfs" quanto serem deduzidas. A própria dedução das fbfs ϕ_i como também da fbf $\sigma \wedge \neg\sigma$ pode ser por pertinência ou via alguma regra de inferência. Σ poderia se tornar inconsistente por incorporar dentro de si fbfs inconsistentes preexistentes, mas esse risco foi eliminado devido a maneira como Σ foi construído, isto é, no processo de escolha, cada ϕ_i escolhida que tornasse Σ inconsistente seria automaticamente eliminada. Então, é óbvio supor que Γ $(= \cup \Sigma_n)$, de fato não poderia se mostrar inconsistente. Mas Γ ainda corria o risco de se tornar inconsistente na dedução de $\sigma \wedge \neg\sigma$. Para tanto bastaria que houvesse uma regra de inferência inconsistente no sistema formal dedutivo da LRP. Mas o teorema da corretude mostrou que esse não é o caso. Portanto, Γ permanece consistente*].

2. Γ é maximal consistente. Suponha um conjunto Δ consistente e $\Gamma \subset \Delta$. Seja $\phi_n \in \Delta$. Daí, $\Sigma_n \cup \{\phi_n\}$ é consistente e $\Sigma_{n+1} = \Sigma_n \cup \{\phi_n\}$.

Assim, $\phi_n \in \Gamma$ (pois $\Gamma = \cup \Sigma_n$), e desse modo $\Gamma = \Delta$. [*Pois qualquer ϕ_i escolhida de Δ também pertence a Γ. Por igualdade de conjuntos,* $\Gamma = \Delta$].

7.3 Compacidade de um Conjunto de Fbfs da LRP

A comprovação da existência de compacidade em um conjunto de fbfs Σ é um dos melhores resultados que podemos obter. Um conjunto de fbfs que goza de compacidade possui características especiais que podem ser exploradas, digamos, para se efetuar computações sobre o mesmo. A compacidade de um conjunto de fbfs, por exemplo, garante o mapeamento de todas as fbfs que o compõe; de posse dessa característica pode-se, por exemplo, tomar uma determinada fbf τ e perguntar se τ é conseqüência tautológica de Σ ($\Sigma \models \tau$). As fbfs de Σ podem ir sendo tomadas uma a uma, e através de uma atribuição de verdade se verificar a satisfatibilidade de expressões do tipo $\{\sigma_1, \sigma_2, \sigma_3\} \models \tau$. O teorema da compacidade garante que haverá uma resposta positiva [**[BRA 74]**, **[DAV 83]**, **[SOM 88]**] toda vez que $\Sigma \models \tau$.

Definição 7.3.1 - Um conjunto Σ de fbfs da LRP é dito satisfatível se e somente se existe uma atribuição de verdade ν que satisfaz todo membro de Σ.

Definição 7.3.2 - Um conjunto Σ de fbfs da LRP é dito finitamente satisfatível se e somente se todo subconjunto finito de Σ é satisfatível.

Teorema 7.3.1 - [Compacidade] Qualquer conjunto Σ de fbfs da LRP é satisfatível se e somente se todo subconjunto finito de Σ também o for.

O teorema contempla três aspectos de Σ: (1) Se Σ é satisfatível, logo é automaticamente finitamente satisfatível. (2) Se Σ é finito, é também finitamente satisfatível pois é subconjunto de si mesmo. (3) Se Σ é infinito e finitamente satisfatível, então é satisfatível.

[*A idéia da primeira parte da prova é tomar Σ que é finitamente satisfatível, expandir até Δ que é finitamente satisfatível e mostrar que cada subconjunto de Δ é satisfatível. Daí, tal propriedade vale para Σ, pois ambos (Δ e Σ) têm a mesma natureza (finitamente satisfatível)*]

Demonstração: ($\Rightarrow$) Considere Σ ser finitamente satisfatível. Vamos expandir Σ até um conjunto maximal Δ. Seja $\phi_1, \phi_2, \ldots, \phi_n, \ldots$ uma enumeração fixa das fbfs da LRP. Vamos definir Δ por recursão sobre os naturais do seguinte modo:

$$\Delta_0 = \Sigma.$$

$$\Delta_1 = \begin{cases} \Delta_0 \cup \phi_1, & \text{se esta expressão for finitamente satisfatível,} \\ \Delta_0 \cup \neg\phi_1, & \text{senão.} \end{cases}$$

$$\vdots$$

$$\Delta_{n+1} = \begin{cases} \Delta_n \cup \phi_{n+1}, & \text{se esta expressão for finitamente satisfatível,} \\ \Delta_n \cup \neg\phi_{n+1}, & \text{senão.} \end{cases}$$

Assim, cada Δ_n é finitamente satisfatível.

Prova: Seja $\Delta = \Sigma \cup \phi$ e ν uma atribuição de verdade que satisfaz Δ.

$\bar{\nu}(\Delta) = \bar{\nu}(\Sigma \cup \phi) = \bar{\nu}(\Sigma) + \bar{\nu}(\phi)$. Se Δ é finitamente satisfatível, $\bar{\nu}(\Sigma) = \mathbf{V}$ e $\bar{\nu}(\phi) = \mathbf{V}$. Se $\bar{\nu}(\phi) = \mathbf{F}$, então $\Delta = \Sigma \cup \neg\phi$ é finitamente satisfatível, pois $\bar{\nu}(\Sigma) = \mathbf{V}$ e

$\overline{\nu}(\neg\phi) = \mathbf{V}$.

Considere o limite de todos os Δ_n como sendo $\Delta = \cup \Delta_n$. Daí, fica claro o seguinte: (1) $\Sigma \subseteq \Delta$. (2) Para qualquer fbf ϕ, ou $\phi \in \Delta$ ou $\neg\phi \in \Delta$, nunca ambos. Portanto, qualquer atribuição de verdade ν para Δ, ou $\overline{\nu}(\phi) = \mathbf{V}$ ou $\overline{\nu}(\neg\phi) = \mathbf{V}$. (3) Δ é finitamente satisfatível, pois qualquer subconjunto finito de Δ já é um subconjunto finito de algum Δ_n, sendo portanto satisfatível.

($\Leftarrow$) Considere uma atribuição de verdade ν para todos os símbolos sentenciais da LRP atomicamente ligados pela relação "é esquema para", de modo que, para qualquer $\alpha_i = a_i \Vert\!\rightarrow A_i$, $\nu(\alpha_i) = \mathbf{V}$ se e somente se $\alpha_i \in \Delta$. Então, para qualquer fbf ϕ, afirmamos que ν satisfaz ϕ se e somente se $\phi \in \Delta$.

Fazendo uso de indução sobre os conectivos sentenciais de ϕ, temos:

(1) $\phi = \neg\alpha$		<u>$\neg\alpha$ é ν-verdade</u>	$[\overline{\nu}(\neg\alpha) = \mathbf{V}]$,
	sse	α **não** é ν-verdade,	
	sse	$\alpha \notin \Delta$,	
	sse	$\neg\alpha \in \Delta$.	
(2) $\phi = \alpha \curlywedge \beta$		<u>$\alpha \curlywedge \beta$ é ν-verdade</u>,	
	sse	α é ν-verdade e β é ν-verdade,	
	sse	$\alpha \in \Delta$ e $\beta \in \Delta$,	
	sse	$\alpha \curlywedge \beta \in \Delta$.	

(3) $\phi = \alpha \curlyvee \beta$ $\underline{\alpha \curlyvee \beta \text{ é } \nu\text{-verdade}}$,

sse α é ν-verdade ou β é ν-verdade,

sse $\alpha \in \Delta$ ou $\beta \in \Delta$,

sse $\alpha \curlyvee \beta \in \Delta$.

