Isabel Tonin

Formas Normais de Fórmulas em Lógica de Primeira Ordem e Aplicações em Dedução Automática

Florianópolis
2003

Universidade Federal de Santa Catarina

Programa de Pós-Graduação em Engenharia Elétrica

Formas Normais de Fórmulas em Lógica de Primeira Ordem e Aplicações em Dedução Automática

Tese submetida à
Universidade Federal de Santa Catarina
como parte dos requisitos para a
obtenção do grau de Doutor em Engenharia Elétrica.

Isabel Tonin

Florianópolis, Fevereiro de 2003.

Formas normais de fórmulas em lógica de primeira ordem e aplicações em dedução automática

Isabel Tonin

Esta Tese foi julgada adequada para obtenção do Título de Doutor em Engenharia Elétrica, na Área de Concentração de *Sistemas de Informação*, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.

Bittencourt

Guilherme Bittencourt, Dr.Rer.Nat.
Orientador

Edson Roberto de Pieri, Dr.
Coordenador do Programa de Pós-graduação em Engenharia Elétrica

Newton C. A. da Costa, Dr.
Relator

Banca Examinadora:

Bittencourt

Guilherme Bittencourt, Dr.Rer.Nat.
Presidente

Ana Teresa de Castro Martins

Ana Teresa de Castro Martins, Dra.

Jacques Wainer, Dr.

Rosvelter João Coelho da Costa, Dr.

Edson Roberto de Pieri, Dr.

À memória de meu pai,
que creio estar orgulhoso pela realização deste trabalho.

À minha mãe,
minha grande companheira de todos os momentos.

À minha amada afilhada Tiquinha,
e à não menos amada Quica,
que moram no meu coração,
(e seus nomes nas letras góticas deste trabalho)
pelo amor infantil, puro e sincero, a mim dedicado.

O Bambu Chinês

Bem lentamente, um diminuto broto desabrocha a partir do bulbo. Depois de plantada a semente deste incrível arbusto, por aproximadamente cinco anos nada se vê, exceto este lento desabrochar. Durante cinco anos, todo o crescimento é subterrâneo. Invisível a olho nu, mas... uma maciça e fibrosa estrutura de raiz que se estende vertical e horizontalmente pela terra está sendo construída. Então, no final do quinto ano, o bambu chinês cresce até atingir a altura de 25 metros.

Um escritor de nome Covey escreveu: "Muitas coisas na vida pessoal e profissional são iguais ao bambu chinês. Você trabalha, investe tempo, esforço, faz tudo o que pode para nutrir seu crescimento, e às vezes, não se vê nada por semanas, meses ou anos. Mas se tiver paciência para continuar trabalhando, persistindo e nutrindo, o seu quinto ano chegará, e com ele virão um crescimento e mudanças que você jamais esperava..."

O bambu chinês nos ensina que não devemos facilmente desistir de nossos projetos e de nossos sonhos... em nosso trabalho especialmente, que é um projeto fabuloso que envolve mudanças de comportamento, de pensamento, de cultura e de sensibilização, devemos sempre lembrar do bambu chinês para não desistirmos facilmente diante das dificuldades que surgirão.

Procure cultivar sempre dois bons hábitos em sua vida: a persistência e a paciência, pois você merece alcançar todos os seus sonhos !!!

"É preciso muita fibra para chegar às alturas e, ao mesmo tempo, muita flexibilidade para se curvar ao chão."

(autor desconhecido)

Agradecimentos

Relembrar cinco anos, o tempo de execução desta tese ... difícil tarefa esta de agradecer a todos que de alguma forma participaram e me auxiliaram neste trabalho. Primeiramente agradeço a minha família pelo apoio e torcida incondicionais. Em especial a minha irmã Angela, sem a qual, minhas idas e vindas à Alemanha seriam bem menos freqüentes.

Sem par também, foi o apoio dos amigos e colegas. Um especial agradecimento a Augusto C.P.L. da Costa e Frederico L.G. de Freitas, amigos e colegas, pelos momentos (e muita cerveja, prost!) compartilhados em nossas viagens à Alemanha, onde boa parte desta tese foi desenvolvida. Agradeço a todos os amigos e colegas que colaboraram nesta "fase alemã" da tese, dentre eles Ulla Dietrich, Peter Kullmann e Yoshimi Uchida Siebert, por me fazerem sentir tão bem vinda nesse país.

Gostaria de agradecer ao Prof. Robert Kowalski, do Imperial College, Londres, pelo empenho pessoal no envio de uma cópia de sua tese. Pela consideração, muito obrigada! Ao Prof. Newton C. A. da Costa, pelos valiosos conselhos e atenção demonstrada. Aos membros da banca examinadora, pelas críticas e sugestões.

Um agradecimento muito especial a James J. Hunt, namorado, amigo e companheiro, grande responsável pelo término desta tese, escrita em sua casa, na Alemanha, entre junho e novembro de 2002. Seu apoio compreendeu desde equipamentos e livros necessários, discussões teóricas e críticas até o "tu podes!" nos momentos de insegurança. E tudo isto regado com muito carinho, boa comida e bons vinhos ... que se pode querer mais?

E finalmente, a Guilherme Bittencourt, orientador desta tese, o agradecimento mais difícil de expressar em palavras. Não posso sinceramente dizer que sempre lhe quis agradecer. Que nada! No entanto hoje, após tantos anos de convivência, o sentimento que lhe dedico é uma mistura de gratidão, respeito e amizade.

A todos, muito obrigada!

Resumo da Tese apresentada à UFSC como parte dos requisitos necessários para a obtenção do grau de Doutor em Engenharia Elétrica.

Formas normais de fórmulas em lógica de primeira ordem e aplicações em dedução automática

Isabel Tonin

Fevereiro/2003

Orientador: Guilherme Bittencourt, Prof. Dr.Rer.Nat.
Área de Concentração: Prova Automática de Teoremas, Inteligência Artificial
Palavras-chave: Inteligência Artificial, Prova Automática de Teoremas, Dedução Automática, Lógica, Método de Inferência, Formas Normais, Subsunção
Número de Páginas: 191

Pesquisas recentes em prova automática de teoremas para lógica clausal de primeira ordem lidam com fórmulas representadas em apenas uma de suas formas normais, conjuntiva ou disjuntiva, para realizar inferências lógicas. No entanto, a autora toma outro caminho e examina o uso simultâneo de ambas formas normais de uma fórmula para realizar estas inferências. A motivação original foi o desenvolvimento de uma estratégia que melhorasse a eficiência do *Método de Inferência por Transformação Dual* (*Método Dual*). Este método, para lógica de primeira ordem, processa uma inferência lógica sem a explícita aplicação de nenhuma regra de inferência; ao invés, ele transforma sucessivamente a fórmula em suas formas normais conjuntiva e disjuntiva e remove as contradições tornadas explícitas por estas transformações. A fim de desenvolver uma estratégia que explorasse a representação de uma fórmula nas duas formas normais, relações estruturais apresentadas por esta representação foram formalizadas e usadas como base para expressar e filtrar inferências realizadas durante o processo de dedução. Conseqüentemente, este trabalho apresenta resultados em duas áreas: as formas normais de uma fórmula lógica, suas relações, simplificação e transformação de uma para a outra; e a estratégia propriamente dita. Embora o uso simultâneo de ambas formas normais pela estratégia tenha se mostrado redundante, este estudo deu origem a um novo método de inferência com interessantes características: (i) ele combina atributos de ambos cálculos: de saturação e de tableau e conexão, (ii) ele permite o processamento paralelo de inferências, e (iii) ele facilita o reuso de inferências previamente efetuadas. Estas são características não encontradas em nenhum outro método conhecido pela autora.

Abstract of Thesis presented to UFSC as a partial fulfillment of the requirements for the degree of Doctor in Electrical Engineering.

Normal forms of first-order logic formulae and applications on automated deduction

Isabel Tonin

February/2003

Adviser: Guilherme Bittencourt, Prof. Dr.Rer.Nat.
Area of Concentration: Automated Theorem Proving, Artificial Intelligence
Keywords: Artificial Intelligence, Automated Theorem Proving, Automated Deduction, Logic, Inference Method, Normal Forms, Subsumption
Number of Pages: 191

Current research in automated theorem proving for first order clausal logic relies on formulas represented in only one of their normal forms, either conjunctive or disjunctive, for logical inferencing. However the author takes another tack and investigates the simultaneous application of both normal forms of a formula for logical inference. The original motivation was to develop a strategy to improve the efficiency of the *Inference Method through Dual Transformation* (*Dual Method*). This method for first order logic processes a logical inference without an explicit application of any inference rule; rather it successively transforms a formula in its conjunctive and disjunctive normal forms and removes the contradictions made explicit by these transformations. In order to develop a strategy that exploits the representation of a formula in its two normal forms, structural relations presented by this representation were formalized and used as a basis to express and filter inferences performed during the deduction process. Consequentially, this work presents results in two areas: the logical formula normal forms, their relation, simplification, and transformation from one to the other; and the strategy itself. Although the simultaneous application of the normal forms by the strategy proved to be redundant, the investigation gave rise to a new inference method with interesting characteristics: (i) it combines capabilities of both saturation calculi and connection and tableau calculi in a single method, (ii) it enables parallel processing of inferences, and (iii) it facilitates the reuse of previously generated inferences. These are attributes not found in any other method known to the author.

Sumário

Lista de Figuras

Lista de Tabelas

"Where shall I begin, please your Majesty?" he asked.
"Begin at the beginning," the King said, gravely, "and go on till you come to the end: then stop."

Lewis Carroll, Alice's Adventures in Wonderland, 1865

Capítulo 1

Introdução

Inferência Lógica desperta, desde há muito, interesse na comunidade de Inteligência Artificial (IA) e, mais especificamente, na comunidade da IA Simbólica [26]. Seu estudo define a área de pesquisa conhecida como *Prova Automática de Teoremas* (Automated Theorem Proving) — ou, mais atualmente, como *Dedução Automática* (Automated Deduction) ou ainda *Raciocínio Automático* (Automated Reasoning) — que tem se mantido bastante ativa.

A fim de melhor introduzir este trabalho, inicialmente discorremos sobre alguns pontos relativos à área[1]:

- *Inteligência Artificial?!*

  Qual a primeira idéia que nos vem à mente quando ouvimos falar de Inteligência Artificial? Robôs ... ou pelo menos máquinas pensantes. Nos lembramos dos filmes de ficção científica, com seus computadores capazes de responder (e até decidir) qualquer coisa, todo o tipo de maquinaria, eletrodomésticos e outras tantas parafernálias fazendo tudo automaticamente, robôs andando de um lado para outro, comunicando-se e até afeiçoando-se aos seres humanos. A dupla de robôs C3P0 e R2D2, por exemplo, leais ao seu dono Lucky Skywalker no filme *Guerra nas Estrelas*? O corpo era de lata, mas suas reações: alegres, preocupados, com medo, atrapalhados... Ou ainda, o robô "humanóide" assassino enviado do futuro para o presente pelas "forças inimigas" no filme *O Exterminador do Futuro I* e que, na continuação deste filme, é enviado do futuro pelas "forças do bem", passando a desempenhar o papel de herói da estória? Neste último, pelo menos,

[1]Os conceitos apresentados estão baseados em [7], [51] e [30]

é passada a idéia de um ser programável, ou seja, que existe alguém que define o papel que este humanóide deve desempenhar.

Seguindo ainda a linha de robôs humanóides programados tem-se os "replicantes" do filme *Blade Runner*, humanóides que buscam por seu criador para exigir-lhe o prolongamento de suas vidas. Ou, mais recentemente, o humanóide criança no filme *Inteligência Artificial*, programado para amar! Teríamos ainda, como exemplo de "seres artificiais inteligentes", HAL no filme *2001 Uma Odisséia no Espaço*, não um robô mas um computador que a tudo controla, sabe e responde. Ou ainda *Matrix*, no filme de mesmo nome, um computador que dominou o mundo e utiliza os seres humanos como "baterias vivas". Mas vamos parar por aqui não dizendo que isto tudo seja uma bobagem, mas que estamos longe de tal realidade.

A possibilidade de criar uma "inteligência artificial" parece fascinar o ser humano desde há muito. De fato, a IA foi construída a partir de idéias filosóficas, científicas e tecnológicas herdadas de outras ciências, algumas tão antigas como a lógica, com seus 23 séculos. Questões filosóficas como o chamado *problema corpo-mente*, motivam e dividem opiniões de pesquisadores da IA. Porém, oficialmente, enquanto ciência, a IA tem uma história de pouco mais de quarenta anos, iniciados com muita polêmica, expectativas exageradas e conseqüentes fracassos. Definições básicas como seu nome, seus objetivos, seu objeto de estudo ou o que vem a ser exatamente *inteligência* até hoje não obtiveram consenso na comunidade de IA.

Independentemente de definição formal precisa, o objetivo central da IA é imitar, através do uso de computadores, o máximo possível da atividade mental humana. As duas principais áreas de conhecimento que levaram à IA são a Lógica Matemática e a Computação, mas seus estudos apresentam hoje intersecção com outras grandes áreas de conhecimento como a filosofia, psicologia, biologia, entre outras.

Sua evolução deu-se em duas linhas distintas de pesquisa: a linha *Conexionista*, que visa a modelagem da inteligência humana através da simulação dos componentes do cérebro humano e que deu origem às Redes Neuronais Artificiais; e a linha *Simbólica*, cujos princípios são apresentados no artigo de 1980 de A. Newell *Physical Symbol Systems* [25], e segue a tradição da lógica, tentando simular a inteligência humana enquanto comportamento lingüístico e simbólico. Observa-se hoje o crescimento de uma nova linha da IA, a *IA Evolutiva*, baseada na observação de mecanismos evolutivos encontrados na natureza, como auto-organização e comportamento adaptativo (observados nos modelos de algoritmos genéticos e autômatos celulares, por exemplo).

Cada uma destas linhas de pesquisa depara-se com limitações teóricas e tecnológicas, vantagens e desvantagens, oferecendo muitos desafios à pesquisa atual. Muito se tem avançado e estudado, porém embora muitas coisas inteligentes tenham sido feitas, a simulação de algo que pudesse passar por inteligência autêntica está ainda a uma enorme distância.

- *Símbolos e IA, um casamento lógico*

Para muitos, a lógica é a "matemática" da IA. A lógica pode ser vista como o estudo do raciocínio, mais propriamente como o estudo do raciocínio válido.[2] Podemos também ver este raciocínio válido como inteligência. Assim sendo, a busca da IA por "dispositivos" capazes de apresentar um comportamento inteligente pode ser vista como a busca por dispositivos capazes de apresentar um comportamento que requer um raciocínio válido. Assim, o estudo do "raciocínio válido" é o ponto de contato entre estas duas áreas.

O uso da lógica simbólica na IA foi proposto inicialmente por P. Hayes e J. McCarthy [24]. Eles observaram que programas de IA tinham de manipular uma grande quantidade de conhecimento que deveria, de alguma forma, estar representada nestes programas. Pareceu-lhes bastante apropriado representar formalmente este conhecimento independentemente do programa. A lógica clássica de primeira ordem era uma boa linguagem para isto, pois ela possui uma estreita relação com a linguagem natural, sua semântica é clara e bem estudada, e além disto apresenta um mecanismo formal para tornar explícito o conhecimento implícito em uma dada formalização: a dedução lógica.

Inicialmente, a pesquisa na IA Simbólica concentrou-se na busca de formalismos gerais capazes de resolver qualquer tipo de problema; pouco depois viu-se que tal rumo de pesquisas era inviável, pois esbarrava em dois problemas:

  - o primeiro problema é teórico e advém do uso de modelos baseados em lógica

[2] Esta é a visão da lógica tradicionalmente adotada pela comunidade de IA e, também, neste trabalho. Porém, como apontado pelo Prof. Newton C.A. da Costa [12], esta é uma visão bastante limitada do poder da lógica. Conferir matéria no apêndice B.

de primeira ordem como formalismo básico. É conhecido como "Explosão Combinatória" e é inerente aos métodos baseados em lógica: a memória e o tempo necessários para a resolução de um problema cresce exponencialmente com o tamanho do mesmo;

– o segundo recai sobre o fato de ser necessário, para resolver qualquer tipo de problema, a representação do conhecimento total do mundo real — inclusive conhecimentos de senso comum do comportamento humano. Estes são normalmente de difícil manipulação pela lógica de primeira ordem por serem incompletos, ou parcialmente incoerentes. Mesmo assim, lógicas não clássicas podem ser utilizadas para representar diversos tipos de problemas mais específicos (garantindo uma certa eficiência na solução de problemas, mesmo que resultando em lógicas "mais fracas" que a lógica clássica de primeira ordem).

Muitas soluções foram e continuam sendo propostas e, apesar dos problemas, a lógica tem desempenhado dois importante papéis na IA:

– ser fonte de linguagens e "lógicas" para o desenvolvimento de programas inteligentes;

– fornecer subsídio matemático e conceitual subjacente às várias áreas de pesquisa em IA, como Representação de Conhecimento, Prova Automática de Teoremas e Sistemas Especialistas por exemplo.

Dois importantes resultados são a base do poder da lógica e também de suas limitações: o primeiro, provado independentemente em 1930 por Kurt Gödel e Jacques Herbrand, afirma que existem sistemas lógicos de prova, ou métodos de prova, que são *completos*, ou seja nos quais toda fórmula verdadeira pode ser provada. O segundo, provado em 1936, também independentemente por Alonzo Church e Alan Turing, afirma que a lógica é *semi-decidível*, isto é, não existe um método geral capaz de decidir, em um número finito de passos, se uma fórmula é verdadeira. Isto significa que, dada uma fórmula verdadeira é sempre possível encontrar uma prova, mas que existem fórmulas não verdadeiras para as quais não existe nenhum método finito que, garantidamente, seja capaz de decidir sobre sua não validade (então, métodos para a prova automática de teoremas podem nunca parar).

Este resultado, um tanto negativo, da lógica é um dos grandes argumentos dos críticos da IA simbólica. Já, para seus adeptos, apesar de muito importante, ele não impossibilita que, utilizando a lógica, as máquinas um dia estejam aptas a raciocinar quase tão bem quanto pessoas. Justificam que, a despeito da incompletude de alguns sistemas, seres humanos conseguem utilizá-los de maneira bastante razoável.

- *Lógica $\cap$ IA = Raciocínio Válido*

  Raciocínio válido é, de certa forma, um raciocínio formal, que segue algumas regras pré-estabelecidas. Ao falar-se em "raciocínio formal" pode parecer que está-se tentando criar regras para raciocinar através delas. Porém, a idéia é exatamente oposta. Pensa-se, ou raciocina-se, simplesmente. O papel da lógica é tentar modelar formalmente este raciocínio. Sua história remonta aos antigos filósofos gregos, principalmente Aristóteles, que estabeleceu seus fundamentos de maneira sistemática.

  Este raciocínio, ao ser "formalizado", define *métodos de inferência lógica*, ou seja, métodos formais de dedução lógica. Através destes métodos, conclusões podem ser obtidas, baseadas em premissas. Sendo as premissas vistas como conhecimentos, pode-se afirmar que, nelas baseados, novos conhecimentos são explicitados através dos métodos de inferência lógica. Estes métodos de inferência são também chamados de *métodos de prova* ou *cálculos.* Os primeiros métodos de prova foram concebidos para serem utilizados por seres humanos. A partir da introdução por Robinson [32] e Smullyan [35], em 1960, de métodos eficientes para prova automática de teoremas por computador, a lógica passou a ser estudada também como método computacional para a solução de problemas.

Dada uma idéia do objeto de estudo da IA e seus ramos, pode-se introduzir este trabalho. Para tal, é contada resumidamente a história de seu desenvolvimento.

Este trabalho foi iniciado durante a escrita do artigo *Concurrent Inference through Dual Transformation* [6]. Nele o Prof. Guilherme Bittencourt propõe um método de inferência lógica baseado unicamente na *transformação dual,* isto é, a transformação de uma fórmula lógica em forma normal conjuntiva para sua forma normal disjuntiva e vice-versa. Este método foi denominado *Método Dual* e sua implementação e estudo, bem como de um algoritmo que realiza a transformação dual, foram assunto da dissertação de mestrado intitulada *Um Método para Inferência Lógica baseado na Transformação Dual* [46].

Pronta esta primeira implementação confirmou-se o que já se sabia teoricamente: apesar de apresentar características interessantes e ser completamente diferente dos demais métodos conhecidos, o método dual, como apresentado no artigo, era bastante ineficiente (considerando-se o número de teoremas gerados durante o processo de prova). Justificava-se, então, o esforço de lhe desenvolver uma *estratégia* que o tornasse mais eficiente. Por estratégia entende-se a definição de uma maneira de proceder a inferência de uma conclusão desejada visando minimizar o número de conclusões intermediárias geradas.

Uma estratégia foi desenvolvida para o método dual mantendo sua característica principal: a consideração das duas formas normais de uma fórmula lógica. Ao longo deste desenvolvimento, muitos problemas teóricos intermediários apareceram e foram estudados, alguns relativos diretamente à estratégia em si e outros relativos às formas normais e à relação entre elas. Deste estudo foram obtidos importantes resultados relativos às formas normais, dentre eles, a constatação da não unicidade da forma condensada de uma fórmula na forma disjuntiva. Estes resultados encontram-se devidamente publicados inclusive no respeitado *Journal of Automated Reasoning.*

Um fato inusitado, no entanto, foi resultante deste estudo: a consideração simultânea das formas normais pela estratégia, como primeiramente pensada, mostrou-se redundante pois a consideração apenas da forma conjuntiva é suficiente àquela estratégia. Porém, com o tempo percebeu-se que a estratégia, primeiramente pensada para o método dual, quando considerada apenas a forma conjuntiva de uma fórmula, define, por si, um outro método de prova (cujo nome temporário é *Goal*) com características bastante interessantes, entre elas, a definição de um conceito, o de *Eliminação de Literais*, que auxilia no armazenamento e reutilização de inferências realizadas durante um processo de prova (mesmo em prova de diferentes teoremas). Esta característica pode ser entendida como a atribuição de "memória" a provadores automáticos de teoremas e não é encontrada em nenhum provador automático que conhecemos.

Do estudo realizado foram obtidos os seguintes resultados:

- relativo às formas normais de uma fórmula, sua simplificação e transformação de uma para a outra, bem como sobre a relação de subsunção entre fórmulas expressas em forma normal:
  - o artigo *"Forma normal disjuntiva em lógica de primeira ordem"* apresentado no o artigo *LAPTEC'2000 - Primeiro Congresso de Lógica Aplicada à Tecnologia* [43] realizado em São Paulo – SP;
  - o artigo *"A Concurrent Algorithm for Logical Subsumption"* apresentado no *LAPTEC'2001 Segundo Congresso de Lógica Aplicada à Tecnologia* [44], realizado em São Paulo – SP, e publicado na série *Frontiers in Artificial*

Intelligence and its Applications - IOS Press [45];

- o artigo *"An Algorithm for Dual Transformation in First-Order Logic"* publicado no respeitado *Journal of Automated Reasoning* [10];

- relativo à estratégia, o artigo *"A Proof Strategy Based on a Dual Representation"* apresentado no *AISC'2000 Fifth International Conference Artificial Intelligence and Symbolic Computation Theory, Implementations and Applications* realizado em Madri, Espanha [9].

Este trabalho insere-se na área de *Prova Automática de Teoremas*, ramo da *IA Simbólica* e apresenta o estudo desenvolvido desde a estratégia, como primeiramente pensada, até a proposta do novo método de prova. Este estudo é parte do projeto de um sistema híbrido para modelagem cognitiva, proposto pelo Prof. Guilherme Bittencourt, descrito nos artigos *In the Quest of the Missing Link* [5] e *A Multi-Agent Approach to First-Order Logic* [8].

"And what is the use of a book," thought Alice, "without pictures or conversations?"

Lewis Carroll, Alice's Adventures in Wonderland, 1865

Capítulo 2

Como não tem figuras ...

Grande é o número de tópicos estudados pela comunidade de dedução automática. Para introdução de *Automated Deduction - a basis for Applications, volume 1: foundations - Calculi and Methods* [4] foi escrito, pelos professores Wolfgang Bibel e Peter H. Schmitt, com auxílio do professor Eric Rosenthal, um diálogo fictício entre um pesquisador da área de dedução automática e um cientista da computação. Neste diálogo são apresentados os principais tópicos pesquisados atualmente pela comunidade, os problemas e as expectativas de avanço.

A fim de dar uma idéia atual da área de dedução automática, este diálogo é utilizado como "um guia" para os tópicos abordados. Para tal, são apresentadas partes deste diálogo[1] seguidas dos comentários e informações extras que se fazem necessários.

2.1 Estado da arte - um diálogo

P: Dedução automática têm sido objeto de pesquisa por 40 anos. Quais tópicos interessantes ainda restam pesquisar?

R: Não existe uma resposta simples. Eu ainda lembro o que primeiro me impressionou quando eu me deparei com lógica matemática. Foi a existência de sistemas formais – por exemplo, cálculo de seqüentes – que cobrem todo o possível raciocínio lógico. Com isto eu quero dizer que muitos destes sistemas são completos para a lógica de primeira ordem. A pesquisa corrente mais importante refere-se ao projeto de sistemas eficientes, isto é, sistemas que possam realmente fazer algo não trivial em um computador.

[1]Com autorização do Prof. Peter Schmitt para tradução e utilização neste trabalho.

P: Então o senhor acha que ajustar o cálculo de seqüentes é a resposta?

R: De fato, não. Diferentes sistemas formais completos existem, cada um enfatizando um ponto de vista particular ou tendo sido desenvolvido para um propósito específico. A contribuição da ciência da computação tem sido a pesquisa de sistemas formais de dedução adequados à automação em computadores. Vários sistemas emergiram e o desenvolvimento de novos continua sendo objeto de pesquisa.

P: Francamente, esta variedade de cálculos[2] no estilo "Torre de Babel" me perturba. Pode-se decidir qual é o melhor e esquecer os outros?

R: Bem, de fato não. Existe uma boa razão para esta variedade. Cada cálculo tem suas próprias vantagens, e não existe nenhum evidente vencedor, pelo menos até agora. Mas isto não é tão ruim como parece, e você certamente não precisa conhecer todas as ramificações se você não pretende tornar-se um especialista em dedução automática. Existem basicamente duas categorias de cálculos:

- cálculos de saturação: o método de resolução pertence à esta categoria e durante um tempo foi visto como o único cálculo bem-sucedido para dedução automática. Mas esta visão tem mudado, e provadores automáticos de teoremas que utilizam diferentes cálculos são, agora, fortes concorrentes.
- cálculos de tableau e de conexões: estes são largamente tratados na literatura ...

Uma boa fonte para saber-se o que tem sido produzido na área é acompanhar os trabalhos apresentados na *Cade - Conference on Automated Deduction* (resultados e publicações são encontrados no endereço http://www.cs.albany.edu/~nvm/cade.html) cuja primeira conferência aconteceu em 1974 e encontra-se atualmente em sua 18^a edição. Nestes trabalhos pode-se comprovar a variedade de sistemas formais completos atualmente pesquisados.

Pode-se também perceber a forte tendência, no início, ao desenvolvimento de métodos baseados na Resolução e a mudança atual de enfoque. Não que métodos baseados em Resolução não sejam mais estudados, longe disto, de fato a impressão que se tem

[2]Métodos de Prova.

é que métodos baseados na Resolução encontram-se em um diferente estágio de estudo: vários métodos "de propósito geral" foram desenvolvidos e, agora, a tendência é restringir o domínio dos mesmos e desenvolver "habilidades" específicas para cada domínio (como tratamento de equações, inequações, reescrita de fórmulas, ...).

No entanto, referente à categoria de métodos baseados em Tableau e Conexões, novos métodos baseados têm sido amplamente desenvolvidos e aplicados com sucesso. Nos resultados da *CASC - CADE ATP System Competition* (que podem ser encontrados em http://www.math. miami.edu/~tptp/CASC/), uma "competição" de provadores automáticos de teoremas realizada juntamente com a *CADE*, é apresentada a comparação do desempenho de alguns provadores automáticos de teoremas. Observa-se que a grande vantagem apresentada pela Resolução nas primeiras edições desta competição é hoje suplantada por provadores automáticos de teoremas baseados em Tableau e Conexões. Outro sinal deste desta mudança de enfoque foi o surgimento, em 1992, de uma conferência específica para os estudos referentes aos métodos desta categoria, a *TABLEAUX - International Conference on Automated Reasoning with Analytic Tableaux and Related Methods* (http://i12www.ira.uka.de/TABLEAUX/).

> P: Quando o senhor diz "largamente tratados" o senhor provavelmente quer dizer que os autores primeiramente descrevem seus métodos e então apresentam uma prova elaborada de completude. Isto, não sendo da área, me confunde: prova-se a completude de um cálculo embora sabendo que, por causa da semi-decidibilidade do problema de satisfazibilidade (satisfiability) e devido à restrições "da vida real", qualquer implementação de seu cálculo será incompleta.
>
> R: Você está certo. Mesmo assim, faz diferença um provador automático ser incompleto pelas razões que você expôs ou ser teoricamente incompleto. No primeiro caso, o aumento de recursos computacionais pode melhorar a performance, ou mesmo resolver a incompletude. Já no segundo caso isto nunca acontecerá. Mas, deixe-me enfatizar que o grande esforço da pesquisa atual é por melhor desempenho; provas de completude tornaram-se rotina profissional e não o objetivo em si. Os principais resultados são integração e apresentação unificada de diferentes variações de cálculos e redução do espaço de busca. Grande parte das técnicas de "poda" deste espaço preservam a completude, mas heurísticas são também comumente empregadas.

Esta busca por desempenho pode, por sua vez, ser classificada em duas categorias.

A primeira busca por soluções computacionais, de engenharia de software, para melhorar o desempenho dos provadores automáticos. Nesta categoria encontram-se estudos de representações, tabelas, linguagens, enfim, técnicas que melhorem o armazenamento e recuperação das informações necessárias ao processamento das inferências. Já a outra categoria, muitas vezes considerada mais nobre, busca a eficiência de um método pela restrição de procedimentos efetuados por este durante a prova. Por exemplo, o desenvolvimento de estratégias que diminuam o número de inferências necessárias, o estudo da unificação, problema comum a todos os método de prova (como por exemplo o estudo da *cycle unification*, uma tentativa de identificar inferências cíclicas e realizá-las em um único passo), a tentativa de identificar o tipo de fórmula e decidir qual a estratégia mais adequada para sua prova, etc. De fato ambas abordagens são importantes e podem ser combinadas para atribuir eficiência a provadores automáticos de teoremas.

P: Deixe-me retornar à minha pergunta anterior relativa à "Torre de Babel". Está certo, eu aceito que este é o estado da arte, e que isto não é tão ruim. Mas, é necessário continuar e ainda criar novos cálculos? Já não está tudo lá?

R: Quem sabe? Algumas vezes eu acredito que todos os princípios e métodos de dedução são já conhecidos, e então surge uma nova idéia inesperadamente bem sucedida. É semelhante à situação das linguagens de programação: pensa-se que os princípios de programação estão todos entendidos e exaustivamente pesquisados, mas novas linguagens aparecem e prosperam! Deixe-me dizer mais sobre minha resposta anterior. Lá eu me referi apenas a cálculos de propósito geral. Pode acontecer alguém estar interessado somente em problemas de um domínio específico. O exemplo mais comum é o raciocínio de igualdades, onde o interesse é apenas relativo a equações (nem mesmo inequações) entre objetos. Para tal, cálculos específicos foram desenvolvidos. É natural usar-se métodos específicos para tarefas específicas. O desafio está em construir sistemas de prova automática de propósito geral que mudem ("chaveiem") para provadores específicos sempre que possível e combinem resultados intermediários em um único e abrangente esforço de prova. Os domínios de problemas restritos são descritos, no jargão da comunidade de dedução automática, como "teoria", e as combinações de deduções de propósitos gerais e de propósitos específicos são chamadas "raciocínio em teorias" (theory reasoning).