(4) $\phi = \alpha \Rightarrow \beta$ $\underline{\alpha \Rightarrow \beta \text{ é } \nu\text{-verdade}}$,

sse $\neg\alpha$ é ν-verdade ou β é ν-verdade,

sse α não é ν-verdade ou β é ν-verdade,

sse $\alpha \notin \Delta$ ou $\beta \in \Delta$,

sse $\neg\alpha \in \Delta$ ou $\beta \in \Delta$,

sse $\alpha \Rightarrow \beta \in \Delta$.

Como $\Sigma \subseteq \Delta$ e ν satisfaz cada parte finita de Δ, então $\nu(\Sigma) = V$ e portanto Σ é satisfatível.

7.4 Completude do Sistema Formal Dedudivo da LRP

Dizer que um sistema de regras de inferência é *completo* é dizer que com seu auxilio pode-se deduzir, a partir de qualquer conjunto de fbfs dado, qualquer conseqüência daquele conjunto. Assim, para mostrar que o Sistema Formal Dedutivo da LRP (baseado em Ω e nas regras de inferência Rf_{MP}, Rf_{MPII}, $Rf_{\rightarrow}$, $Rf_{\wedge}$ e $Rf_{\vee}$ é completo deveremos mostrar que para qualquer fbf ϕ e conjunto de fbfs Γ, se ϕ é conseqüência de Γ, então ϕ é dedutível de Γ.

A completude de um sistema formal dedutivo é uma poderosa ferramenta de síntese a disposição do raciocínio abdutivo. A História da Ciência está repleta de ocasiões nas quais cientistas estavam de posse de um *conjunto de fatos* e de *alguma(s) conseqüência(s)* advinda(s) de tais *fatos*, sem no entanto estarem de posse do caminho completo que conduzia dos *fatos* até a(s) *conseqüência(s)*, isto é, não possuíam a *dedução* de tal(is) *conseqüência(s)*. A garantia de completude, portanto, garante que a dedução existe; garante a existência de uma prova para um fato desde que este fato seja conseqüência lógica de outros.

Antes de entrarmos na definição dos termos que nos auxiliarão no desenvolvimento do teorema da completude para a LRP, gostaríamos de acrescentar um breve comentário no que diz respeito a consistência de um conjunto de fbfs. Na realidade queremos estabelecer uma ligeira diferença entre as situações de consistência, isto é, um conjunto Γ de fbfs pode ser considerado *originariamente* consistente desde que não contenha dentro de si quaisquer fbfs contraditórias; e um conjunto Γ de fbfs pode ser consistente *sob uma sucessão de operações de dedução*, desde que o sistema formal dedutivo operando sobre Γ não gere fbfs contraditórias.

O caminho a ser percorrido daqui em diante baseia-se no fato de que um sistema formal dedutivo correto operando sobre um conjunto maximal consistente e compacto é um sistema formal dedutivo completo.

Definição 7.4.1 - Um conjunto de fbfs Γ é *consistente com respeito à dedutibilidade (d-consistente)* sse a fbf $(S \wedge \neg S)$ não é dedutível de Γ ($\Gamma \nvdash \bot$).

É interessante nos apercebermos do fato de que d-consistência e dedutibilidade estão intimamente relacionadas.

Teorema 7.4.1 - Para qualquer fbf ϕ e conjunto de fbfs Γ, $\Gamma \vdash \phi$ sse $\Gamma \cup \{\neg\phi\}$ não é d-consistente.

<u>*Prova:*</u> ($\Rightarrow$) Suponha que na i-ésima linha de uma dedução a partir de Γ tenhamos ϕ. Na linha i+1 introduzimos $\neg\phi$ como premissa; daí a partir de

$\Gamma \cup \{\neg\phi\}$, temos na linha i+2, $(\phi \wedge \neg\phi) \vdash (S \wedge \neg S)$, de modo que $\Gamma \cup \{\neg\phi\}$ não é d-consistente.

(<≡) Pela definição de d-consistência, temos $\Gamma \cup \{\neg\phi\} \vdash \square$; donde pelo teorema da dedução, temos $\Gamma \vdash \neg\phi \Rightarrow \square$ ≡> $\Gamma \vdash \neg\neg\phi \vee \square$ ≡> $\Gamma \vdash \phi$].

Definição 7.4.2 - Um conjunto de fbfs Γ é *d-consistente maximal* sse Γ não é subconjunto próprio de qualquer conjunto d-consistente Δ $(\Gamma \subsetneq \Delta)$.

Assim, Γ é d-consistente maximal sse Γ é d-consistente, *perdendo* essa propriedade se lhe for *acrescentada* qualquer fbf que *não lhe pertencia*.

Teorema 7.4.2 - Seja ϕ e ψ fbfs quaisquer de $\mathcal{L}_3$ (a linguagem na qual expressamos as idéias da LRP), e Δ qualquer conjunto d-consistente maximal. Δ goza das seguintes propriedades:

(1) $\phi \in \Delta$ sse $\neg\phi \notin \Delta$;

(2) $\phi \in \Delta$ sse ϕ é dedutivel de Δ;

(3) $(\phi \wedge \psi) \in \Delta$ sse $\phi \in \Delta$ e $\psi \in \Delta$;

(4) $(\phi \vee \psi) \in \Delta$ sse $\phi \in \Delta$ ou $\psi \in \Delta$, e

(5) $(\phi \Rightarrow \psi) \in \Delta$ sse $\phi \notin \Delta$ ou $\psi \in \Delta$.

Demonstração de (1) - Suponha $\phi \in \Delta$ e $\neg\phi \in \Delta$. Então, $\Delta \vdash \phi$ e $\Delta \vdash \neg\phi$ e, assim, $\Delta \vdash (S \wedge \neg S)$, o que contraria a hipótese de Δ ser d-consistente. Suponha agora que nem $\phi \in \Delta$ nem $\neg\phi \in \Delta$. Como Δ é d-consistente maximal, então, $\Delta \cup \{\phi\}$ e $\Delta \cup \{\neg\phi\}$ não são d-consistentes. Daí tanto $\Delta \vdash \phi$ como $\Delta \vdash \neg\phi$, contrariando a hipótese. Assim, só um elemento do par ϕ, $\neg\phi$ pertence a Δ.

<u>*Demonstração de (2)*</u> - Se $\phi \in \Delta$, obviamente $\Delta \vdash \phi$. Suponha que $\phi \notin \Delta$, então por (1) $\neg\phi \in \Delta$, portanto $\Delta \vdash \neg\phi$. Assim, como Δ é d-consistente, $\Delta \nvdash \phi$. Por outro lado, se $\Delta \vdash \phi$, então $\Delta \nvdash \neg\phi$ (por consistência). Daí, $\neg\phi \notin \Delta$, e finalmente, por maximalidade, $\phi \in \Delta$.

<u>*Demonstração de (3)*</u> - Desde que $(\phi \wedge \psi) \in \Delta$, então por (2), $\Delta \vdash (\phi \wedge \psi)$, e por dedução $\Delta \vdash \phi$ e $\Delta \vdash \psi$. Dai, por (2) $\phi \in \Delta$ e $\psi \in \Delta$. Por outro lado, se $\phi \in \Delta$ e $\psi \in \Delta$, por (2) temos $\Delta \vdash \phi$ e $\Delta \vdash \psi$. Daí, por corretude $\Delta \models \phi$ e $\Delta \models \psi$, e por conseqüência tautológica $\Delta \models (\phi \wedge \psi)$, portanto $\phi \wedge \psi \in \Delta$.

O mesmo raciocinio é usado para provar (4) e (5).