Este "chaveamento" refere-se ao que chamamos de "tentativa de identificar o tipo de fórmula". Esta solução, acreditamos, pode representar o grande avanço da área de raciocínio automático: a reunião dos esforços despendidos no desenvolvimento de diversos métodos, utilizando para cada teoria o método que lhe é mais adequado. Porém, infelizmente, ainda percebe-se na área um certo "ranço" separatista do tipo "O meu método é o melhor" e não é comum a integração entre grupos de estudos de métodos de diferentes categorias.

> P: Deixe-me perguntá-lo sobre algo que sempre me incomodou. O senhor mencionou sua empolgação relativa a sistemas lógicos completos. Eu posso não ser um especialista em lógica mas eu lembro do meu curso de sistemas formais. Lá eu aprendi que teoremas de completude são somente possíveis para lógica de primeira ordem. Para lógica de alta ordem, onde pode-se dizer algo como "para todos os conjuntos ..." ou "para todas as funções ...", nenhum sistema de prova completo é possível.
>
> R: Eu estou contente que você tenha levantado esta questão. A princípio é possível reduzir-se todo o raciocínio lógico para lógica de primeira ordem, por exemplo codificando-se todo um domínio dado em uma teoria de conjuntos de primeira ordem não tipados. Mas esta não é uma solução prática. Esta codificação é muito difícil. Como um aparte, deixe-me dizer que a distinção entre lógicas de primeira e de alta ordem é um pouco mais sutil. ... Existem sistemas de prova de teorema de alta ordem que permitem livremente o uso de operações de alta ordem. O preço a ser pago, como você mencionou, é que nenhum procedimento de prova automático completo é possível. Ao invés, provas nestes sistemas são construídas interativamente pelo usuário, com passos do procedimento sendo preenchidos automaticamente. Por um lado isto pode até ser considerado uma vantagem, uma vez que o usuário pode utilizar seu conhecimento e intuição. Este tipo de prova de teorema é radicalmente diferente dos sistemas de dedução totalmente automáticos discutidos até agora. Por exemplo, a noção de espaço de busca, que é uma preocupação básica em dedução automática, não faz nem sentido em provadores interativos. Isto talvez explique porque ambas comunidades desenvolveram-se sem perceberem-se muito. Mas uma vez que você começa a pensar sobre isto, elas parecem "casar" perfeitamente, e muitos grupos estão agora trabalhando na combinação de sistemas totalmente automático e interativos em um único sistema.

(Praticamente nada a acrescentar. Provadores automáticos de teorema para lógicas de alta ordem não foram estudados pela autora e não são abordados neste trabalho. Contudo, cabe ressaltar que são amplamente utilizados em aplicações práticas, como por exemplo, em aplicações que envolvem verificação, comentadas na seqüência.)

P: O senhor sabe, eu não estou assim tão interessado em aprender como provadores de teoremas são implementados. Eu estou muito mais interessado no produto final.

R: Como alguém de fora, cujo interesse é o de usuário final, você pode ter tal atitude. Mas você me perguntou quais tópicos de pesquisa existem em dedução automática, e nós estamos discutindo os pontos que são cruciais para aplicações bem sucedidas.

P: Aha! Aplicações. A palavra que eu estava esperando ouvir. O que o senhor pode fazer com todos seus sistemas de dedução?

R: A área de prova automática de teoremas – como qualquer outro esforço de pesquisa – deve muito de seu vigor às perspectivas vislumbradas. Prova automática de teoremas tem várias áreas de aplicação. Deixe-me começar com a mais teórica delas. Estou falando de prova de teoremas em matemática, o que continuará sendo, sem dúvida, uma importante e desafiante área de aplicação. Mais de uma conjectura antes em aberto, que desafiaram esforços humanos, foram resolvidas por sistemas de prova automáticas. Outra abordagem é o desenvolvimento de sistemas de dedução usados como assistentes matemáticos. Isto é, os sistemas não resolvem todo o problema automaticamente; ao invés, eles auxiliam o matemático, freqüentemente aliviando-o das tarefas mais elementares. A aplicação que provavelmente mais tem recebido atenção é a verificação automática de software e hardware para garantir a ausência de erros/falhas em programas e "chips".

P: Eu concordo que esta é uma idéia maravilhosa, mas seus frutos ainda parecem ser para um futuro bem distante. Especialmente uma vez que tudo o que se pode verificar são simples algoritmos de classificação.

R: Você deve estar brincando! Sua informação está totalmente desatualizada. O atual estado de arte está muito além disto. Dados tempo suficiente e esforço humano, qualquer projeto de hardware ou programa pode ser verificado por um provador de teorema interativo. Mesmo provadores totalmente automáticos podem manusear teorias com centenas de axiomas. O objetivo agora é desenvolver sistemas que façam a verificação economicamente.

P: E alguém se importa o suficiente para investir dinheiro em verificação? Veja a situação atual. Nenhum dos softwares amplamente vendidos no mundo vem sem erros. Algumas vezes eles são detectados e corrigidos na próxima versão. E todos parecem satisfeitos com isto.

R: Isto é uma pergunta referente a mercado. Ninguém pode prever o que o mercado exigirá. Em aplicações críticas as exigências já são muito mais rigorosas. Fabricantes de hardware estão começando a ficar mais preocupados quanto à verificação. Todo o mercado pode, um dia, recusar-se a aceitar o padrão atualmente estabelecido para softwares. ... Não faz sentido apenas sentar e esperar. Mais pode ser feito. Ao invés de verificar programas, você poderia tentar sintetizá-los. Áreas de aplicação para sistemas dedutivos, outras que verificação, são quase ilimitadas.

Muitas são atualmente as aplicações para a lógica (ou as lógicas) em diversas áreas do conhecimento: filosofia, lingüística, matemática, engenharia, química, biologia, direito, psicologia etc. Algumas destas aplicações podem ser conferidas nos anais do *LAPTEC - Congress of Applied Logic to Technology*, um congresso internacional que ocorre anualmente em São Paulo com o apoio da *Faculdade SENAC de Ciências Exatas e Tecnologia*. E muitas destas aplicações incluem o uso de provadores automáticos de teoremas. Estes, porém, não devem ser vistos como um "produto final" de prateleira que compra-se, instala-se e usa-se. Devem, no entanto, ser vistos como ferramentas de alto nível que seguem um antigo princípio da computação: "lixo entra, lixo sai" (garbage in, garbage out).

Como mostrado neste capítulo, muitos são os métodos desenvolvidos, muito esforço tem sido despendido para torná-los eficientes, muito estudo para identificar a que tipo de problema cada um é mais adequado. Mesmo assim, as aplicações encontram-se quase que restritas a laboratórios de pesquisa. Poucos são os provadores de teoremas realmente utilizados, e mesmo assim, normalmente por matemáticos para a solução de problemas matemáticos.

A razão para esta realidade é, de certa forma, bastante simples: pensar em aplicação é pensar no uso de provadores automáticos de teoremas como auxiliares na solução de algum problema. Para tanto, este "algum problema" deve ser um teorema a ser provado, ou refutado, sob o escopo de alguma fórmula. Ou seja, o problema em si e toda a realidade que o contorna, devem estar representados em sentenças lógicas ... e esta é a razão, normalmente, que dificulta seu uso. Representar um problema do mundo real em sentenças lógicas não é uma tarefa trivial e envolve, primeiramente, a escolha da melhor lógica para expressá-lo. (Muitas vezes esta escolha tem recaído em lógicas não clássicas — fuzzy, tipada, modal, ...) para representar tais problemas. Além da escolha da lógica, deve-se garantir que o problema está bem representado, ou seja, que as sentenças lógicas que o descrevem são suficientes para a prova desejada. Por isto todo o projeto que inclui o uso de um provador automático de teoremas deve ser acompanhado por um profissional especialista com capacidade de representar um problema em linguagem lógica.

2.2 Conclusão

Com a apresentação do "estado da arte" da área, esperamos ter passado uma idéia dos principais tópicos estudados atualmente pela comunidade de prova automática de teoremas. Este trabalho insere-se neste contexto no tópico "desenvolvimento de novos métodos". Como visto, nenhum método é totalmente melhor que outro. Cada método possui características que o torna mais adequado para um determinado tipo de problema que para outro. E assim é, certamente, o método apresentado neste trabalho.

"Contrariwise," continued Tweedledee, "if it was so, it might be, and if it were so, it would be; but as it isn't, it ain't. That's logic!"

Lewis Carroll, Through the Looking Glass, 1871

Capítulo 3

A lógica da IA

Inferência lógica, ou o uso da lógica na IA, mais especificamente na área de dedução automática, é o tema abordado neste capítulo. Para tal, são apresentados os conceitos gerais de dedução utilizados nesta área. Esta apresentação é feita de maneira informal no intuito de fornecer as noções de dedução lógica necessárias ao entendimento do objeto de estudo da área de dedução automática e deste trabalho. Primeiramente são apresentados os conceitos relativos à dedução lógica propriamente dita e, por fim, sua automatização.

3.1 Se é verdade, então prova!

Para formalizar um raciocínio, a lógica trata fundamentalmente de dois conceitos: *Verdade* e *Prova.* Estes conceitos foram investigados durante séculos por filósofos, matemáticos e lingüistas. A parte da lógica que estuda os *valores verdade* (do inglês "truth values") é chamada *Teoria dos Modelos* e a que estuda as *provas* é chamada de *Teoria das Provas.*

Neste trabalho são apresentados apenas alguns conceitos da teoria dos modelos. As frases em lógica são chamadas de fórmulas. As fórmulas constituem uma linguagem formal. Cada fórmula pode ser associada a um *valor verdade* isto é, ao valor verdade *verdadeiro (V)* ou ao valor verdade *falso (F).* Uma maneira formal de atribuição de valores verdade a fórmulas foi proposta por Alfred Tarski [40] e ficou conhecida como *Semântica de Tarski.* Esta atribuição é chamada de *interpretação.* Uma interpretação que torna uma fórmula verdadeira, isto é, que *satisfaz* uma fórmula, é chamada de *modelo.* Uma fórmula é dita *satisfazível* (do inglês "satisfiable") se ela possui pelo menos um modelo. Se toda interpretação possível for um modelo para a fórmula, esta é dita *válida* ou *tautológica.* Por outro lado, uma fórmula que é tornada falsa em pelo menos uma interpretação é dita *inválida* e, se ela é tornada falsa por toda interpretação possível (não possui nenhum modelo) é dita *insatisfazível* (do inglês "unsatisfiable") ou *contraditória* [11].

Apesar de utilizar os conceitos da teoria dos modelos, neste trabalho a ênfase recai na teoria das provas. Nela, é explorada a estrutura dedutiva da linguagem lógica, isto é, seus mecanismos de inferência de novas fórmulas.

3.1.1 A sintaxe da linguagem lógica

Uma linguagem lógica de primeira ordem — notada por $\mathcal{L}(\mathcal{P}, \mathcal{F}, \mathcal{C}, \mathcal{V})$ — é formalmente determinada pela especificação dos seguintes conjuntos finitos de símbolos:

- um conjunto $\mathcal{P}$ de símbolos de predicados, onde cada símbolo de predicado é usualmente notado por letras maiúsculas $(P, Q, R, \ldots)$ ou expressões significativas formadas por letras maiúsculas $(MAIOR, HOMEM, IGUAL, \ldots)$,
- um conjunto $\mathcal{F}$ de símbolos de função, onde cada símbolo de função é usualmente notado por letras minúsculas $(f, g, h, \ldots)$ ou expressões significativas formadas por letras minúsculas $(pai, mae, mais, \ldots)$,
- um conjunto $\mathcal{C}$ de símbolos de constante, onde cada símbolo de constante é usualmente o nome dos objetos $(Joao, a, 16, \ldots)$, e
- um conjunto $\mathcal{V}$ de símbolos de variável, onde cada símbolo de variável é usualmente notado por letras minúsculas $(x, y, x_1, \ldots)$.

A cada símbolo de predicado e de função é associada uma *aridade*, isto é, o número de argumentos do predicado e da função. Os símbolos de predicado com aridade zero são chamados *símbolos proposicionais*[1], além disto os símbolos de constante podem ser também considerados como funções de aridade zero. Além destes conjuntos, o *alfabeto* da linguagem lógica, compreende os *conectivos lógicos* — $\neg$ (negação), $\vee$ (ou), $\wedge$ (e), $\rightarrow$ (implicação) e $\leftrightarrow$ (equivalência) — e os *quantificadores* — $\forall$ (para todo) e $\exists$ (existe). A *sintaxe* da linguagem lógica de primeira ordem é dada pela definição de *termos*, *fórmulas atômicas* e *fórmulas bem formadas* da lógica de primeira ordem, definidos a seguir.

Um *termo* é definido recursivamente como segue:

[1]Uma linguagem lógica onde todos os símbolos de predicado têm aridade zero é chamada *Proposicional*, neste caso os conjuntos $\mathcal{F}$,$\mathcal{C}$ e $\mathcal{V}$ não são relevantes para a definição da linguagem.

1. um símbolo de constante é um termo,
2. um símbolo de variável é um termo,
3. se f é um símbolo de função enária e $t_1, \ldots, t_n$ são termos, então $f(t_1, \ldots, t_n)$ é um termo
4. todos termos são gerados pela aplicação das regras acima.

Uma *fórmula atômica*, ou *átomo*, é definida como segue: se P é um símbolo de predicado enário, e $t_1, \ldots, t_n$ são termos, então $P(t_1, \ldots, t_n)$ é um átomo. Nenhuma outra expressão é um átomo.

Uma *fórmula bem formada*, ou simplesmente *fórmula*, em lógica de primeira ordem é definida recursivamente como segue:

1. um átomo é uma fórmula,
2. se F e G são fórmulas, então $\neg F, F \vee G, F \wedge G, F \rightarrow G$ e $F \leftrightarrow G$ são fórmulas,
3. se F é uma fórmula e x é uma variável livre[2] em F, então $\forall x F$ e $\exists x F$ são fórmulas,
4. fórmulas somente são geradas por um número finito de aplicações das regras acima.

Fórmulas que contêm variáveis são chamadas *abertas*, caso contrário são chamadas de *fechadas*. Uma ocorrência de uma variável em uma fórmula é dita *ligada* se, e somente se, está no escopo de um quantificador ou é a própria varável do quantificador. Uma ocorrência de uma variável em uma fórmula é dita *livre* se, e somente se, esta ocorrência não é ligada. Já uma variável é dita *livre* em uma fórmula se pelo menos uma de suas ocorrências na fórmula é livre. E, se pelo menos uma ocorrência de uma variável na fórmula é ligada, então nesta fórmula esta variável é dita *ligada*.

Variáveis livres podem ser substituídas por termos mais complexos. A operação que permite isto é chamada *substituição*. Uma substituição consiste em um mapeamento entre o conjunto de variáveis V e o conjunto de termos. Uma substituição θ é notada por:

$$\theta = \{x_1/t_1, \ldots, x_n/t_n\}$$

[2] Definição dada na seqüência.

onde $x_i \in V$, $x_i \neq x_j$ para $i \neq j$ e os t_i's são termos. Aplicar uma substituição θ sobre uma variável x_i – notado $x_i\theta$ – significa substituir x_i por t_i. Apesar de ser definida como um mapeamento sobre o conjunto de variáveis, a ação de uma substituição pode ser facilmente estendida para termos e fórmulas atômicas:

- $c\theta = c$, para $c \in C$.
- $\phi(t_1, \ldots, t_n)\theta = \phi(t_1\theta, \ldots, t_n\theta)$, para $\phi \in F$ ou $\phi \in P$.

A aplicação de uma substituição θ em uma fórmula W é notada por $W\theta$. Substituições podem ser compostas de maneira associativa de maneira que :

$$(W\theta)\tau = W(\theta\tau)$$

As fórmulas obtidas por substituição a partir de uma fórmula dada são chamadas *instâncias* desta fórmula. Caso a fórmula obtida não apresente mais nenhuma variável, ela é chamada *instância fechada*, caso contrário *instância aberta.* Assim, por exemplo, $P(a)$ é uma instância fechada de $P(x)$, uma vez que pode ser obtida desta pela aplicação da substituição $\{x/a\}$.

3.1.2 Inferência

A *inferência lógica* (ou simplesmente inferência) é um mecanismo formal para geração de conclusões (fórmulas) verdadeiras a partir de premissas (fórmulas) verdadeiras.