Teorema 7.4.3 - [Completude] Seja ϕ uma fbf de $\mathcal{L}_3$ e Γ um conjunto de fbfs da LRP. Desde que $\Gamma \models \phi$, então o Sistema Formal Dedutivo da LRP é completo. ($\Gamma \models \phi \Rightarrow \Gamma \vdash \phi$)

A idéia da prova é usar os conceitos de d-consistência e maximalidade definidos anteriormente. Por isso faremos uso dos seguintes lemas:

Lema 7.4.1 - Para qualquer conjunto de fbfs Γ da LRP, se Γ é d-consistente, então Γ é consistente.

De posse do lema 1, podemos firmar a completude da seguinte maneira: Seja ϕ uma fbf de $\mathcal{L}_3$ e Γ um conjunto de fbfs da LRP, supondo-se que $\Gamma \models \phi$. Então, $\Gamma \cup \{\neg\phi\}$ não é consistente. Logo, em virtude desse lema, $\Gamma \cup \{\neg\phi\}$ não é d-consistente, e por isso, $\Gamma \vdash \phi$. Assim, nosso sistema formal dedutivo é completo.

Como não sabemos quantas nem quais são as fbfs ϕ_i de $\mathcal{L}_3$ que são conseqüências de Γ, precisamos garantir a d-consistência do *conjunto original* unido às conseqüências dele próprio - de acordo com a necessidade do lema 7.4.1. Uma maneira de garantir a d-consistência é *construindo* tal conjunto. Assim, entra o conceito de maximalidade consistente.

Lema 7.4.2 - Para qualquer conjunto Γ de fbfs da LRP, se Γ satisfaz a hipótese do lema 7.4.1, haverá um conjunto de fbfs Δ que inclui Γ e que é d-consistente maximal.

<u>*Demonstração do* lema 7.4.2</u> : (Supondo que Γ satisfaça a hipótese do lema 7.4.1).

1) Faremos uso do fato, admitido sem demonstração, de que todas as fbfs de $\mathcal{L}_3$ podem ser ordenadas em uma lista da forma $\phi_1, \phi_2, \phi_3, \ldots, \phi_n, \ldots$ de modo que cada fbf de $\mathcal{L}_3$ ocorra pelo menos uma vez nessa lista [O teorema 0B e o teorema da enumerabilidade em [**END 72**], nos dão a idéia de como tal ordenação pode ser feita].

2) De posse da lista acima podemos *construir* uma infinidade de conjuntos $\Delta_0, \Delta_1, \Delta_2, \ldots, \Delta_n, \ldots$ do seguinte modo:

$$\Delta_0 = \Gamma;$$

$$\Delta_1 = \begin{cases} \Delta_0 \cup \{\phi_1\}, & \text{se a união for d-consistente} \\ \Delta_0, & \text{caso contrário.} \end{cases}$$

E, de modo geral para cada inteiro positivo n,

$$\Delta_n = \begin{cases} \Delta_{n-1} \cup \{\phi_n\}, & \text{se a união for d-consistente} \\ \Delta_{n-1}, & \text{caso contrário.} \end{cases}$$

Seja Δ a união de todos esses Δ_i. Assim, uma fbf ϕ é um elemento de Δ sse é um elemento de pelo menos um desses Δ_i. Mostraremos que Δ, que inclui Γ - segundo os reclames do lema 7.4.2 - é d-consistente maximal.

3) Δ *é d-consistente*. (i) Note que cada Δ_i é d-consistente por construção, já que Γ é d-consistente - pela hipótese do lema 7.4.1. (ii) Por outro lado, suponha que $\Delta \vdash (S \wedge \neg S)$. Assim, $\Delta' \vdash (S \wedge \neg S)$, Δ' subconjunto finito de Δ - devido o caráter finitário de uma dedução - e que por sua vez $\Delta' \subset \Delta_j$, para algum *j*. Logo, $\Delta_j \vdash (S \wedge \neg S)$, contrariamente a (i).

4) Δ é *d-consistente maximal*. Seja ϕ uma fbf. Suponha que $\phi \notin \Delta$. Mas $\phi = \phi_i$, para algum *i*. Como $\phi \notin \Delta$, então $\phi \notin \Delta_i$. Assim, $\Delta_{i-1} \cup \{\phi\}$ não é d-consistente e, $\Delta \cup \{\phi\}$ não é d-consistente - Δ é uma união de conjuntos d-consistentes encaixantes.

Resta-nos agora provar o lema 7.4.1. Para tanto faremos uso de outro lema.

Lema 7.4.3 - Todo conjunto de fbfs d-consistente é consistente.

<u>*Demonstração do*</u> <u>lema 7.4.3</u>: Seja Δ um conjunto de fbfs d-consistente maximal. Na realidade Δ é o conjunto de conseqüências de Γ. Assim, consoante o conceito de conseqüência tautológica, cada fbf ϕ de $\mathcal{L}_3$ será acrescentada a Δ se $\bar{v}(\phi) = V$, de acordo com a função de atribuição de verdade definida na seção 6.3. Assim, podemos redefinir Δ da seguinte maneira

$$\Delta = \{\phi : \bar{v}(\phi) = V\};$$

ou em outras palavras: (a) $\overline{v}(\phi) = V$ sse $\phi \in \Delta$. Vamos fazer uso de (a) para provar o lema 7.4.3.

Considere ϕ, ψ e χ fbf's da LRP, com ψ e χ com índices[1] menores que o de ϕ. Provaremos o lema 7.4.3 supondo que existe pelo menos um $\phi \notin \Delta$, e chegaremos a um absurdo. Faremos a prova por indução sobre o menor índice de possivel de ϕ.

1) $\phi = \neg\psi$. $\neg\psi$ é $\overline{v}$ - *verdade* $[\overline{v}(\neg\psi) = V]$

sse ψ não é $\overline{v}$ - *verdade*;

sse $\psi \notin \Delta$;

sse $\neg\psi \in \Delta$. [por (1)]

2) $\phi = (\psi \wedge \chi)$. $(\psi \wedge \chi)$ é $\overline{v}$ - *verdade*

sse ψ é $\overline{v}$ - *verdade* e χ é $\overline{v}$ - *verdade*;

sse $\psi \in \Delta$ e $\chi \in \Delta$;

sse $(\psi \wedge \chi) \in \Delta$. [por (2)]

3) $\phi = (\psi \vee \chi)$. $(\psi \vee \chi)$ é $\overline{v}$ - *verdade*

sse ψ é $\overline{v}$ - *verdade* ou χ é $\overline{v}$ - *verdade*;

sse $\psi \in \Delta$ ou $\chi \in \Delta$;

sse $(\psi \vee \chi) \in \Delta$. [por (3)]

[1] Define-se o *índice* de uma fbf proposicional como a quantidade de conectivos sentenciais da mesma.

4) $\phi = (\psi \Rightarrow \chi)$. $(\psi \Rightarrow \chi)$ é $\bar{v}$ - *verdade*

sse $\neg\psi$ é $\bar{v}$ - *verdade* ou χ é $\bar{v}$ - *verdade*;

sse ψ não é $\bar{v}$ - *verdade* ou χ é $\bar{v}$ -*verdade*;

sse $\psi \notin \Delta$ ou $\chi \in \Delta$;

sse $(\psi \Rightarrow \chi) \in \Delta$. [por (4)]

Assim, chegamos ao absurdo de que $\phi \notin \Delta$ deveria ser um dos casos 1) - 4), não sendo, entretanto, nenhum deles. Logo, Δ, que é d-consistente maximal, é consistente.

A idéia por trás da prova do lema 7.4.3 é que jamais se daria o caso de que algumas fbfs de Δ, componentes de ϕ, pertencessem a Δ e ϕ não pertencesse a Δ. O que seria um desastre do ponto de vista dedutivo. Assim, ficou estabelecido que tudo que é conseqüência de Δ pertence a Δ e é passível de dedução. Essa é a idéia da completude.