Exemplo 1 *Qualquer leitor das tiras de quadrinhos de Waterson intitulada "Calvin e Haroldo" sabe que (i)* sempre que a mãe de Calvin está muito irada é porque ele aprontou "uma das suas". *Conhecendo este fato, o que se pode concluir vendo um quadrinho onde (ii)* a mãe de Calvin está muito irada*? Utilizando-se os conhecimentos (i) e (ii) como premissas pode-se deduzir que (iii)* Calvin aprontou uma das suas.

Nota-se que a partir dos fatos já conhecidos (i) e (ii), por inferência, foi gerado um novo conhecimento (iii); não exatamente novo, mas "tornado explícito", pois o mesmo já era implicitamente expresso pelas premissas. Esta habilidade de realização de inferências é uma parte essencial da inteligência.

O raciocínio empregado no exemplo anterior é bastante simples: acreditar em uma premissa é considerá-la verdadeira; acreditar na premissa (i) não significa afirmar que a mãe de Calvin esteja muito irada, mas afirmar que quando ela está muito irada ele aprontou uma das suas; por sua vez, acreditar na premissa (ii) é afirmar que a mãe está muito irada; portanto, para acreditar-se nas duas premissas, *tem-se* de acreditar na conclusão (iii). Este raciocínio é uma *inferência lógica.*

Exemplo 2 *Porém, se ao invés das premissas do exemplo 1 descreverem a relação entre as artes de Calvin e a ira de sua mãe, elas descrevessem a natureza imaginária das conversas de Calvin com Haroldo, seu tigre de pelúcia. Elas poderiam expressar que (i)* Se Haroldo é um tigre de pelúcia, então suas "falas" são imaginárias *e (ii)* Haroldo é um tigre de pelúcia, *uma conclusão logicamente inferida seria (iii)* Suas falas são imaginárias.
Para a premissa (i) deste novo exemplo ser verdadeira, Haroldo pode ou não ser um tigre de pelúcia. Mas se o for, dizer que suas falas são imaginárias expressa uma verdade.

Novamente acreditando-se nas duas premissas, acredita-se também na conclusão, ou seja, a conclusão é logicamente implicada pelas premissas. Nos dois exemplos o raciocínio necessário para se alcançar a conclusão é o mesmo. De fato, a estrutura das sentenças (premissas e conclusões) que os compõem é a mesma, o que os diferencia é o significado que cada termo da linguagem utilizada possui, dando a cada exemplo uma interpretação própria. Uma interpretação representa um significado atribuído a uma fórmula lógica. Cada exemplo pode ser visto como uma *interpretação* para a mesma fórmula lógica que têm como premissas (i) $\alpha \rightarrow \beta$ e (ii) α, e como conclusão (iii) β. No exemplo 1 a interpretação de α é "a mãe de Calvin está muito irada" e a interpretação de β é "Calvin aprontou uma das suas". Já no exemplo 2, a interpretação de α é "Haroldo é um tigre de pelúcia" e a de β é "as falas de Haroldo são imaginárias".

Este tipo de raciocínio, utilizado para gerar a conclusão, independe da interpretação. Sabe-se no entanto que, baseados nas duas premissas padrão, a conclusão obtida é uma conclusão lógica. O raciocínio em questão expressa uma verdade lógica, estrutural, uma *tautologia* — verdade lógica que se mantém sob qualquer interpretação. Pode-se atribuir qualquer significado para α e β que, ao acreditar-se nas premissas está-se, também, acreditando na conclusão. "Logicamente" falando, este é um *raciocínio válido.*

Ao modelar um raciocínio, sentenças são representadas por símbolos lógicos de acordo com uma sintaxe pré-estabelecida pela linguagem lógica. Porém ao representar-se sentenças em fórmulas lógicas, normalmente tem-se em mente uma interpretação específica para elas. Foram mostradas nos exemplos anteriores duas situações distintas que podem ser representadas exatamente pela mesma fórmula. Além destas, existe uma infinidade de outras interpretações possíveis para aquela fórmula. Como garantir que uma conclusão obtida é uma conclusão verdadeira para uma específica interpretação? Uma forma é obter conclusões através de regras formais de raciocínio que garantam que tais conclusões são verdadeira em todas as interpretações onde as premissas o são.

3.1.3 Regras de Inferência

A regra formal que modela um raciocínio é chamada de *regra de inferência*. Uma regra de inferência é simplesmente uma função sintática que, dado um conjunto de fórmulas lógicas – as premissas – gera uma nova fórmula. Uma regra de inferência consiste de (i) um conjunto de fórmulas padrão chamado de *condições* e (ii) um outro conjunto de fórmulas padrão chamado de *conclusões*. Sempre que as premissas "casam" com as condições está habilitada a geração de sentenças que "casam" com as conclusões. Se, a partir de fórmulas verdadeiras, uma regra de inferência produzir apenas fórmulas verdadeiras, a regra é dita *correta* (do inglês "sound").

Por serem sempre verdadeiras, tautologias, quando utilizadas como base para regras de inferência, levam necessariamente a regras corretas. O que permite que tautologias sejam utilizadas como regras de inferência é uma outra regra de inferência, mais geral, conhecida como *regra da substituição*, que afirma que uma tautologia formada por símbolos proposicionais permanece uma tautologia quando estes símbolos são substituídos por fórmulas lógicas arbitrárias. Sejam α e β fórmulas quaisquer, as seguintes tautologias são freqüentemente utilizadas como regra de inferência:

$$\begin{array}{rl}
(\alpha \wedge (\alpha \rightarrow \beta)) \rightarrow \beta & \textit{Modus Ponens} \\
(\neg\beta \wedge (\alpha \rightarrow \beta)) \rightarrow \neg\alpha & \textit{Modus Tollens} \\
((\alpha \rightarrow \beta) \wedge (\beta \rightarrow \delta)) \rightarrow (\alpha \rightarrow \delta) & \textit{Silogismo Hipotético} \\
\forall x.\alpha \rightarrow \alpha\{x/a\} & \textit{Especialização} \\
\alpha\{x/a\} \rightarrow \exists x.\alpha & \textit{Generalização}
\end{array}$$

Assim, por exemplo, segundo a regra *Modus Ponens*, ao acreditarmos (i) em uma sentença α e (ii) que a sentença α implica a sentença β, temos de acreditar também (iii) na sentença β. Neste caso, os ítens (i) e (ii) são as condições da regra e o item (iii) é a conclusão obtida.

Da mesma forma, pela regra *Modus Tollens*, se não acreditamos em uma sentença β e acreditamos que esta é implicada por uma sentença α, então não podemos acreditar em α. A regra *Silogismo Hipotético*, por sua vez, expressa a transitividade da implicação lógica, isto é, ao acreditarmos simultaneamente que uma sentença α implica uma sentença β e que esta implica uma sentença δ, então deve-se acreditar que a sentença α também implica a sentença δ. Já a regra da *Especialização* expressa que, se uma sentença α é verdadeira independentemente do valor atribuído à variável x, então α é verdadeira quando a variável x tem um valor arbitrário. Por fim, a regra da *Generalização* expressa que, se uma sentença α é verdadeira pelo menos quando a variável x tem um valor arbitrário, então existe pelo menos um valor para x no qual esta sentença é verdadeira.

É claro que, para ser utilizada em uma dedução, uma regra de inferência deve ser necessariamente correta. Esta é a base da inferência lógica: a conclusão é uma *conseqüência lógica* das premissas, isto é, é implicada por elas. Uma conseqüência lógica é verdadeira em todas as interpretações nas quais suas premissas o forem, ou seja, toda interpretação que é um modelo para as premissas também o é para suas conseqüências lógicas.

3.1.4 Métodos de Prova

Um *teorema* é uma fórmula válida. Provar um teorema δ é demonstrar a sua validade, normalmente demonstrando que o mesmo pode ser inferido a partir de um conjunto inicial de fórmulas Δ, que a fórmula $\Delta \rightarrow \delta$ é uma tautologia. Ou seja, provar um teorema consiste em demonstrar, pela aplicação sucessiva de regras de inferência, que o mesmo é uma conseqüência lógica das premissas. A *prova* é esta seqüencia de deduções realizadas.

De uma forma geral, um sistema lógico de prova ou *método de prova* consiste de um conjunto de premissas (uma fórmula), um conjunto de regras de inferência corretas e uma estratégia de aplicação de tais regras [23]. Em 1930, Kurt Gõdel e Jacques Herbrand provaram, independentemente, que existem *métodos de prova*, que são *completos* (do inglês "complete"), ou seja, nos quais todo teorema da fórmula pode ser provado pela aplicação sucessiva de suas regras de inferência, de acordo com a estratégia pré-definida.

Existem duas abordagens para provar um teorema: ou demonstra-se que ao acreditar nas premissas deve-se acreditar também no teorema propriamente dito, ou demonstra-se que ao acreditar nas premissas não é possível acreditar na negação do teorema a ser provado. Estas duas abordagens definem dois tipos de métodos de prova: os *afirmativos* (ou dedutivos simplesmente) e os de *refutação* (que também são métodos dedutivos).

Em um método de prova afirmativo, dado um conjunto de premissas e um teorema a ser provado, teoremas implicados pelas premissas são gerados sucessivamente até que o teorema desejado seja gerado constituindo, assim, sua prova.

Figura 3.1: uma tira de quadrinhos de Calvin e Haroldo, de Waterson

Exemplo 3 *Voltando ao "Calvin e Haroldo", vê-se em uma de suas tiras, ilustrada na figura 3.1, uma seqüencia de cenas onde ambos estão dançando freneticamente. O último quadro mostra um quarto escuro com a mãe de Calvin sentada na cama, recém acordada, comentando com o marido: "ou eu ainda estou sonhando ou ele está tocando música clássica a 78 rpm". Neste caso, pode-se querer provar o óbvio: Calvin está realmente tocando música clássica a 78 rpm.*

Como premissas têm-se: (i) Se ela (mãe) não está sonhando, então ele (Calvin) está tocando *, (ii)* Se ela está acordada, então ela não está sonhando *e (iii)* Ela está acordada. *Pelas premissas (ii) e (iii) em um primeiro passo conclui-se, por Modus Ponens, que (iv)* Ela não está sonhando. *Adotando-se (iv) como premissa e a premissa inicial (i) pode-se concluir, também por Modus Ponens, que (v)* Ele está tocando, *como era de se esperar. A prova foi alcançada em dois passos. A seqüencia de deduções lógicas realizadas para alcançar o teorema ((ii) e (iii), (iv) e (i), (v)) é a sua prova. Esta seqüência é também chamada de árvore de prova e pode ser visualizada na figura 3.2.*

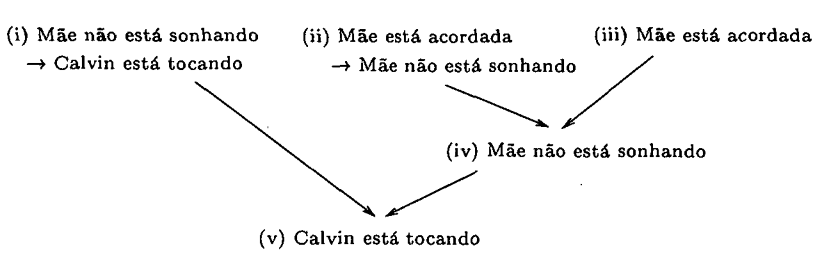

Figura 3.2: árvore de prova do exemplo 3 obtida por Modus Ponens

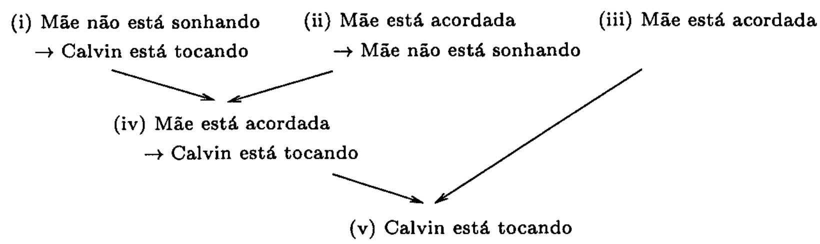

Figura 3.3: árvore de prova do exemplo 3 obtida por silogismo hipotético e Modus Ponens

Por outro lado, pelas premissas (ii) e (i) pode-se, por silogismo hipotético, concluir que (iv) Se ela está acordada, então ele está tocando. *A conclusão final (v) pode, então, ser obtida pelas premissas (iii) e (iv). A prova neste caso, apesar de representar um caminho diferente de deduções ((i) e (ii), (iii) e (iv), (v)), também foi obtida em dois passos. A árvore desta prova é mostrada na figura 3.3.*

No exemplo 3 a prova do teorema foi alcançada quase que diretamente. Em problemas mais complexos, porém, percebe-se que, ao serem gerados teoremas na busca pela prova, muitos deles são "ruins", não relevantes, no sentido de não fazerem parte da seqüencia de deduções que leva à prova. A busca pelos "bons" teoremas, ou relevantes, define *estratégias de prova.* Esta estratégias "guiam" a prova, evitando assim o crescimento exponencial do espaço de busca (número de possibilidades a serem avaliadas a cada passo). Ainda hoje mantém-se bastante ativa a pesquisa de estratégias de prova.

Como dito anteriormente, outra forma de provar um teorema é refutá-lo, isto é, é demonstrar que não se pode acreditar na sua não implicação pelas premissas. Seja Δ um conjunto de premissas, refutar o teorema δ é demonstrar que $\neg(\Delta \rightarrow \delta)$ é uma contradição. Assim, pelo princípio do terceiro excluído[3], a sua implicação pelas premissas $\Delta \rightarrow \delta$ tem de ser verdadeira. Como $\neg(\Delta \rightarrow \delta)$ pode ser transformada para $\Delta \wedge \neg\delta$, normalmente, para refutar um teorema une-se a sua negação às premissas e demonstra-se que a fórmula resultante $\Delta \cup \{\neg\delta\}$ é contraditória.

O raciocínio subjacente aos métodos de refutação é simétrico ao subjacente aos métodos de prova afirmativos. Como antes, crer nas premissas é crer que elas possuem pelo menos uma interpretação que as satisfaça simultaneamente. Demonstrar que uma fórmula é contraditória é demonstrar que nenhuma interpretação a satisfaz. Portanto é demonstrar que não se pode acreditar nas premissas e na negação do teo-

[3]Do latim *Tertium Non Datur*, princípio da lógica clássica: se uma fórmula é verdadeira a sua negação é falsa e vice-versa.

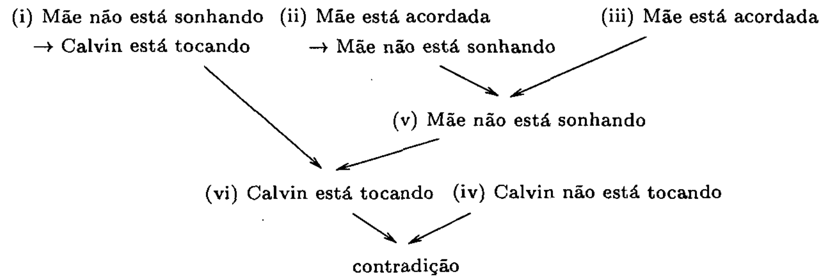

Figura 3.4: refutação do exemplo 3

rema simultaneamente. Novamente, como o número de interpretações pode ser infinito, demonstra-se pelas contradições estruturais, sintáticas, da fórmula.

Exemplo 4 *Vejamos como seria refutar um teorema utilizando o exemplo 3. Para refutar o teorema "Calvin esta tocando música clássica a 78 rpm" deve-se unir sua negação à premissas. Com isto têm-se como premissas : (i)* Se ela não está sonhando então ele está tocando, *(ii)* Se ela está acordada, então ela não está sonhando, *(iii)* Ela está acordada *e a negação do teorema como nova premissa (iv)* Calvin não está tocando. *Na figura 3.4 uma possível refutação para esta fórmula é mostrada. Pelas premissas (ii) e (iii) infere-se que (v)* ela não está sonhando, *de (i) e (v) infere-se que (vi)* Calvin está tocando, *o que contradiz o teorema negado (iv). Nenhuma interpretação pode satisfazer uma fórmula e sua negação ao mesmo tempo. Alcançada a contradição sabe-se, então, que a negação da premissa (iv) é realmente um teorema, isto é, é conseqüência lógica das premissas originais (i), (ii) e (iii).*

Assim como a prova afirmativa, a prova por refutação é a seqüência de fórmulas geradas até a contradição ser alcançada. Valem para um método de refutação as mesmas noções de *correção* (do inglês "soundness") e *completude* (do inglês "completeness") apresentadas para os métodos de prova afirmativa. Os métodos de refutação mais conhecidos não são completos, mas *completos para a refutação*[4]. Um método é completo para a refutação se, através dele, todo teorema pode ter a sua negação refutada.

[4] Nem toda fórmula verdadeira pode ser provada (isto é, gerada das premissas por regras de inferência) mas sua negação pode ser refutada. Por exemplo, utilizando-se o método de resolução, que é um método de refutação, do conjunto de premissas {P(x)} não se pode gerar a fórmula P(a) mas apenas refutar a sua negação, $\neg P(a)$

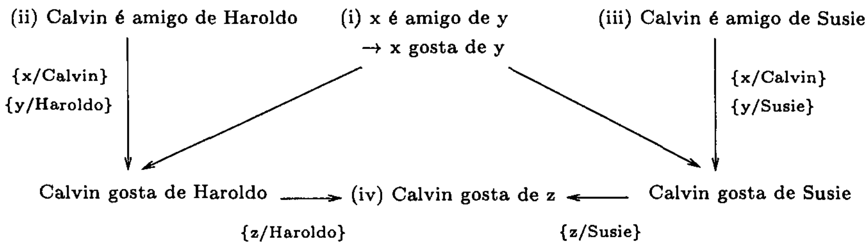

Figura 3.5: árvore de prova do exemplo 5

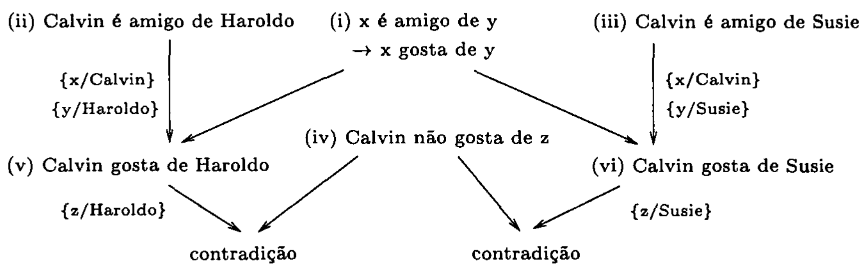

Figura 3.6: árvore de prova por refutação do exemplo 5

3.1.5 A prova como solução de problemas

Até agora foram apresentados nos exemplos premissas e teoremas cuja única função da prova era demonstrar se o teorema era de fato um teorema ou não. No entanto, se as premissas forem vistas como a descrição de um fato e o teorema como uma pergunta, esta seria (até agora) respondida com sim ou não. Porém, utilizar um método de inferência para solução de problemas é mais que isto. É poder obter respostas para questões desconhecidas. Este uso para solução de problemas é mostrado no exemplo 5.

Exemplo 5 *Assumindo-se que toda pessoa gosta de seus amigos e que Calvin é amigo de Haroldo e de Susie, pode-se querer saber* **de quem** *Calvin gosta. Não é o mesmo que perguntar explicitamente se ele gosta de Susie ou/e se ele gosta de Haroldo. As premissas do problema são: (i)* Se x é amigo de y, então x gosta de y, *(ii)* Calvin é amigo de Haroldo *e (iii)* Calvin é amigo de Susie. *E, como a pergunta é "De quem Calvin gosta?" e a resposta esperada é "Calvin gosta de fulano", o teorema é (iv)* Calvin gosta de z. *A prova é mostrada na figura 3.5.*

Se, ao invés, quisermos a prova por refutação, substitui-se o teorema por sua negação (iv) Calvin não gosta de z. *A refutação é mostrada na figura 3.6.*

Percebe-se que, em cada caso, o teorema pode ser provado de duas formas, sendo *cada uma, uma resposta para a pergunta expressa pelo teorema. Estas respostas são dadas pelos valores que a variável "z" do teorema assume nas prova, no caso, Haroldo e Susie.*

Neste exemplo nota-se que, para expressar relações gerais como a da premissa (i), precisou-se adotar o uso de variáveis, no caso, x e y. Também no teorema, precisou-se utilizar uma variável, no caso, z. As variáveis são como lacunas que podem ser preenchidas com diversos valores. Ao expressar o teorema com uma variável, quer-se na realidade, saber que valores a ela podem ser atribuídos para tornar o teorema verdadeiro. O preenchimento destas lacunas é dado pela substituição de toda ocorrência de uma variável por um termo. Esta troca de toda ocorrência de uma variável pelo termo definido na substituição é chamada de *aplicação da substituição*. As substituições aplicadas nas provas do exemplo anterior são notadas pelo par {variável/termo}.

Com isto terminamos nossa apresentação dos conceitos lógicos subjacentes aos métodos de prova. Os primeiros métodos de prova corretos e completos foram criados por Gödel e Herbrand. Existem hoje diversos métodos de prova corretos e completos. Este conceito de prova é muito importante para a IA porque ele é a base da automatização de um raciocínio lógico.

Infelizmente, da semi-decidibilidade da lógica de primeira ordem decorre que sendo uma fórmula um teorema das premissas, ela pode ser provada dentro do sistema lógico (e sua negação pode ser refutada, como será visto a seguir). No entanto, não existe um método geral capaz de decidir, em um número finito de passos, se uma fórmula é ou não um teorema das premissas. Disto resulta que métodos de prova automática podem não terminar. Vários tópicos são encontrados na literatura (por exemplo em [30]) relacionados a esta propriedade de decidibilidade, como por exemplo: o teorema de Gödel e a incompletude de qualquer axiomatização finita da aritmética, o problema da *parada* de Turing e o conceito de "computabilidade", a matemática não recursiva e a descrição de conjuntos não recursivos mas recursivamente enumeráveis (quando apenas os elementos do conjunto podem ser calculados algoritmicamente e não os elementos de seu complemento), o teorema de Herbrand e as árvores semânticas infinitas com fechamento finito.

3.2 Dedução _ _ Automática _ _ bip!

Os primeiros métodos de prova desenvolvidos eram mais apropriados para serem utilizados por seres humanos, por exemplo, *sistemas de axiomas*, *dedução natural* e *seqüentes de Gentzen* [15]. A implementação destes métodos em computadores mostra-se normalmente ineficiente, ou pela quantidade de regras de inferência disponíveis, ou devido ao fato das regras envolverem escolhas ou, ainda, pela complexidade das mesmas. Somente a partir da introdução do *método de resolução* por Robinson [32] e do *método de tableaux* Smullyan [35], métodos de prova eficientes para dedução automática por computador, foi considerado viável o desenvolvimento de sistemas computacionais que realizassem inferências lógicas – os *provadores automáticos de teoremas*. Ambos são métodos de refutação e exploram o fato de expressões lógicas poderem ser colocadas em *formas canônicas*. O resultado, embora não seja muito adequado para ser manipulado por seres humanos devido à perda da estrutura original das expressões lógicas, permite uma manipulação computacional bastante eficiente.

Desde então, diversas técnicas de representação de expressões lógicas, bem como diversas estratégias de controle para o processo de prova foram propostas, especialmente para o método de resolução [38]. Ainda hoje a área de dedução automática permanece bastante ativa, sendo objeto de diversas conferências internacionais como, por exemplo, a "International Conference on Automated Deduction - CADE" que encontra-se em sua décima oitava edição.

O fato de ser possível associar uma semântica operacional a um método de prova automática [47] permitiu a definição de uma linguagem de programação baseada em lógica, a linguagem Prolog (por exemplo, [37]). Esta linguagem, inicialmente restrita a laboratórios de pesquisa, hoje é amplamente utilizada, e seus aperfeiçoamentos e possíveis extensões são objetos de intensas pesquisas.

3.2.1 Um pouco de teoria ...

Antes de apresentarmos alguns métodos de prova utilizados em dedução automática, faz-se necessária a formalização do que foi apresentado até agora e a introdução de mais alguns conceitos.

Dada uma linguagem lógica de primeira ordem e um formalismo, chamado *interpretação*, que permite associar de maneira sistemática fórmulas desta linguagem a elementos do conjunto $\{V, F\}$, podemos determinar o valor verdade, verdadeiro ou falso, de qualquer fórmula lógica a partir de uma interpretação dada.

Um problema mais interessante consiste em determinar se, dado um conjunto G de fórmulas, uma fórmula W é ou não uma *conseqüência lógica* de G, isto é, se toda interpretação que satisfaz todas as fórmulas em G, simultaneamente, satisfaz também a fórmula W. Esta relação entre fórmulas é notada por:

$$G \models W.$$

Este problema é mais geral do que a determinação de valores verdade, pois não depende de nenhuma interpretação em particular, sendo antes uma relação entre as fórmulas pertencentes ao conjunto $G \cup \{W\}$. Sua interpretação intuitiva é análoga ao "raciocínio hipotético" humano, isto é, à capacidade de assumir certas hipóteses como verdadeiras e considerar as conseqüências.

A dificuldade em relação a este problema vem do fato de que a definição de conseqüência lógica envolve *todas* as interpretações, que são em número infinito. A necessidade de um método operacional capaz de determinar se uma fórmula é ou não conseqüência lógica de um conjunto de fórmulas levou ao desenvolvimento de *métodos de prova* tais que, a partir do conjunto G e utilizando certas *regras de inferência*, seja possível gerar novas fórmulas, eventualmente levando à geração da fórmula W.

Caso exista uma seqüência de aplicações de regras de inferência que leve das fórmulas de G em W, então diz-se que W pode ser *provado* a partir de G, e nota-se:

$$G \vdash W.$$

Caso o método de prova seja correto, isto é, contenha apenas regras de inferência corretas, então:

$$G \vdash W \Rightarrow G \models W.$$

Caso o reverso desta expressão – $G \models W \Rightarrow G \vdash W$ – também seja verdadeiro, isto é, se, para toda a fórmula W, conseqüência lógica de G, seja possível encontrar uma prova de W a partir de G, então o método de prova é dito *completo.*

Então, em métodos corretos e completos

$$G \vdash W \Leftrightarrow G \models W$$

as propriedades de *provabilidade* e *implicação lógica* são equivalentes.

Em resumo, dada uma fórmula G e um teorema W, provar W é demonstrar que $G \rightarrow W$ é uma tautologia, ou que $\neg(G \rightarrow W)$ (ou $G \cup \{\neg W\}$) é uma contradição.

Muitas vezes, como no método de resolução, a prova dá-se em *ciclos de inferência*, isto é, os teoremas de uma fórmula G são inferidos e, caso o teorema W não tenha sido provado, os teoremas inferidos são unidos à G e novo ciclo de inferências se inicia. Este procedimento é repetido até que a prova seja alcançada. Dois aspectos devem ser salientados no tocante à *até que a prova seja alcançada*:

- devido à semi-decidibilidade da lógica, caso W não seja um teorema de G, pode significar *para sempre*, e
- devido ao excesso de teoremas inferidos (e retidos) a cada ciclo, o que aumenta exponencialmente o espaço de busca[5], pode significar *até que os recursos computacionais/paciência para esperar se esgotem.*

O problema do aumento do espaço de busca é um dos maiores problemas enfrentados na área de dedução automática. Para minimizá-lo, duas são as linhas de pesquisa:

- o desenvolvimento de estratégias e métodos mais eficientes, isto é, que minimizem o número de inferências realizadas no processo de prova,
- o desenvolvimento de estratégias de simplificação da fórmula propriamente dita. Esta simplificação dá-se pela retirada de redundâncias contidas na fórmula original ou à ela agregadas durante o processo de inferência.

Na seqüência apresentamos os métodos mais utilizados para dedução automática. Não é nosso intuito defini-los formalmente, mas apenas apresentar informalmente seus mecanismos de prova.

3.2.2 Método de Resolução

O método da Resolução, ou resolução simplesmente, introduzido por Robinson e baseado em trabalhos de Herbrand, Davis e Putnam, é um método de refutação: para provar que $G \models W$, prova-se que $H = G \cup \{\neg W\}$ é contraditório.

Para aplicar o método de resolução é necessário inicialmente transformar as fórmulas de H para a forma canônica conhecida como *forma normal conjuntiva*[6]. Nesta transformação, entre outras mudanças, toda a fórmula $\alpha \rightarrow \beta$ passa a ser representada como $\neg\alpha \vee \beta$. Quando em forma normal conjuntiva, a fórmula é um conjunto de cláusulas, onde cada cláusula é uma sentença com forma pré-estabelecida.

O método é baseado em uma regra de inferência única, chamada *regra de resolução*, uma generalização da regra *Modus Ponens*, que pode ser definida como segue:

$$\frac{(\alpha \vee \beta) \ \wedge \ (\neg\alpha \vee \delta)}{(\beta \vee \delta)}$$

[5]Isto é, o número de possibilidades de inferências.

[6]Definições e notações mais detalhadas são apresentadas em capítulo subseqüente.

onde a conclusão – $(\beta \vee \delta)$ é chamada de *resolvente.*

A resolução, é na realidade, um *princípio* que dá origem a diversos métodos nele baseados. Para ser um método, além da regra de inferência, uma estratégia de aplicação da regra de resolução deve ser definida. Para mostrar que a regra de resolução é correta, basta mostrar que o resolvente é uma conseqüência lógica das premissas – $\alpha \vee \beta$ e $\neg\alpha \vee \delta$. Já a prova de que a aplicação apenas da regra de resolução pode levar a métodos completos de refutação de teoremas é mais complexa (ver [22], e [27]). Apresentamos um método completo de prova baseado na resolução, o chamado *método de saturação de nível.* Dado um conjunto H de cláusulas, este método consiste na geração sucessiva dos conjuntos de cláusulas $H^0, H^1, H^2, \ldots$ definidos da seguinte maneira:

$$
\begin{aligned}
H^0 &= H \\
H^n &= \{\text{Resolventes de } C_1 \text{ e } C_2 \mid C_1 \in H^0 \cup \ldots \cup H^{n-1} \wedge C_2 \in H^{n-1}\} \\
&\ n = 1, 2, 3, \ldots
\end{aligned}
$$

onde C_i é uma cláusula. Caso o conjunto H seja insatisfazível, demonstra-se que sempre existirá um inteiro k tal que $NIL \in H^k$, onde NIL é uma cláusula vazia.

Ou seja, neste método, todos os possíveis resolventes de um conjunto de premissa são gerados. Se a cláusula vazia faz parte destes resolventes o teorema está refutado. Caso contrário, os resolventes gerados são unidos às premissas e novo ciclo de geração de resolventes se inicia. Estes ciclos são repetidos até que o teorema seja refutado (ou interrompido devido ao crescimento do conjunto de premissas).

Exemplo 6 *As premissas do exemplo 3, com o teorema negado, em forma normal conjuntiva tornam-se: (1)ou ela está sonhando ou ele está tocando, (2)ou ela não está acordada, ou ela não está sonhando, (3)ela está acordada e (4)ele não está tocando. Aplicando-se o método de saturação de nível, temos em H_0 a união destas premissas.*

A prova (e os resolventes gerados a cada ciclo de prova) é mostrada a seguir:

(1)	$Sonhando \vee Tocando$	*premissa*
(2)	$\neg Acordada \vee \neg Sonhando$	*premissa*
(3)	$Acordada$	*premissa*
(4)	$\neg Tocando$	*teorema*
(5)	$\neg Acordada \vee Tocando$	*de (1) e (2)*
(6)	$Sonhando$	*de (1) e (4)*
(7)	$\neg Sonhando$	*de (2) e (3)*
(8)	$Tocando$	*de (1) e (7)*
(9)	$\neg Acordada$	*de (2) e (6)*
(10)	$\neg Acordada$	*de (4) e (5)*
(11)	Nil	*de (6) e (7)*

e dá-se em dois ciclos: no primeiro, os resolventes (5), (6) e (7) são gerados e adicionados à H_0 dando origem à H_1. No segundo ciclo, os resolventes de H_1, (8), (9), (10) e (11) são gerados e, como entre eles encontra-se a cláusula vazia, o teorema está provado e o ciclo é interrompido.