Uma maneira mais acessível de se verificar parcialmente a completude do Sistema Formal Dedutivo da LRP é a seguinte:

Considere $\Gamma = \{a \Vert\!\rightarrow A,\ b \Vert\!\rightarrow B,\ c \Vert\!\rightarrow C\}$ e $\Delta = \{a \wedge b \Vert\!\rightarrow A \wedge B,\ a \vee b \Vert\!\rightarrow A \vee B,\ a \rightarrow b \Vert\!\rightarrow A \rightarrow B,\ (a \wedge b) \wedge c \Vert\!\rightarrow (A \wedge B) \wedge C,\ a \rightarrow (b \rightarrow a) \Vert\!\rightarrow A \rightarrow (B \rightarrow A)\}$. Qualquer fbf de Δ é conseqüência tautológica de Γ e pode ser deduzida de Γ por alguma regra de inferência Rf_{MP}, Rf_{MPII}, $Rf_{\rightarrow}$, $Rf_{\wedge}$ e $Rf_{\vee}$ junto com o conjunto de axiomas Ω.

Capítulo 8

Esboço para uma Lógica de Primeira Ordem de Resolução de Problemas

Neste capitulo apresentamos a LRP como uma possivel base para o desenvolvimento de uma lógica de primeira ordem (LPO) visando a resolução de problemas. A intenção é apenas apresentar os elementos básicos para a definição da lógica. Mesmo assim, discorreremos um pouco a respeito das potencialidades de uma LPO. Também introduziremos algumas noções sobre uma interpretação para uma LPO para que haja uma rápida comparação com as funções de atribuição de verdade inerentes às lógicas sentenciais. O objetivo desse capitulo é apenas sugerir que as idéias sobre o mundo da resolução de problemas, de acordo com a TGP, podem ser descritas de modo potencialmente mais rico através de uma LPO para problemas, isto é, uma Linguagem de Primeira Ordem de Resolução de Problemas (LPORP).

8.1 As Linguagens de Primeira Ordem e a Formalização do Conhecimento Declarativo

Muito do conhecimento a respeito de um mundo é descritivo, e pode ser expresso de uma forma *declarativa*. A formalização do conhecimento em forma declarativa começa com uma *conceitualização*. Isto inclui os objetos do mundo, cuja existência é real ou hipotética, e o inter-relacionamento entre eles. Um *objeto* pode ser qualquer coisa sobre a qual queremos dizer algo. O conjunto de objetos considerados é chamado de *universo do discurso* **[GEN 87]**.

Um mundo pode ser conceitualizado de diversas maneiras sem haver, entretanto, uma correspondência obrigatória entre os objetos e relações em uma conceitualização com os objetos e relações em outra. A questão crucial a esse respeito é que dependendo da conceitualização adotada em alguns casos, torna-se impossível expressar certo tipo de conhecimento. Em outros, embora não seja impossível, torna-se muito difícil.

Dada uma conceitualização do mundo, podemos começar a formalizar conhecimento como sentenças numa linguagem apropriada àquela conceitualização. Uma maneira de fazer isto é utilizando uma *linguagem de primeira ordem* (LPO).

A principal característica de uma LPO é que a disponibilidade de quantificadores e variáveis aumenta nossa flexibilidade. O quantificador $\forall$ permite-nos estabelecer fatos sobre todos os objetos no nosso universo do discurso sem que no entanto tenhamos de enumerá-los. O quantificador $\exists$ permite-nos asseverar a existência de um objeto com certas propriedades sem no entanto identificar o objeto. Uma *variável* é usada para expressar propriedades de objetos no universo do discurso sem nomeá-los explicitamente.

A partir desse entendimento podemos definir o alfabeto e a linguagem para a LPORP. A LPORP se constitui numa extensão da LRP.

8.2 Extensão da Linguagem da LRP

Embora a extensão do alfabeto $\mathcal{A}_3$ só se dê a nível de inclusão dos quantificadores universal e existencial, mesmo assim iremos especificar todo o alfabeto $\mathcal{A}_4$ para manter o formalismo.

Seja $\mathcal{L}_4$ a linguagem utilizada para expressar as idéias da LPORP. Definiremos o alfabeto $\mathcal{A}_4$, no qual se baseia esta linguagem, da seguinte maneira:

1) *Símbolos lógicos*:
   - i) *Símbolos de pontuação*: Parênteses, à esquerda e à direita: (,);
   - ii) *Símbolos conectivos sentenciais*: $\neg$, $\wedge$, $\vee$ e $\Rightarrow$, para expressar, respectivamente, o sentido de "não", "e", "ou" e "implica" lógicos;
   - iii) *Variáveis*: As letras maiúsculas A, B, C, ..., Y, Z para expressar as sentenças da LEP, e as minúsculas a, b, c, ..., y, z para expressar as sentenças da LES.

2) *Parâmetros*:
   - i) *Símbolos Quantificadores*: Universal e Existencial: $\forall$, $\exists$
   - i) *Símbolo predicativo*: $\Vert\!\!\rightarrow$, para indicar a relação "é esquema para".

As fórmulas atômicas da LPORP são expressões do tipo a $\Vert\!\!\rightarrow$ A e sua negação significando respectivamente "a é esquema para A" e "a não é esquema para A".

As fórmulas bem formadas nesta linguagem são definidas como segue:

i) As fórmulas atômicas são fórmulas bem formadas;

ii) Se α e β são fórmulas bem formadas, então $\neg\alpha$, $\alpha \wedge \beta$, $\alpha \vee \beta$, $\alpha \Rightarrow \beta$, $\forall a \forall A\, \alpha$, $\forall a\, \exists A\, \alpha$, $\exists a \forall A\, \alpha$, $\exists a\, \exists A\, \alpha$ também o são;

iii) As fórmulas bem formadas são somente obtidas de i e ii acima.

8.3 Interpretação e Verdade

O objetivo principal da lógica é dizer alguma coisa sobre o mundo imaginável. Conseqüentemente, devemos associar os símbolos e suas relações a coisas mais tangíveis **[WIN 84]**. Temos um conjunto de sentenças e uma conceitualização do mundo, e associamos os símbolos usados nas sentenças com os objetos e relações de nossa conceitualização. Nós avaliamos a verdade das sentenças de acordo com essa associação, dizendo que uma sentença é verdade se e somente se ela acuradamente descreve o mundo de acordo com nossa conceitualização **[GEN 87]**. Uma interpretação é um mapeamento entre os elementos da linguagem e os elementos da conceitualização. Objetos em algum domínio correspondem a símbolos objeto na lógica. Relações em algum domínio correspondem a predicados na lógica. Quando uma relação se mantém com respeito a alguns objetos, o predicado correspondente é verdadeiro quando aplicada aos símbolos objetos correspondentes.

Ao fazer essa afirmação assumimos a perspectiva do observador na figura abaixo.