No entanto, este método, apesar de correto e completo, é extremamente ineficiente e muitas vezes impraticável devido ao rápido crescimento dos conjuntos H^n. Ao vermos a árvore de prova do exemplo anterior:

(1)	$Sonhando \vee Tocando$	premissa
(2)	$\neg Acordada \vee \neg Sonhando$	premissa
(3)	$Acordada$	premissa
(4)	$\neg Tocando$	teorema
(6)	$Sonhando$	de (1) e (4)
(7)	$\neg Sonhando$	de (2) e (3)
(11)	Nil	de (6) e (7)

mesmo para este pequeno conjuntos de sentenças, percebe-se que muitos resolventes foram gerados desnecessariamente. Grande parte da pesquisa em dedução automática tem-se concentrado na busca de critérios para a definição de estratégias que tornem o método de resolução mais eficiente [38]. Estas estratégias podem ser divididas em dois tipos: de simplificação e de refinamento.

Estratégias de simplificação visam à eliminação de cláusulas que por algum motivo não sejam necessárias para o processo de prova. É importante que a eliminação dessas cláusulas não altere a insatisfazibilidade do conjunto original de cláusulas. Estas estratégias ajudam a reduzir a taxa de acréscimo de novas cláusulas a cada ciclo de inferência, mas não têm influência sobre os pares de cláusulas que são escolhidos como candidatos à resolução. As principais estratégias de simplificação são as seguintes [18]: eliminação de tautologias, de subsunções e de literais puros.

Estratégias de refinamento baseiam-se na constatação de que, dado um conjunto de cláusulas, nem todas resoluções possíveis a cada nível precisam ser realizadas para a prova ser alcançada. Cada estratégia define um critério diferente para a escolha das cláusulas candidatas à resolução. A maioria das estratégias de refinamento utilizadas são completas. No entanto, algumas estratégias não completas são utilizadas devido à sua eficiência, pois mesmo incompletas podem ser suficientemente poderosas para provar os teoremas desejados. Algumas das estratégias de refinamento são as seguintes: *conjunto de suporte*, *unitária*, *linear*, etc.

A grande vantagem do uso de métodos baseados em resolução em dedução automática é a facilidade de sua implementação. Esta facilidade deve-se ao uso da fórmula expressa em forma normal conjuntiva, que reduz o número de operadores da fórmula a apenas três – $\wedge$, $\vee$ e $\neg$, e a existência de uma regra única de inferência. Com uma única regra, as estratégias podem preocupar-se apenas com "quando aplicar" e não "que regra aplicar" o que reduz, de certa forma a complexidade do método.

3.2.3 Método de Tableaux

O Método de Tableaux, proposto por Smullyan a partir de trabalhos de Gentzen e Beth é, analogamente ao método de resolução, um método de prova por refutação. Este, porém, contém um grande número de regras de inferência:

- Regras para fórmulas conjuntivas:

$$\frac{A \wedge B}{\begin{array}{c}A\\B\end{array}} \qquad \frac{\neg(A \vee B)}{\begin{array}{c}\neg A\\\neg B\end{array}} \qquad \frac{\neg(A \to B)}{\begin{array}{c}A\\\neg B\end{array}}$$

- Regras para fórmulas disjuntivas:

$$\frac{A \vee B}{A \mid B} \qquad \frac{\neg(A \wedge B)}{\neg A \mid \neg B} \qquad \frac{A \to B}{\neg A \mid B}$$

- Regra para a negação:

$$\frac{\neg\neg A}{A}$$

- Regras para fórmulas quantificadas universalmente:

$$\frac{\forall x.A}{A\{x/t\}} \qquad \frac{\neg\exists x.A}{\neg A\{x/t\}} \qquad \text{onde } t \text{ é um termo.}$$

- Regras para fórmulas quantificadas existencialmente:

$$\frac{\exists x.A}{A\{x/\pi\}} \qquad \frac{\neg\forall x.A}{\neg A\{x/\pi\}} \qquad \text{onde } \pi \text{ é um parâmetro.}$$[7]

Estas regras são utilizadas para a construção de um *tableau lógico*, a partir de uma fórmula lógica, na forma de uma árvore. Um ramo de um tableau é dito *fechado* se ele contém uma contradição, isto é, duas fórmulas, ou sentenças, com sinais contrários, A e $\neg A$, onde A é uma fórmula qualquer. Um tableau cujos ramos são todos fechados é chamado *tableau fechado*. Como se trata de um método de prova por refutação, uma prova pelo método de tableaux de W, notada $\vdash W$, é um tableau fechado para $\neg W$.

Diferentemente da resolução, neste método a fórmula não precisa ser transformada para nenhuma forma normal. Uma prova em tableaux é mostrada no seguinte exemplo.

Exemplo 7 *Uma prova por tableaux para a fórmula do exemplo 4:*

$$\{(\neg Sonhando \rightarrow Tocando), (Acordada \rightarrow \neg Sonhando), Acordada\} \vdash Tocando$$

consiste no seguinte tableau, onde as linhas com o comentário R.I. correspondem a aplicações da regra de introdução de hipóteses:

<table>
<tr><td>(1)</td><td colspan="5">$\neg Tocando$</td><td>$\neg W$</td></tr>
<tr><td>(2)</td><td colspan="5">$(\neg Sonhando \rightarrow Tocando)$</td><td>$R.I.$</td></tr>
<tr><td>(3)</td><td colspan="3">$Sonhando$</td><td>de (2)</td><td>$Tocando$</td><td>de (2)</td></tr>
<tr><td>(4)</td><td colspan="3">$(Acordada \rightarrow \neg Sonhando)$</td><td>$R.I.$</td><td></td><td></td></tr>
<tr><td>(5)</td><td>$\neg Acordada$</td><td>de (4)</td><td>$\neg Sonhando$</td><td>de (4)</td><td></td><td></td></tr>
<tr><td>(6)</td><td>$Acordada$</td><td>$R.I.$</td><td></td><td></td><td></td><td></td></tr>
</table>

[7]Formalmente, é necessário definir uma nova linguagem $L^{\Pi}(\mathbf{P}, \mathbf{F}, \mathbf{C} \cup \Pi, \mathbf{V})$ onde Π é um novo conjunto de constantes, disjunto de $\mathbf{C}$, chamado *parâmetros.*

O maior problema do método de tableaux, além do grande número de regras de inferência, é seu não-determinismo: é possível, utilizando as regras para fórmulas quantificadas universalmente, construir um tableau infinito para um conjunto de fórmulas que, no entanto, é contraditório. Mesmo assim, com a introdução de algumas regras e estratégias para sua aplicação este problema pode ser evitado. Muitas vezes, também, a forma normal conjuntiva é utilizada para conferir eficiência ao método.

3.2.4 Método de Conexões

O Método das Conexões tem suas raízes nos métodos de Tableaux, Seqüentes de Gentzen e Dedução Natural e inspirou diversos outros métodos de dedução automática (por exemplo o *Consolution Method* [13]). Ele começou a ser desenvolvido, principalmente, por W. Bibel e, independentemente, por P.B. Andrews (com diferente terminologia) na metade da década de setenta [3].

Como seus predecessores, este método pode trabalhar com fórmulas genéricas da lógica de primeira ordem, isto é, elas não precisam estar representadas em forma normal canônica. Porém, seu uso em provadores automáticos de teoremas, normalmente exige a transformação da fórmula para sua forma normal disjuntiva[8], na qual uma fórmula é um conjunto de cláusulas duais. Restringimos nossa apresentação à "versão clausal" do método.

O método das conexões é um método para prova de uma fórmula em forma normal disjuntiva. Em linhas gerais, o critério de término, ou seja, de obtenção da prova, é a geração de um *spanning complementary mating* da fórmula. Um *mating* é um conjunto de conexões. Uma *conexão* é um par não ordenado de fórmulas atômicas com complementares, ou seja, com sinal contrário. Um *caminho* através de uma fórmula é um conjunto de fórmulas atômicas, onde cada uma foi tomada de uma cláusula dual. Finalmente, um mating *cobre (spans)* a fórmula (ou é uma *cobertura (spanning)* da fórmula) se cada caminho contém uma conexão deste mating.

Por se tratar de um método afirmativo, dada uma fórmula Δ e um teorema δ a ser provado, prova-se que $\Delta \rightarrow \delta$ é uma tautologia. Na transformação para a forma normal disjuntiva esta fórmula transforma-se em $\neg\Delta \vee \delta$.

Exemplo 8 *Por se tratar de um método afirmativo, dada uma fórmula Δ e um teorema δ a ser provado, prova-se que $\Delta \rightarrow \delta$ é uma tautologia. Na transformação para a forma normal disjuntiva esta fórmula transforma-se em $\neg\Delta \vee \delta$. Assim, as premissas do exemplo 3 em forma normal disjuntiva tornam-se:*

[8] A ser definida na seqüência do trabalho

- *ela não está sonhando e ele não está tocando* $(\neg Sonhando \wedge \neg Tocando)$ *ou,*
- *ela está acordada e ela está sonhando* $(Acordada \wedge Sonhando)$, *ou*
- *ela não está acordada* $(\neg Acordada)$, *ou*
- *ele está tocando* $(Tocando)$

onde cada premissa é uma cláusula dual. Nelas podemos identificar:

- *quatro caminhos:*
  - $\{\neg Sonhando, Acordada, \neg Acordada, Tocando\}$,
  - $\{\neg Sonhando, Sonhando, \neg Acordada, Tocando\}$,
  - $\{\neg Tocando, Acordada, \neg Acordada, Tocando\}$ *e*
  - $\{\neg Tocando, Sonhando, \neg Acordada, Tocando\}$;
- *três conexões:*
  - $\{Sonhando, \neg Sonhando\}$,
  - $\{Acordada, \neg Acordada\}$ *e*
  - $\{Tocando, \neg Tocando\}$;
- *o mating formado pelas três conexões cobre a fórmula pois cada caminho contém uma de suas conexões.*

O método das conexões pode ser considerado um método de prova de alto nível, já que ele busca uma propriedade global da fórmula, a "spaning mating", e não local como no caso da resolução (onde apenas cláusulas individuais são analisadas). Ainda diferentemente da resolução, cláusulas inferidas não são guardadas durante o processo de dedução, embora, para lógica de primeira ordem, muitas vezes a fórmula tenha que ser *expandida* pela duplicação de algumas de suas cláusulas duais.

3.3 Conclusão

Encerra-se aqui a introdução informal, na medida do possível, dos conceitos dedutivos da lógica utilizados na área de dedução automática. Nela foram apresentados os principais conceitos e definições da lógica considerados, na área de dedução automática, no desenvolvimento de *provadores automáticos de teoremas* (ATP - Automated Theorem Provers), isto é, sistemas computacionais (programas) que implementam um método de prova. Por fim, foram apresentados, em linhas gerais, os principais métodos de prova utilizados nestes provadores. Para escrevê-la, além vários textos lidos durante estes últimos anos, foram utilizados principalmente os livros *Handbook of Logic in Artificial Intelligence and Logic Programing - Deduction Methodologies* [17], *Symbolic Logic and Mechanical Theorem Proving* [11] e *Inteligência Artificial - Ferramentas e Teorias* [7]. Um agradecimento especial ao Prof. Guilherme Bittencourt, autor deste, pela autorização de "praticamente" copiar partes do texto de seu livro, especialmente nas subseções intituladas *Um pouco de teoria.*

Dados os conceitos gerais subjacentes ao desenvolvimento de qualquer provador automático de teoremas, nos capítulos seguintes são apresentados os conceitos, bem como os estudos desenvolvidos, relativos aos tópicos relevantes a este trabalho.

"Then you should say what you mean," the March Hare went on. "I do," Alice hastily replied; "at least I mean what I say, that's the same thing, you know." "Not the same thing a bit!" said the Hatter. "Why, you might just as well say that 'I see what I eat' is the same thing as 'I eat what I see'!"

Lewis Carroll, Alice's Adventures in Wonderland, 1865

Capítulo 4

Duas formas de dizer

Sendo apresentado o uso da lógica na IA, devemos, antes de iniciar o estudo do método dual propriamente dito, estudar mais detalhadamente as *formas normais canônicas*, conjuntiva e disjuntiva, nas quais uma fórmula lógica pode ser representada, sua simplificação e a transformação de uma para a outra. Em ambas operações a *subsunção* desempenha um papel relevante. A subsunção é uma relação lógica que expressa implicação entre elementos de uma fórmula. Devido à considerável redução da fórmula, e conseqüente redução do espaço de busca, a simplificação por subsunção é bastante utilizada por provadores automáticos de teoremas [14].

Por ter um papel relevante no método dual e por seu cálculo envolver certas sutilezas, consideramos oportuno dedicar, às formas normais, um capítulo exclusivo.

4.1 As formas normais

A forma sintática na qual uma fórmula lógica é representada desempenha um importante papel em áreas de estudo cujo formalismo subjacente é a linguagem lógica, como por exemplo nas áreas de *representação de conhecimento* e *prova automática de teoremas*. Nesta, especificamente, o uso de formas normais não apenas confere eficiência à automação de um método de prova como, muitas vezes, viabiliza esta automação. Métodos de prova baseados no método de *Resolução* [32, 11] requerem que a fórmula seja transformada para sua forma normal conjuntiva antes do processo de prova ser iniciado. O método de *Tableau Semântico* [35] é em si um algoritmo de transformação de uma fórmula para sua forma normal disjuntiva. Outros métodos mais recentes seguem o mesmo caminho: ou trabalham com fórmulas representadas em forma normal conjuntiva (e.g., [1]) ou são em si similares ao processo de transformação da mesma para sua forma normal disjuntiva (e.g., [3, 13]).

4.1.1 A definição das formas normais

Uma fórmula lógica qualquer pode ser representada em uma forma padrão bem definida, isto é, em *forma normal canônica.* Estas formas normais, além de padronizarem a fórmula, simplificam-na pelo uso restrito de operadores[1]. Na busca de maior eficiência, a maioria dos métodos de prova voltados para a prova automática de teoremas baseia-se nestas formas normais, as chamadas *forma normal conjuntiva* (ou *forma clausal*) e *forma normal disjuntiva* (ou *forma clausal dual*). O uso destas formas normais impõe aos métodos nelas baseados diversas características comuns. Maiores detalhes destas características podem ser encontrados em [51].

Para apresentá-las é necessário antes apresentar dois conceitos: disjunção e conjunção. Uma *disjunção* de fórmulas lógicas é a conexão destas pelo operador lógico $\vee$ e é notada por $[\ \ldots\ ]$. Já uma *conjunção* é a conexão de fórmulas lógicas pelo operador lógico $\wedge$ e é notada por $\langle \ldots \rangle$. Estas definições e notações podem ser encontradas em [15].

Transformar uma fórmula para sua forma normal conjuntiva é transformá-la em uma *conjunção de cláusulas*, também chamado *conjunto de cláusulas.* Uma cláusula é uma *disjunção de literais.* Um *literal* é uma fórmula atômica ou sua negação. Então uma fórmula está em forma normal conjuntiva, se ele está na forma

$$C_1 \wedge C_2 \wedge \ldots \wedge C_n$$

onde cada C_i é uma cláusula. Uma fórmula é uma cláusula se está na forma

$$L_1 \vee L_2 \vee \ldots \vee L_n$$

onde cada L_i é um literal. Como os conectivos "$\wedge$" e "$\vee$" são únicos no corpo do conjunto de cláusulas e no corpo das cláusulas respectivamente, os mesmos são omitidos na notação adotada. Assim, a forma conjuntiva de uma fórmula – W_c – é notada por

$$\langle C_1, C_2, \ldots, C_n \rangle$$

e uma cláusula C é notada por

$$[L_1, L_2, \ldots, L_n]$$

[1] Existem diversos subconjuntos de operadores lógicos que são completos no sentido de poderem expressar qualquer fórmula lógica, por exemplo, $\{\neg, \rightarrow\}$, $\{\neg, \wedge\}$, $\{\neg, \vee\}$ e $\{\neg, \wedge, \vee\}$, e qualquer fórmula lógica pode ser reescrita para qualquer um destes subconjuntos de operadores.