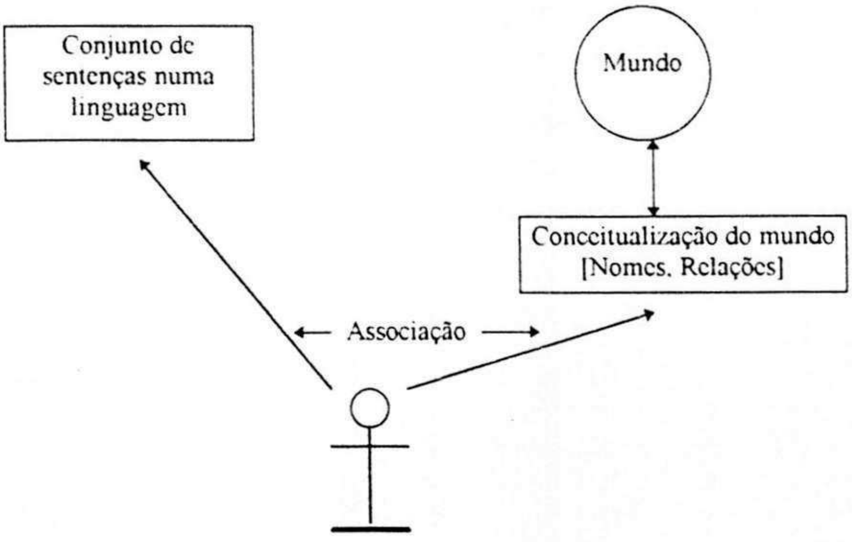

Fig. 8.3.1 - Visão esquemática de Interpretação

Uma interpretação ***I*** é um mapeamento entre elementos da linguagem e elementos da conceitualização. Vamos representar este mapeamento pela função $I(\sigma)$, onde σ é um elemento da linguagem, e abreviá-lo para σ^I. O nosso universo de discurso será representado por $|I|$. Para I ser uma interpretação, deve satisfazer as seguintes propriedades:

i) I associa ao símbolo quantificador $\forall$ o conjunto não-vazio $\vert I \vert$.
ii) Se σ é um objeto, então $\sigma^I \in \vert I \vert$.
iii) Se ρ é uma relação de aridade n, então $\rho^I \subseteq \vert I \vert^n$.

Agora precisamos de uma definição matemática de "σ é verdade em I", isto é, queremos tomar nossa noção intuitiva de "σ é verdade em I", e fazê-la parte da matemática. Vamos ligar o conceito de *verdade* ao de *satisfação* **[END 72]**.

Vamos representar "σ é verdade em I" por $\models_I \sigma$. Seja σ uma fbf da nossa linguagem, I uma interpretação para a linguagem, e $s: V \rightarrow |I|$ uma função atribuição definida do conjunto V de todas as variáveis no universo de discurso $|I|$.

Definição 8.3.1 - Dizemos que I *satisfaz* σ com a função de atribuição s, e representamos por $\models_I \sigma\ [s]$, se e somente se a tradução de σ, onde a variável x é traduzida por $s(x)$, determinada por I é verdade **[END 72]**.

Vamos definir uma função atribuição $\bar{s} : T \rightarrow |I|$, definida do conjunto T de todos os termos no universo de discurso $|I|$, como uma extensão de s. Vamos definir a aplicação de $\bar{s}$ para:

1) *Termos*

i) Se x é variável, $\bar{s}(x) = s(x)$.

2) *Fórmulas atômicas*

i) Para um predicado $\|\!\rightarrow$ de aridade 2 e termos a, A,

$$\models_I (\mathrm{a} \|\!\rightarrow \mathrm{A})\ [s] \quad \text{se e somente se} \quad \langle \bar{s}(\mathrm{a}), \bar{s}(\mathrm{A}) \rangle \in \|\!\rightarrow^I.$$

3) *Outras fbfs*

i) $\models_I \neg\sigma\ [s]$ se e somente se $\not\models_I \sigma\ [s]$.

ii) $\models_I (\sigma \wedge \varphi)\ [s]$ se e somente se $\models_I \sigma\ [s]$ e $\models_I \varphi\ [s]$.

iii) $\models_I (\sigma \vee \varphi)\ [s]$ se e somente se $\models_I \sigma\ [s]$ ou $\models_I \varphi\ [s]$.

iv) $\models_I (\sigma \Rightarrow \varphi)\ [s]$ se e somente se $\not\models_I \sigma\ [s]$ ou $\models_I \varphi\ [s]$.

v) $\models_I \forall x \forall X \sigma\ [s]$ se e só se para todo $\mathbf{d}$, $\mathbf{D} \in |I|$, $\models_I \sigma\ [s(x\backslash\mathbf{d})s(X\backslash\mathbf{D})]$.

vi) $\models_I \forall x \exists X \sigma\ [s]$ sse para todo $\mathbf{d}$ e algum $\mathbf{D} \in |I|$, $\models_I \sigma\ [s(x\backslash\mathbf{d})s(X\backslash\mathbf{D})]$.

vii) $\models_I \exists x \forall X \sigma\ [s]$ sse para algum $\mathbf{d}$ e todo $\mathbf{D} \in |I|$, $\models_I \sigma\ [s(x\backslash\mathbf{d})s(X\backslash\mathbf{D})]$.

viii) $\models_I \exists x \exists X\ \sigma\ [s]$ sse para algum $\mathbf{d}$, $\mathbf{D} \in |I|$, $\models_I \sigma\ [s(x\backslash\mathbf{d})s(X\backslash\mathbf{D})]$.

A função $s(x\backslash\mathbf{d})$ tem a mesma função de s exceto que no lugar da variável x ela assume o valor $\mathbf{d}$. Idem para $\mathbf{D}$.

8.4 Uma Interpretação para a LPORP

Vamos definir uma interpretação I para a LPORP do seguinte modo:

i) O universo de discurso ou domínio $|I|$ será o conjunto dos esquemas de solução $\{\mathbf{a}, \mathbf{b}, \mathbf{c}, \ldots, \mathbf{y}, \mathbf{z}\}$ unido com o conjunto de esquemas de problema $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \ldots, \mathbf{Y}, \mathbf{Z}\}$;

ii) $a^I = \mathbf{a}$, $b^I = \mathbf{b}, \ldots, z^I = \mathbf{z}$;

iii) $A^I = \mathbf{A}$, $B^I = \mathbf{B}, \ldots, Z^I = \mathbf{Z}$;

iv) $\|\!\rightarrow^I$ - relação binária "é esquema para" que liga o esquema de solução **a** ao esquema de problema **A**. Quando **a** é realizado em **A**, isto é, **a** é um esquema funcional existente definido de **D** em **R**, domínios do problema **A**, e **a** $\|\!-$ **A**, a relação binária é verdadeira.

A função de atribuição **s** atribui aos símbolos variáveis a, b, ...,z, A, B, ...,Z os respectivos correspondentes funcionais **a**, **b**, ..., **z**, e problemas **A**, **B**, ..., **Z** .

Definição 8.4.1 - Uma interpretação I satisfaz uma fbf σ da LPORP, onde as variáveis x e X são traduzidas pelos respectivos funcionais de solução e problemas através da função de atribuição s: $V \rightarrow |I|$ ($s(x) =$ **a**, $s(X) =$ **A**, etc), se somente se, **a** realizado em **A** implica que **a** é solução para **A** [vide Definição 1.2.3]. Representamos essa noção de satisfação por $||\Rightarrow_I \sigma\, [s]$ **[LOP 85]**.

Definição 8.4.2 - Uma fbf σ da LPORP é satisfatível se e somente se existe uma interpretação I com uma função de atribuição s: $V \rightarrow |I|$ que satisfaz σ.

Definição 8.4.3 - Um conjunto Σ de fbfs da LPORP é satisfatível se e somente se existe uma interpretação I com uma função de atribuição s: $V \rightarrow |I|$ que satisfaz cada fbf de Σ.

Definição 8.4.4 - Uma interpretação I é *modelo* para uma fbf σ da LPORP se e somente se toda função de atribuição s: $V \rightarrow |I|$ satisfaz σ.

Definição 8.4.5 - Uma interpretação I é modelo para um conjunto Σ de fbfs da LPORP se e somente se toda função de atribuição s: $V \rightarrow |I|$ satisfaz cada fbf de Σ.

Numa lógica de primeira ordem o equivalente às tautologias são as fórmulas válidas.

Definição 8.3.5 - Uma fbf σ da LPORP é *válida* se e somente se toda interpretação I e toda função de atribuição s: $V \rightarrow |I|$ satisfaz σ. Representamos a noção "σ é válida" por $||\Rightarrow \sigma$.

A verificação de validade de uma fbf da LPORP, de acordo com a definição, leva a uma situação muito complicada caso nos deparemos com um universo de discurso de dimensão infinita. Mas a noção de validade torna-se equivalente à noção de *dedutibilidade* cuja definição está mais próxima da noção de finitude. Usando esta equivalência pode-se mostrar, *sob algumas suposições razoáveis*, que o conjunto de fórmulas válidas é efetivamente enumerável. O procedimento de enumeração efetiva produz uma caracterização mais concreta do conjunto de fórmulas válidas **[END 72]**. Nesse caso a LPORP necessita de um sistema formal dedutivo que possa supri-la da característica de dedutibilidade e completude.

8.5 Considerações sobre um Sistema Formal Dedutivo para a LPORP

Um conceito de grande importância para este cálculo dedutivo é o de *adequação*, dado na Definição 6.3.1, pois, o mesmo traz dentro de si o conceito de teorema advindo do Teorema 4.5 de **[LOP 85]**, que por sua vez incorpora a noção de dedução das definições 6.1.1 e 6.1.2.

Cremos também que o conjunto de axiomas deste cálculo deva incorporar a generalização dos resultados obtidos no Teorema 4.4 de **[LOP 85]**, e algum ou alguns axiomas de generalização com os mesmos objetivos daquele da lógica de primeira ordem clássica.

Quanto às regras de inferência, além da clássica Modus Ponens, se conseguirmos os mesmos resultados da $Rf_{\rightarrow}$ para fbfs do tipo ∀x∀X σ, cremos que podemos contar também com as $Rf_{\wedge}$ e $Rf_{\vee}$ para fbfs do mesmo tipo.

Capítulo 9

Conclusão

Antes de tecermos considerações a respeito do resultado geral deste trabalho, gostaríamos de apresentar algumas observações sobre o que consideramos ser o resultado de nosso esforço na composição desta dissertação, isto é, o que em cada capítulo consideramos como sendo nossa positiva contribuição.

No capítulo 2 apontamos o nosso esforço no sentido de colocar a TGP como uma contribuição positiva para a IA em geral, e em particular no contexto da resolução de problemas. Entendemos ser esse esforço pioneiro, visto que dos trabalhos acadêmicos desenvolvidos sobre a TGP nenhum outro se preocupou em explorar esse viés da questão.

Embora o capítulo 3 seja uma repetição dos elementos básicos da TGP, gostaríamos de apontar a demonstração do Teorema 3.3.1. Pela primeira vez é apresentada uma demonstração para tão importante teorema, cujo objetivo é estabelecer uma ligação entre a TGP e a lógica. Ainda nesse capítulo fizemos questão de ressalvar que o aspecto mais historicamente desprezado da questão *problema* - o sintático - foi a pedra fundamental para o desenvolvimento da TGP; mas apropriadamente a formalização da estrutura *problema*.

No capítulo 4 ressaltamos o estudo que empreendemos no esforço para entender as interrogações que envolviam a *negação* dentro da lógica. O objetivo foi justificar dentro da filosofia da lógica clássica toda a análise da *negação* feita em **[LOP 85]** através de outra abordagem.

O capítulo 5 se constitui uma síntese de **[LOP 85]**. O que quisemos ressaltar nesse capítulo foi a apresentação dos sistemas formais dedutivos da LEP e LES; ambos baseados em axiomas e regras de inferência conhecidos da lógica proposicional clássica. Nessa apresentação objetivamos mostrar a preocupação que havia na idealização e realização de tais lógicas, de modo que a composição da LRP culminasse também numa lógica cujo comportamento não destoasse daquele da lógica clássica, isto é, mostrar que desde as origens havia o compromisso de realizar um instrumento legítimo de inferência lógica.

No capítulo 6 ressaltamos dois pontos altos. O primeiro é a *Função de Atribuição de Verdade para a LRP*. O modo como tal função foi idealizada se constitui a contrapartida *semântica* para o fato de que a LRP foi *sintaticamente* concebida como a junção de duas lógicas. Por isso, a idéia de "*verdadeiro*" e "*falso*" envolvendo as proposições da LRP remete a questão para o cerne das estruturas problemas envolvidas; a resposta exige que a análise da questão seja feita usando-se os conceitos da TGP. O segundo ponto foi o desenvolvimento do *Novo Teorema da Dedução para a LRP*. Esse teorema vem de encontro ao desejo de Lopes e Acioly **[LOP 85]** conforme propuseram na conclusão de seu paper que motivou o desenvolvimento do trabalho que culminou com esta dissertação.

O capítulo 7 se constitui nossa principal contribuição para esse trabalho. Esse capítulo sempre foi visto por nós como um divisor de águas; no sentido de que os teoremas do capítulo 7 finalmente iria nos dizer se a LRP era, ou não, um instrumento legítimo e completo de inferência lógica. Os esquemas de demonstração dos quatro principais teoremas desse capítulo seguem aquilo que se acha na bibliografia. De moto próprio são algumas inclusões nas demonstrações, e a preocupação em comentar exaustivamente cada passo das mesmas. Fica também a nossa preocupação em esclarecer a importância para a LRP de cada aspecto positivo garantido por cada um dos teoremas. A confecção do capítulo em referência demandou uma exaustiva pesquisa bibliográfica, pois tínhamos o interesse de apontar para as possibilidades computacionais da LRP, e os aspectos computacionais das lógicas estudadas não são deixados bem claros nos textos consultados. Aqui valeram as conversas mantidas com o professor orientador de nosso trabalho.

O capítulo 8 se constitui um esforço de nossa parte no sentido de que fique como um incentivo para o desenvolvimento de trabalhos futuros. O objetivo é mostrar que uma lógica de ordem superior a da LRP é necessária para expressar muitas idéias envolvendo a interação de problemas, como por exemplo algumas generalizações e particularizações, que obviamente não são contempladas pela LRP por esta ser uma lógica de proposições. A composição de uma lógica de primeira ordem de resolução de problemas traz alguns desafios interessantes. Por exemplo, todas as possíveis expressões escritas com a sintaxe dessa linguagem realmente fazem sentido no mundo dos problemas (isto é, na TGP)? Quais problemas (P) e respectivas soluções (s) se adequam a uma expressão do tipo $\forall s \forall P\ \alpha$, onde α é uma fbf da LRP? Qual a influência que a classe a qual o problema pertence exerce sobre o significado da expressão? Dar-se-á o caso em que uma expressão sem significado numa determinada classe de problemas seja válida em outra classe?

Finalmente apresentamos nossas considerações gerais sobre o trabalho levado a efeito. De início faremos menção dos problemas que tínhamos em mãos, e depois iremos mencionando a superação dos mesmos.

Sendo a LRP a junção de duas lógicas positivas - devido aos já comentados entraves de ordem epistemológica envolvendo a *negação* nas lógicas de esquemas de problema e esquemas de solução - por si só já nos fazia sentir certos receios a respeito de se a LRP seria ou não "bem comportada" nos moldes da lógica proposicional clássica. A verdade é que, embora não estivesse explícito em **[LOP 85]**, o fato de a LRP ser composta pela junção das lógicas positivas LEP e LES, através da relação *"é esquema para"*, foi a grande chave para se superarem as dificuldades com relação à *negação*. Ademais, vale salientar, a inclusão das regras de inferência $Rf_{\rightarrow}$, $Rf_{\wedge}$ e $Rf_{\vee}$, embora tenha sido feita na intenção de não frustar os aspectos de corretude e completude, não nos dava plena certeza de que não seriam gerados alguns subprodutos indesejáveis que viessem perturbar o tão desejado "bom comportamento". Felizmente nossos receios não se confirmaram.