A forma normal disjuntiva é, de certa forma, simétrica à forma normal conjuntiva. Para sua definição, basta tomar-se a definição da forma conjuntiva e inverter-se conjunção por disjunção e vice-versa. Portanto, transformar uma fórmula para sua forma normal disjuntiva é transformá-la em uma *disjunção de cláusulas duais*, também chamado *conjunto de cláusulas duais*. Uma cláusula dual é uma *conjunção de literais*. Então uma fórmula está em forma normal disjuntiva, se ele está na forma

$$D_1 \vee D_2 \vee \ldots \vee D_n$$

onde cada D_i é uma cláusula dual. Uma fórmula é uma cláusula dual se está na forma

$$L_1 \wedge L_2 \wedge \ldots \wedge L_n$$

onde cada L_i é um literal. Novamente, os conectivos são omitidos na notação adotada e a forma disjuntiva de uma fórmula – W_d – é notada por

$$[D_1,D_2,\ldots,D_n]$$

e uma cláusula dual D é notada por

$$\langle L_1, L_2, \ldots, L_n \rangle$$

Uma fórmula representada em uma das formas pode ser transformada para a outra e, para tal, apenas a distributividade dos operadores $\vee$ e $\wedge$ é aplicada. Portanto, uma fórmula qualquer W, sua forma normal conjuntiva W_c e sua forma normal disjuntiva W_d são equivalentes: $W \Leftrightarrow W_c \Leftrightarrow W_d$. Algoritmos para transformação de uma fórmula qualquer para suas formas normais são bastante usuais (e.g. [28]). Estes algoritmos implicam a retirada de alguns operadores e quantificadores lógicos da fórmula para adaptá-la ao subconjunto de operadores estipulado pelas formas normais. A retirada dos quantificadores envolve cuidados, como a troca de nomes de variáveis e a introdução de *funções de Skolem* em substituição a variáveis quantificadas existencialmente [32] [33]. Além de sua definição, as formas normais apresentam outras duas características simétricas:

- **Escopo das Variáveis**

  Com a eliminação dos quantificadores universais e existenciais de uma fórmula, ao ser transformada para uma das formas normais, fica garantido que toda variável de uma cláusula ou cláusula dual é quantificada universalmente. Porém, como o quantificador universal distribui sobre o operador $\wedge$ mas não distribui sobre o operador $\vee$ [11], isto é:

$$\forall x(F_1(x) \wedge F_2(x)) \leftrightarrow \forall x_1(F_1(x_1)) \wedge \forall x_2(F_2(x_2))$$

$$\forall x(F_1(x) \vee F_2(x)) \not\leftrightarrow \forall x_1(F_1(x_1)) \vee \forall x_2(F_2(x_2))$$

onde $F_i(x)$ representa uma fórmula arbitrária contendo a variável x, o escopo das variáveis não é o mesmo nas duas formas.

Na forma conjuntiva, por ser uma conjunção de cláusulas, o escopo de cada variável é a própria cláusula. Assim, mesmo tendo o mesmo nome, variáveis de cláusulas diferentes não são as mesmas. Para evitar ambigüidades, é comum em provadores automáticos de teorema *renomear as variáveis* de modo que um mesmo nome de variável não se repita em mais de uma cláusula. Esta atribuição de nomes é um passo comum a todos algoritmos de transformação de fórmulas expressas em lógica de primeira ordem para sua forma normal conjuntiva [27].

Na forma disjuntiva, por ser uma disjunção de cláusulas duais, o escopo das variáveis é toda a fórmula e, portanto, suas variáveis não podem ser renomeadas.

- **Tautologias e Contradições**

Dois literais são *complementares* se eles unificam[2] e possuem sinal contrário. Por exemplo, os literais $P(x)$ e $\neg P(a)$ são complementares. Uma cláusula que contém literais complementares é dita *potencialmente tautológica.* Já uma cláusula dual que contém literais complementares é dita *potencialmente contraditória.*

Dois literais são *explicitamente complementares* se um é a negação do outro, ou seja se eles são iguais (unificam com substituição vazia) e possuem sinais contrários. Por exemplo, os literais $P(a)$ e $\neg P(a)$ são explicitamente complementares. Uma cláusula que contém literais explicitamente complementares é dita *tautológica* (ou é uma *tautologia*). Já uma cláusula dual que contém literais explicitamente complementares é dita *contraditória* (ou é uma *contradição*).

[2]Dois literais l_1 e l_2 unificam se existe uma substituição θ tal que $l_1\theta = l_2\theta$.

Na forma conjuntiva por ser uma conjunção de cláusulas, para a fórmula ser uma tautologia, todas as cláusulas têm de ser, independentemente, tautológicas. Portanto, a retirada de uma cláusula tautológica não altera o valor verdade da fórmula. Por outro lado, para a fórmula ser contraditória, basta apenas uma cláusula o ser.

Na forma disjuntiva por ser uma disjunção de cláusulas duais, para a fórmula ser uma tautologia, apenas uma cláusula dual tem de ser tautológica. Portanto, a retirada de uma cláusula dual contraditória não altera o valor verdade da fórmula. Por outro lado, para a fórmula ser contraditória, todas as cláusulas duais têm de ser, independentemente, contraditórias.

4.2 Simplificando a forma conjuntiva

Independente de seu uso, sempre encontramos explícita ou implicitamente o uso de procedimentos de simplificação destas fórmulas expressas em forma normal, basicamente pela eliminação de redundâncias. Diferentes tipos de redundância podem existir em um conjunto de cláusulas [20]:

- uma cláusula pode ser redundante por ser tautológica e, como já visto anteriormente, pode ser removida sem afetar o valor verdade do conjunto;

- uma cláusula pode ser redundante por conter literais *puros*, isto é, literais cujo complementar não ocorre no conjunto e, no caso de refutação, esta cláusula pode ser removida pois, certamente, esta cláusula não poderá ser tomada por contraditória;

- uma cláusula pode ser redundante por ser uma conseqüência lógica de algumas cláusulas do conjunto. Este tipo de redundância é usualmente difícil de detectar: o problema é obviamente indecidível para cláusulas em geral, uma vez que é equivalente à tarefa de provar teoremas. Normalmente este tipo de redundância não é tratado por provadores automáticos de teoremas;

- uma cláusula pode ser redundante por ser uma conseqüência lógica de uma única outra cláusula do conjunto. Mesmo a detecção deste caso – um subcaso do caso acima – é, em geral, indecidível. Porém, apesar do subproblema de testar se uma cláusula *subsume* outra ser decidível ainda é NP-completo. Ao teste desta redundância chamamos *subsunção externa* e é vista com mais detalhes no decorrer deste capítulo;

- algumas vezes literais em uma cláusula podem ser redundantes porque são implicados pelos outros literais da cláusula e por outras cláusulas do conjunto. Este teste é também equivalente à prova de um teorema, e normalmente não é tratado. Mas, algumas vezes um literal pode ser redundante por ser uma conseqüência lógica dos demais literais de mesma cláusula. Ao teste desta redundância chamamos *subsunção interna* e também é vista com mais detalhes na seqüência.

4.2.1 Subsunção externa na forma conjuntiva

A subsunção é um tipo de implicação entre fórmulas, em geral, facilmente reconhecível sintaticamente na forma conjuntiva de uma fórmula [14]. Na realidade existem mais de um tipo de subsunção e podem ser mais ou menos adequadas a determinados tipos de problemas [50]. A subsunção que apresentamos é, normalmente[3], denominada na literatura como *θ-subsunção* e é definida como segue:

> uma cláusula C_1 subsume uma cláusula C_2 – notado $C_1 \succ C_2$ ou $C_1 \succ_\theta C_2$[4] – se existe uma substituição θ tal que $C_1\theta \subseteq C_2$.

Assim, por exemplo:

- $[P] \succ [P, Q]$;
- $[P(x), R(a)] \succ_\theta [P(a), R(a), Q(a)]$ com $\theta = \{(x/a)\}$;
- $[P(x), R(a)] \not\succ [P(a), R(y)]$;
- $[P(x, y), P(y, x)] \succ_\theta [P(x, x)]$ com $\theta = \{(y/x)\}$;
- $[P(x)] \succ_\theta [P(y)]$ com $\theta = \{(x/y)\}$ e $[P(y)] \succ [P(x)]$ com $\theta = \{(y/x)\}$.

Devido à ausência de variáveis, no caso proposicional, a relação de subsunção é determinada a partir da igualdade entre literais. Já na lógica de primeira ordem, a unificação entre literais é condição necessária à relação de subsunção. Necessária mas não suficiente, já que a substituição pode ser aplicada em apenas uma das cláusulas, como mostra o terceiro exemplo acima onde todos os literais unificam e a subsunção não ocorre.

[3]Não existe consenso quanto à denominação.

[4]Dependendo do contexto usamos uma ou outra notação.

Percebe-se também que cláusulas que são *variantes alfabéticas*[5], ou simplesmente *variantes*, subsumem-se mutuamente.

É relativamente fácil verificar que a subsunção entre cláusulas implica a implicação entre cláusulas, isto é, sendo C_1 e C_2 cláusulas de uma fórmula, $(C_1 \succ C_2) \rightarrow (C_1 \rightarrow C_2)$. Porém, deve-se atentar que o contrário não se verifica. Por exemplo, para $C_1 = [\neg P(x,y), \neg P(y,z), P(x,z)]$ e $C_2 = [\neg P(a,b), \neg P(b,c), \neg P(c,d), P(a,d)]$, tem-se que $C_1 \rightarrow C_2$ mas $C_1 \not\succ C_2$. Em alguns contextos pode ser interessante testar pela implicação entre cláusulas e não apenas pela subsunção. Porém, devido à complexidade de seu cálculo, este não é o caso usualmente tratado em prova automática de teoremas. Para maiores esclarecimentos sugerimos consultar o artigo *Subsumption and Implication* [19].

O problema da remoção de todas as cláusulas subsumidas de uma fórmula é algumas vezes chamado de *condensação* [20] e é definida como segue:

> uma fórmula em forma conjuntiva está condensada se possui apenas cláusulas condensadas[6] não tautológicas e não subsumidas por nenhuma outra cláusula da fórmula.

Pode-se dizer que a forma condensada de uma fórmula em forma conjuntiva *é única*, pois quaisquer duas formas condensadas de uma mesma fórmula são variantes (basta, portanto, renomear as variáveis para torná-las iguais).

A remoção de cláusulas subsumidas é tida como um dos *problemas básicos de pesquisa a resolver* [51]. Como o teste de subsunção entre cláusulas deve ser repetido muitas vezes durante o processo de prova, seu custo computacional pode ser bastante alto. De fato, devido a este alto custo computacional, alguns autores afirmam que a remoção de cláusulas subsumidas não melhoram a eficiência de provadores automáticos de teorema [48]. Outros, por sua vez, afirmam que esta remoção é uma das mais efetivas dentre as técnicas utilizadas para minimização do espaço de busca [2, 39]. E esta parece ser a opinião mais aceita pela comunidade, uma vez que os mais bem sucedidos provadores automáticos de teoremas realizam esta remoção.

A redução do custo da detecção de cláusulas subsumidas pode ser abordada de duas formas: aumentando a eficiência do próprio teste de subsunção entre fórmulas e reduzindo o número de testes efetuados entre cláusulas [21]. Seguindo a primeira abordagem, muitos algoritmos podem ser encontrados na literatura, por exemplo nas

[5] Duas cláusulas C e D são variantes alfabéticas se diferem apenas nos nomes das variáveis, isto é, se e somente se existem substituição θ e σ tal que $C\theta = D$ e $D\sigma = C$.

[6] Definido na próxima subseção.

referências apresentadas neste capítulo. Já a segunda abordagem não é a mais freqüentemente encontrada na literatura.

No artigo *A Concurrent Algorithm for Logical Subsumption* apresentado no *LAPTEC 2001 - Segundo Congresso de Lógica Aplicada à Tecnologia* [44] e publicado na série *Frontiers in Artificial Intelligence and its Applications - IOS Press* [45], apresentamos um algoritmo que reduz o número de testes necessários à eliminação de cláusulas subsumidas. Devido à sua estrutura subjacente, em forma de um hipercubo[7], este algoritmo é naturalmente concorrente e adequado à implementação em paralelo. O algoritmo não acrescenta nenhuma melhoria ao teste de subsunção entre cláusulas propriamente dito, no entanto esta pode ser considerada uma de suas maiores vantagens: o algoritmo apenas preocupa-se em reduzir o número de testes necessários; no entanto o teste de subsunção entre cláusulas propriamente dito, pode seguir qualquer algoritmo desejado. Com isto, o algoritmo pode somar ganhos nas duas abordagens na detecção da subsunção.

4.2.2 Subsunção interna na forma conjuntiva

Chamamos de subsunção interna a subsunção de literais de uma cláusula por outros de mesma cláusula. De uma forma geral, dada uma cláusula C, um literal $l_1 \in C$ subsume um literal $l_2 \in C$ – notado $l_1 \succ^{or} l_2$ ou $l_1 \succ^{or}_{\theta} l_2$ – se existe uma substituição θ tal que $l_1 = l_2\theta$. No entanto, como uma cláusula é uma disjunção de literais e o quantificador universal não distribui sobre o operador $\vee$, o fato de um literal ser isoladamente subsumido por outro da cláusula não é condição suficiente para considerá-lo irrelevante.

Assim, por exemplo, na cláusula $[P(x), P(a)]$ o literal $P(x)$ é subsumido pelo literal $P(a)$ com $\theta = \{(x/a)\}$ e, neste caso, $P(x)$ é redundante na cláusula e pode ser eliminado. Já na cláusula $[P(x), Q(x), P(a), Q(b)]$, apesar de $P(x)$ ser subsumido por $P(a)$ e $Q(x)$ ser subsumido por $Q(b)$ nenhum deles é irrelevante na cláusula. Isto ocorre devido à *não linearidade*[8] da variável x nesta cláusula.

Como o escopo das variáveis na forma disjuntiva é a cláusula, toda ela deve ser considerada na detecção de literais irrelevantes. O problema da eliminação de todos os literais subsumidos de uma cláusula é comumente chamado de *condensação*[9] (ou,

[7]Um hipercubo de dimensão n é um grafo onde os vértices são todos os elementos do conjunto de n-tuplas $(k_1, ..., k_n)$ onde $k_i \in \{0,1\}$ e onde quaisquer dois vértices cuja representação difere em apenas uma posição são conectados entre si [31].

[8]Uma variável é linear se ocorre em apenas um literal de uma cláusula.

[9]Todas as definições relativas à condensação foram extraídas de [20].

algumas vezes, fatoração) e é definido como segue:

> uma cláusula é dita condensada se não subsume nenhum de seus subconjuntos de literais.

O problema aqui é semelhante ao da detecção de subsunção entre cláusulas: consiste em detectar, em uma dada cláusula C, subconjuntos de literais subsumidos pela cláusula. Uma definição mais algorítmica é dada a seguir:

- a *condensação* de uma cláusula C é o subconjunto C' de C de mínima cardinalidade, tal que, $C \succ C'$;
- seja $Conds(C)$ o conjunto de todas as condensações da cláusula C, C é condensada se $Conds(C) = \{C\}$.

Eis alguns exemplos:

- $Conds([P(x,y), P(x,x)]) = \{[P(x,x)]\}$;
- $Conds([P(x,y), P(a,y)]) = \{[P(a,y)]\}$;
- $Conds([P(x,f(a)), P(y,f(a))]) = \{[P(x,f(a))], [P(y,f(a))]\}$;
- $Conds([P(x,y), P(y,z), P(x,z)]) = \{[P(x,y), P(y,z), P(x,z)\}$ portanto a cláusula já é condensada.

Um importante resultado pode ser verificado no terceiro exemplo acima: *duas condensações quaisquer de uma cláusula são variantes.* Outro importante resultado é que toda a condensação C' de C é equivalente à C, uma vez que C' é um subconjunto de C e toda cláusula é implicada por cada um de seus subconjuntos. Por outro lado, $C \succ C'$ e, portanto, também implica C'. Como C e C' são equivalentes, C *pode ser substituída por* C' *independentemente das demais cláusulas da fórmula.*

Condensação é uma importante técnica para remover (parcialmente) redundâncias internas às cláusulas mas, infelizmente, bastante cara em termos computacionais (NP-hard). Parcialmente pois, se ao invés de considerar a subsunção na definição, considerarmos a implicação entre cláusulas, temos uma técnica mais robusta para eliminação de redundâncias mas ainda indecidível. Por isto, em provadores automáticos encontramos o uso apenas da técnica de condensação e não da condensação forte.