Cremos que com o desenvolvimento do Novo Teorema da Dedução para a LRP (seção 6.5) e os resultados do capitulo 7, os objetivos do nosso trabalho foram alcançados plenamente. Sendo uma lógica decidível, a computação das sentenças da LRP é uma tarefa perfeitamente exeqüível.

Como sugestão para trabalhos futuros, recomendamos a revisão de todo o capitulo 8 de modo que culmine com o desenvolvimento de uma lógica de primeira ordem com poder de expressão mais abrangente sobre o mundo dos problemas, nos moldes da TGP. Ressalvamos ainda que aquelas perguntas feitas em parágrafo anterior, comentando a cerca do capitulo 8, só poderão ser respondidas depois de alguns estudos efetuados na área da TGP propriamente dita. Isso, sem dúvida, também conduzirá a alguns trabalhos.

O grande objetivo de uma lógica de primeira ordem para resolução de problemas é a prova automática de teoremas **[YAS 71]**, **[BET 50]**, **[CHA 73]**; nesse caso o desenvolvimento de uma lógica de ordem superior (LPORP) deveria chegar até um teorema da forma normal prenex **[DAV 83]**, **[LIG 64]**, **[ROB 66]**. Até que ponto a *skolemização* de sentenças da LPORP será exeqüível é apenas uma das desafiantes tarefas sobre a qual o desenvolvedor de tal lógica deverá se debruçar **[DAV 83]**, **[END 72]**, **[ROB 65]**, **[YAS 71]**.

Por fim, depois de superados todos os problemas para o desenvolvimento da LPORP, resta(m) ainda algum(ns) trabalho(s) na área da engenharia de software: Tornar exeqüível em uma máquina computadora a tarefa de provar teoremas da LPORP - por exemplo, numa linguagem nos moldes do PROLOG **[FIT 87]**. Tal ou tais trabalhos terão que envolver concomitantemente conceitos da LPORP, da TGP e da matemática.

[ALM 90] ALMEIDA, A. P. **Especificação e Solubilidade de Classes de Problemas com Paradigmas e de Classes Nomeáveis**. Dissertação de Mestrado, Depto. de Informática/CCEN/UFPE, Recife-Pernambuco-Brasil, 1990.

[BAN 69] BANERJI, R. B. **Theory of Problem Solving: An Approach to Artificial Intelligence**. American Elsevier Publishing Company, Inc., New York. 1969.

[BAN 80] BANERJI, R. B. **Artificial Intelligence: A Theoretical Approach**. North-Holland/Elsevier, New York. 1980. 254p.

[BAR 75] BARNES, D. W & MACK, J. M. **An Algebraic Introduction To Mathematical Logic**. Springer-Verlag New York, Inc. 1975. 121p.

[BED 87] BEDREGAL, B. R. C. **Especificação de Problemas Solúveis por Decomposição via Análise da Intencionalidade**. Dissertação de Mestrado, Depto. de Informática/CCEN/UFPE, Recife-Pernambuco--Brasil, 1987.

[BEL 60] BELLMAN, R. **On the Concept of Problem and Problem Solving**. American Mathematics, 67. 1960.

[BET 50] BETH, E, W, **Les Fondements Logiques des Mathématiques**. Gauthier-Villars, Paris. 1950. 222p.

[BRA 74] BRAINERD, W. S. & LANDWEBER, L. H. **Theory of Computation**. John Wiley and Sons, Inc. 1974. 336p.

[CAS 87] CASANOVA, M. A.; GIORNO, F. A. C. & FURTADO, A. L. **Programação em Lógica e a Linguagem PROLOG**. Edgard Blücher, São Paulo. 1987.

[CHA 73] CHANG, C.-L. & LEE, R. C.-T. **Symbolic Logic and Mechanical Theorem Proving**. Academic Press, Inc. 1973. 331p.

[CHA 77] CHANG, C.-L. & KEISLER, H. J. **Model Theory**. North-Holland Publishing Company. Amsterdam. 1973.

[COP 72] COPI, I. M. **Introducción a la Lógica**. Trad Néstor Míguez. Título do Original: Introduction To Logic. Editorial Universitaria de Buenos Aires. 1972. 456p.

[COS 79] COSTA, N. C. A. **Os Fundamentos da Lógica**. HUCITEC, São Paulo. 1979. 255p.

[DAV 83] DAVIS, M. D. & WEYUKER, E. J. **Computability, Complexity and Languages. Fundamentals of Theoretical Computer Science**. Academic Press, Inc. 1983. 426p.

[END 72] ENDERTON, H. B. **A Mathematical Introduction To Logic**. Academic Press, Inc, New York.1972. 294p.

[FIR 89] FIREBAUGH, M. W. **Artificial Intelligence: A Knowledge-Based Approach**. PWS-KENT Publishing Company, Boston. 1989. 740p.

[FIT 87] FITTING, M. **Computability Theory, Semantics and Logic Programming**. New York Oxford University Press, NY. 1987. 198p.

[GEN 87] GENESERETH, M. R. & NILSSON, N. J. **Logical Foundations of Artificial Intelligence**. Morgan Kaufmann Publishers, Inc, Palo Alto, California. 1987. 406p.

[KOW 79] KOWALSKI, R. **Logic for Problem Solving**. Elsevier-North-Holland, New York. 1979. 288p.

[LIA 68] LIARD, L. **Lógica**. Trad Godofredo Ranger. Titulo do Original: Logique. Companhia Editora Nacional, São Paulo. 1968. 214p.

[LIG 64] LIGHTSTONE, A. H. **The Axiomatic Method. An Introduction To Mathematical Logic**. Prentice-Hall, Inc, New Jersey. 1964. 246p.

[LOP 81] LOPES, M. A. **Introdução a uma Teoria Geral de Problemas**. Tese de Doutorado. Departamento de Informática - PUC-Rio, Rio de Janeiro. Maio/81. 140p.

[LOP 85] LOPES, M. A. & ACIOLY, B. M. **Lógica de Resolução de Problemas - Uma Introdução**. CCEN-Informática - UFPE, Recife. 1985. 19p.

[MAN 66] MARINO, L. R. **Winning and Non-Losing Strategies in Games and Control**. Report No. SRC-91-A-66-36, Case Institute of Technology, Cleveland, Ohio. 1966.

[MAR 62] MARITAIN, J. **Elementos de Filosofia II - A Ordem dos Conceitos - Lógica Menor (Lógica Formal)**. Trad Ilza das Neves. Título do original: Éleménts de Philosophie II - L'Orde des Concepts - Petite Logique (Logique Formelle). Livraria Agir Editora, Rio de Janeiro.1962. 282p.

[MAT 68] MATES, B. **Lógica Elementar**. Trad Leonidas Hegenberg e Octanny Silveira da Mota. Título do Original: Elementary Logic. Companhia Editora Nacional, São Paulo. 1968. 298p.

[MES 65] MESAROVIC, M. D. **Toward a Formal Theory of Problem Solving - Computer Augmentation of Human Reasoning**. M. Sass & W. Wilkinson, eds. Spartan Books, Washington D.C.. 1965.

[MOR 67] MORA, J. F. & LEBLANC, H. **Lógica Matemática**. Fondo de Cultura Económica, México. 1967. 227p.

[NEW 59] NEWELL, A., SHAW, J. C. and SIMON, H. **Report on General Problem Solving Program**. Proceedings of International Conference Conference on Information Processing. Unesco, Paris. 1959.

[NIL 71] NILSSON, N. J. **Problem Solving Methods in Artificial Intelligence**. Mc Graw-Hill, New York. 1971.

[OLI 85] OLIVEIRA JR., W. R. **Especificação e Características dos Problemas Solúveis por Decomposição**. Dissertação de Mestrado, Depto. de Informática/CCEN/UFPE, Recife-Pernambuco-Brasil, 1985.

[POL 95] POLYA, G. **A Arte de Resolver Problemas**. Trad e Adap Heitor Lisboa de Araújo. Título do Original: How to Solve It. Editora Interciência Ltda, Rio de Janeiro. 1995. 196p.

[PRU 97] PRUDÊNCIA Santos, V. L. **Desenvolvimento de Uma Metodologia para Resolução de Problemas do Tipo Determine**. Tese de Mestrado. UFPB/CCT/DSC/COPIN, Campina Grande - Paraíba. Julho/97.

[QUI 72] QUINE, W. V. **Filosofia da Lógica**. Trad Therezinha Alvim Cannabrava. Título do Original: Philosophy of Logic. Zahar Editores, Rio de Janeiro. 1972. 140p.

[QUL 68] QUINLAN, J. R. and HUNT, E. B. **A Formal Deductive Problem Solving System**. JACM. October, 1968.

[RIC 94] RICH, E. e KNIGHT, K. **Inteligência Artificial**. MAKRON Books do Brasil Editora Ltda., São Paulo. 1994. 722p.

[RIN 88] RINGLAND, G. A. and DUCE, D. A. (Editors) **Approaches to Knowledge Representation: An Introduction**. John Wiley and Sons, New York. 1988. 260p.

[ROB 65] ROBINSON, A. **Introduction To Model Theory and To The Metamathematics of Algebra**. North-Holland Publishing Company, Amsterdam. 1965. 284p.

[ROB 66] ROBINSON, A. **Non-Standard Analysis**. North-Holland Publishing Company, Amsterdam. 1966. 294p.

[SAL 69] SALMON, W. C. **Curso Moderno de Filosofia - Lógica**. Trad Leonidas Hegenberg e Octanny Silveira da Mota. Título do Original: Logic. Zahar Editores, Rio de Janeiro. 1969. 144p.

[SAN 96] SANTOS, G. O. **Uma Metodologia para Resolução de Problemas via Refinamento da Especificação**. Tese de Mestrado. UFPB/CCT//DSC/COPIN, Campina Grande - Paraiba. Dezembro/96.

[SAW 69] SANDEWALL, E. **A Planning Problem Solver Based on Look-Ahead in Sthocastic Game Trees**. JACM. July, 1969.

[SIL 85] SILVA, I. P. **Uma Visão da Teoria de Banerji através da Teoria Geral de Problemas**. Dissertação de Mestrado, Depto. de Informatica/CCEN/UFPE, Recife-Pernambuco-Brasil, 1985.

[SOM 88] SOMMERHALDER, R. & Van WESTRHENEN, S. C. **The Theory of Computability. Programs, Machines, Effectiveness and Feasibility**. Addison-Wesley Publishing Company. 1988. 440p.

[TAR 76] TARSKI, A. **Introduction To Logic and Methodology of Deductive Sciences**. New York Oxford University Press, NY. 1976. 252p

[WID 64] WINDEKNECHT, T. G. **Problem Solving and Finite Automata**. A Status Report on Research in Artificial Intelligence and Linguistics, Case Institute of Technology, Cleveland, Ohio. 1964.

[WIL 73] WILDE, D. U. **An Introduction to Computing: Problem Solving, Alghoritms and Data Structures**. Prentice Hall, Inc., Englewood Cliffs, New Jersey. 1973.

[WIN 84] WINSTON, H. P. **Artificial Intelligence**. Addison-Wesley Publishing Company, Inc. 1984. 524p.

[YAS 71] YASUHARA, A. **Recursive Function Theory and Logic**. Academic Press, Inc, New York. 1971. 338p.

Dedução do Teorema $(a \Vert\!\!\rightarrow A \mathbin{\pmb{\wedge}} b \Vert\!\!\rightarrow B) \Rightarrow b \Vert\!\!\rightarrow B$ a partir do Sistema Formal Dedutivo da LRP

(1)	$a \Vert\!\!\rightarrow A$	hipótese
(2)	$b \Vert\!\!\rightarrow B$	hipótese
(3)	$a \wedge b \Vert\!\!\rightarrow A \wedge B$	(1), (2), $Rf_{\wedge}$
(4)	$a \wedge b \rightarrow b \Vert\!\!\rightarrow A \wedge B \rightarrow B$	(1), (3), $Rf_{\rightarrow}$
(5)	$(a \wedge b \Vert\!\!\rightarrow A \wedge B) \Rightarrow b \Vert\!\!\rightarrow B$	item c) seção 6.5
(6)	$(a \Vert\!\!\rightarrow A \mathbin{\pmb{\wedge}} b \Vert\!\!\rightarrow B) \Rightarrow b \Vert\!\!\rightarrow B$	item a) seção 6.5

$\vdash (a \Vert\!\!\rightarrow A \mathbin{\pmb{\wedge}} b \Vert\!\!\rightarrow B) \Rightarrow b \Vert\!\!\rightarrow B$

Apêndice B

Axiomas Lógicos do Sistema Formal Dedutivo da LRP

Para mostrar que Ax_1, Ax_2 e Ax_3 são tautologias basta fazer as atribuições de verdade necessárias às sentenças referente aos axiomas. Assim, temos:

$\bar{v}(Ax_1) = \bar{v}(\alpha \Rightarrow (\beta \Rightarrow \alpha))$

$\bar{v}(Ax_2) = \bar{v}((\alpha \Rightarrow (\beta \Rightarrow \gamma)) \Rightarrow ((\alpha \Rightarrow \beta) \Rightarrow (\alpha \Rightarrow \gamma)))$

$\bar{v}(Ax_3) = \bar{v}((\neg\beta \Rightarrow \neg\alpha) \Rightarrow (\alpha \Rightarrow \beta))$

Operando com as regras b) e e) da função de atribuição de verdade, temos:

$\bar{v}(Ax_1) = \bar{v}(\alpha \Rightarrow (\beta \Rightarrow \alpha))$

α	$\Rightarrow$	β	$\Rightarrow$	α
F	V	F	V	F
F	V	V	F	F
V	V	F	V	V
V	V	V	V	V

$\bar{v}(Ax_3) = \bar{v}((\neg\beta \Rightarrow \neg\alpha) \Rightarrow (\alpha \Rightarrow \beta))$

$\neg\beta$	$\Rightarrow$	$\neg\alpha$	$\Rightarrow$	α	$\Rightarrow$	β
V	V	V	V	F	V	F
F	V	V	V	F	V	V
V	F	F	V	V	F	F
F	V	F	V	V	V	V

$$\bar{v}(Ax_2) = \bar{v}((\alpha \Rightarrow (\beta \Rightarrow \gamma)) \Rightarrow ((\alpha \Rightarrow \beta) \Rightarrow (\alpha \Rightarrow \gamma)))$$

F	**V**	F	V	F	**V**	F	V	F	**V**	F	V	F
F	**V**	F	V	V	**V**	F	V	F	**V**	F	V	V
F	**V**	F	F	F	**V**	F	V	V	**V**	F	V	F
F	**V**	F	V	V	**V**	F	V	V	**V**	F	V	V
V	**V**	F	V	F	**V**	V	F	F	**V**	V	F	F
V	**V**	F	V	V	**V**	V	F	F	**V**	V	V	V
V	**F**	F	F	F	**V**	V	V	V	**F**	V	F	F
V	**V**	F	V	V	**V**	V	V	V	**V**	V	V	V