4.3 Simplificando a forma disjuntiva

Nosso interesse na forma normal disjuntiva simplificada tem origem na sua aplicação no método dual, o qual utiliza informações sobre a ocorrência de cada literal, e dos literais a ele associados por relações de subsunção, em ambas as formas duais para a obtenção da prova. Como o método utiliza ambas as formas normais simultaneamente, é importante que estas sejam condensadas, de forma a otimizar o mecanismo de prova.

Assim como na forma conjuntiva, diferentes tipos de redundâncias podem ocorrer em um conjunto de cláusulas duais, como por exemplo:

- uma cláusula dual pode ser redundante por ser contraditória e, como já visto anteriormente, pode ser removida sem afetar o valor verdade do conjunto;
- uma cláusula dual pode ser redundante por ser subsumida por outra do conjunto (subsunção externa);
- um literal de uma cláusula dual pode ser redundante por ser uma conseqüência lógica dos demais literais de mesma cláusula (subsunção interna).

Destes, a remoção de cláusulas duais contraditórias é um problema simples de tratar e não necessita de maiores esclarecimentos. Já a detecção das redundâncias que envolvem a relação de subsunção na forma disjuntiva, é um problema bastante complexo que merece maior atenção.

De uma forma geral, dada uma cláusula W_d em forma disjuntiva, uma cláusula dual $D_1 \in W_d$ subsume uma cláusula dual $D_2 \in W_d$ – notado $D_1 \succ^{or} D_2$ ou $D_1 \succ^{or}_{\theta} D_2$ – se existe uma substituição θ tal que $D_1 = D_2\theta$. No entanto, como W_d é uma disjunção de cláusulas duais, e o operador $\forall$ não distribui sobre o operador $\vee$, o fato de uma cláusula dual ser isoladamente subsumida por outra da fórmula não é condição suficiente para considerá-la irrelevante.

Da mesma forma, dada uma cláusula dual D, um literal $l_1 \in D$ subsume um literal $l_2 \in D$ – notado $l_1 \succ l2$ ou $l_1 \succ_{\theta} l2$ – se existe uma substituição θ tal que $l_1\theta = l_2$. No entanto, como o escopo das variáveis de D é toda a fórmula, e W_d é uma disjunção de cláusulas duais, o fato de um literal de uma cláusula dual ser isoladamente subsumido por outro da mesma cláusula não é condição suficiente para considerá-la irrelevante.

Embora a simplificação da forma conjuntiva de fórmulas da lógica de primeira ordem, devido à sua aplicação em provadores baseados em resolução, é assunto facilmente encontrado na literatura, este não é o caso da simplificação da forma disjuntiva. Por isto, para a simplificação desta, nos baseamos na adaptação dos conceitos utilizados para a simplificação daquela, explorando as relações existentes entre as duas representações.

4.3.1 A relação entre as formas normais

Um ponto importante a salientar é a relação existente entre as representações de uma fórmula. Sabemos que uma forma normal é obtida pela distribuição dos literais da outra. Assim, dados $W_c \Leftrightarrow W_d$, respectivamente formas normais conjuntiva e disjuntiva de uma fórmula, onde as variáveis de W_c encontram-se devidamente renomeadas[10] e W_d é o resultado da distribuição dos literais de W_c, tem-se que:

- sendo $[l_1, l_2, \ldots]$ uma cláusula de W_c, $\langle l_1, \ldots \rangle$ e $\langle l_2, \ldots \rangle$ são cláusulas duais de W_d e, se l_1 e l_2 não ocorrem em nenhuma outra cláusula de W_c[11], então W_c $\langle l_1, l_2, \ldots \rangle$ não é uma cláusula dual de W_d;
- sendo $[l_1, \ldots]$ e $[l_2, \ldots]$ cláusulas de W_c, $\langle l_1, l_2, \ldots \rangle$ é uma cláusula dual de W_d;
- sendo $\langle l_1, l_2, \ldots \rangle$ uma cláusula dual de W_d, $[l_1, \ldots]$ e $[l_2, \ldots]$ são cláusulas de W_c;
- sendo $\langle l_1, \ldots \rangle$ e $\langle l_2, \ldots \rangle$ cláusulas duais de W_d, $[l_1, l_2, \ldots]$ é uma cláusula de W_c;

e sabendo que sendo l_1 e l_2 dois literais quaisquer, se l_1 subsume l_2 em uma das formas normais de uma fórmula, então l_2 subsume l_1 na outra forma, podemos afirmar que:

- a toda subsunção interna[12] verificada em uma das formas, W_c ou W_d, corresponde uma subsunção externa[13] na outra forma; e,
- a toda subsunção externa verificada em uma das formas, W_c ou W_d, corresponde uma subsunção interna na outra.

4.3.2 Sutilezas da lógica de primeira ordem

Muitas sutilezas envolvidas na simplificação das formas normais devem-se aos seguintes fatos. Por um lado, pela distributividade do quantificador $\forall$ sobre o operador $\wedge$, as variáveis que ocorrem em cada cláusula da forma normal conjuntiva podem (e, na prática, devem) ser independentemente renomeadas. Por outro lado, na forma normal

[10] Nenhum literal aberto de cláusulas diferentes compartilha de mesmo nome.

[11] O que poderia acontecer apenas no caso de literal fechado, uma vez que as variáveis de W_c estão devidamente renomeadas.

[12] Subsunção entre literais de mesma cláusula (dual).

[13] Subsunção entre literais de cláusulas (duais) diferentes.

disjuntiva, as variáveis que ocorrem nas cláusulas duais não podem ser renomeadas, pois devemos respeitar a não distributividade do quantificador $\forall$ sobre o operador $\vee$.

Sendo assim, o escopo das variáveis nesta forma normal abrange todo o conjunto de cláusulas duais. Disto segue que o escopo de aplicação de uma substituição de uma variável x por um termo t, na forma conjuntiva é restrito à cláusula onde x ocorre e, na forma disjuntiva é o conjunto de todas as cláusulas duais que contêm uma ocorrência de x. Assim, ao tratarmos de subsunção entre cláusulas temos de levar em conta que o escopo de uma substituição θ envolvida na subsunção de uma cláusula C_2 por uma cláusula C_1 ($C_1\theta \subseteq C_2$) é a cláusula C_1; e o de uma substituição θ envolvida na subsunção de uma cláusula dual D_2 por uma cláusula dual D_1 ($D_1 \subseteq D_2\theta$) é o conjunto das cláusulas duais que contenham ocorrências das variáveis que ocorrem em θ.

Outro fato a ser levado em consideração é que a substituição θ envolvida na subsunção entre as cláusulas é o resultado da combinação das substituições das subsunções entre seus literais. Por exemplo, considere o seguinte par de cláusulas:

$$C_1 = [l_1[x_1], l_2[x_2]] \text{ e } C_2 = [l_1[t_1], l_2[t_2]]$$

onde $l_i[\ ]$ representa literais que dependem dos termos interiores aos colchetes, x_i é uma variável e t_i é um termo não variável e onde $l_1[x_1] \succ^{or}_{\theta_1} l_1[t_1]$ com $\theta_1 = \{x_1/t_1\}$ e $l_2[x_2] \succ^{or}_{\theta_2} l_2[t_2]$ com $\theta_2 = \{x_2/t_2\}$ (esta definição pode ser generalizada se habilitarmos os símbolos x_i e t_i a representarem n-tuplas de variáveis e termos, respectivamente). Temos que $C_1 \succ_\theta C_2$, onde θ é o resultado da combinação de θ_1 e θ_2. No caso de x_1 e x_2 serem a mesma variável (ou terem variáveis em comum no caso de representarem n-tuplas de variáveis), temos que $C_1 \succ C_2$ somente se as substituições θ_1 e θ_2 forem compatíveis, ou seja, se t_1 e t_2 forem unificáveis. Caso contrário, a subsunção não se verifica, já que o cálculo de θ é impossível devido ao *conflito* existente entre as substituições que unificam os literais envolvidos na subsunção.

Conflito de subsunção interna

Dizemos que uma fórmula apresenta um *conflito de subsunção interna* se:

- ela contém em sua forma normal conjuntiva uma cláusula condensada com dois literais l_1 e l_2 tal que l_1 subsume l_2, e
- as variáveis que ocorrem em l_1 não são lineares na cláusula.

O caso geral de conflito de subsunção interna é dado por cláusulas condensadas da seguinte forma:

$$[l_1[x], l_1[t], \ldots, l_2[x], \ldots]$$

onde, devido a não distributividade do quantificador $\forall$ sobre o operador $\vee$, não podemos eliminar o literal $l_1[x]$ da cláusula, o que seria possível se suas variáveis fossem lineares.

Durante a construção da forma disjuntiva de tais fórmulas encontraremos cláusulas duais tais como:

$$\langle l_1[x], \ldots \rangle \quad e \quad \langle l_1[t], \ldots \rangle$$

que são idênticas exceto pelos literais indicados. Mesmo assim, a última não pode subsumir a anterior porque suas variáveis não são lineares no conjunto de cláusulas duais. Devemos observar que na forma normal disjuntiva a linearidade deve levar em conta todo o conjunto de cláusulas duais. Então, se desejamos simplificar a forma normal disjuntiva de uma fórmula, devemos fazer esta verificação de linearidade antes de qualquer eliminação de cláusula dual subsumida.

Exemplo 9 *Um dos exemplos mais triviais de fórmula onde ocorre subsunção interna pode ser dado pelo seguinte conjunto de cláusulas:*

$$\langle\ [P(x), P(a), Q(x)], [R(b)]\ \rangle$$

cuja forma disjuntiva é dada por:

$$[\ \langle R(b), P(x)\rangle, \langle R(b), P(a)\rangle, \langle R(b), Q(x)\rangle\ ]$$

onde a última cláusula dual contém a variável x, criando a condição de não linearidade que impede a subsunção da primeira cláusula dual pela segunda.

Conflito de subsunção externa

Dizemos que uma fórmula apresenta um *conflito de subsunção externa* se:

- ela contém em sua forma normal conjuntiva duas cláusulas C_1 e C_2 tais que existam literais $l_1, l_1' \in C_1$ e $l_2, l_2' \in C_2$ (onde literais l_i e l_i' podem ser os mesmos), e substituições θ e θ', com as seguintes propriedades: $l_1\theta = l_2$ e $l_1'\theta' = l_2'$, e
- as substituições θ e θ' não podem ser combinadas entre si.

Alguns casos de conflito de subsunção externa são mostrados na tabela 4.1. Durante a construção da forma normal disjuntiva de tais fórmulas encontraremos cláusulas duais

C_1	C_2
$[l_1[x], l_2[x], \ldots]$	$[l_1[t_1], l_2[t_2], \ldots]$
$[l[x, y], \ldots]$	$[l[t_1, t_2], l[t'_1, t'_2], \ldots]$
$[l[x, y], l[y, x], \ldots]$	$[l[t_1, t_2], \ldots]$

Tabela 4.1: Conflito de subsunção externa

tais como:

$$\langle l_1, l_2, \ldots \rangle \quad e \quad \langle l'_1, l'_2, \ldots \rangle$$

nas quais os literais l_2 e l'_2 não podem ser eliminados devido à incompatibilidade das substituições θ e θ'. Se desejamos simplificar a forma normal disjuntiva, antes de cada eliminação de um literal de uma cláusula dual por subsunção, é necessária a verificação da existência de outra relação de subsunção no conjunto de cláusulas duais envolvendo as mesmas variáveis e, se for o caso, a verificação da compatibilidade das substituições envolvidas. É interessante notar que, neste caso, não é apenas a linearidade das variáveis que deve ser verificada mas a ocorrência de conflitos em diferentes cláusulas duais.

Alguns exemplos ajudam a esclarecer o problema.

Exemplo 10 *Considere a fórmula dada pelo seguinte conjunto de cláusulas:*

$$\langle\ [P(x), Q(x)], [P(a), Q(b)]\ \rangle$$

onde a primeira cláusula não subsume a segunda já que as substituições envolvidas nas subsunções – $P(x) \succ_{\theta_1} P(a)$ *e* $Q(x) \succ_{\theta_2} Q(b)$ *– não são compatíveis. Na forma conjuntiva é fácil verificar que não podemos habilitar ambas subsunções porque, neste caso, a segunda cláusula seria subsumida pela primeira e a fórmula reduzir-se-ia simplesmente a* $\langle\ [P(x), Q(x)]\ \rangle$*, o que é claramente incorreto, já que esta fórmula não é equivalente à fórmula original dada. Notamos que na forma disjuntiva daquele conjunto de cláusulas:*

$$[\ \langle P(x), P(a)\rangle, \langle Q(x), Q(b)\rangle, \langle Q(x), P(a)\rangle, \langle P(x), Q(b)\rangle\ ]$$

a verificação não é tão simples. Vemos que as mesmas subsunções externas existentes naquela forma encontram-se, agora, como subsunções internas. Se efetuarmos ambas simplificações por subsunção, teríamos a fórmula reduzida a $[\ \langle P(x)\rangle, \langle Q(x)\rangle\ ]$*, que é equivalente à* $\langle\ [P(x), Q(x)]\ \rangle$ *e, portanto, incorreta.*

Proibindo-se as subsunções, ficamos com a forma normal disjuntiva anterior, que *é correta mas não é condensada, pois existem formas de simplificá-la por eliminação de subsunção, como veremos a seguir. Sabemos que as substituições envolvidas nas subsunções são incompatíveis, por isto podemos pensar em não utilizá-las simultaneamente. Se, ao invés de proibirmos a eliminação de ambas subsunções, optarmos por proibir apenas uma das subsunções, esta fórmula disjuntiva pode ser simplificada de duas formas diferentes:*

$$[\ \langle P(x), P(a)\rangle, \langle Q(x)\rangle, \langle P(x), Q(b)\rangle\ ]$$

$$[\ \langle P(x)\rangle, \langle Q(x), Q(b)\rangle, \langle P(a), Q(x)\rangle\ ]$$

No exemplo anterior torna-se claro que pelo menos uma das subsunções deve ser proibida. Caso contrário, a segunda cláusula desapareceria por subsunção mudando, assim, o significado da fórmula de uma maneira óbvia. No próximo exemplo, a situação não é tão simples.

Exemplo 11 *Considere a fórmula dada pelo seguinte conjunto de cláusulas:*

$$\langle\ [P(x), Q(x), R(a)], [P(a), Q(b)]\ \rangle$$

neste caso, se habilitarmos ambas subsunções, a segunda cláusula não desaparece. Mas, se usarmos a distribuição e subsunção irrestrita, obteremos a seguinte forma normal disjuntiva:

$$[\ \langle P(x)\rangle, \langle Q(x)\rangle, \langle P(a), R(a)\rangle, \langle Q(b), R(a)\rangle\ ]$$

que não é correta por não ser equivalente à forma conjuntiva dada. Certamente, se proibirmos ambas subsunções obteremos uma forma disjuntiva correta, pois estaremos operando apenas com a distributividade dos operadores. Mas, novamente, a forma encontrada não será condensada. Pela proibição de umas das subsunções envolvidas no conflito, podemos obter duas formas disjuntivas condensadas:

$$[\ \langle P(x), P(a)\rangle, \langle Q(x)\rangle, \langle P(x), Q(b)\rangle, \langle P(a), R(a)\rangle, \langle Q(b), R(a)\rangle\ ]$$

$$[\ \langle P(x)\rangle, \langle Q(x), Q(b)\rangle, \langle Q(x), P(a)\rangle, \langle P(a), R(a)\rangle, \langle Q(b), R(a)\rangle\ ]$$

Nos dois exemplos prévios, existiam apenas duas subsunções internas que poderiam ser proibidas isoladamente e a escolha de uma ou outra nos levaria a uma forma disjuntiva simplificada condensada de mesmo tamanho. Porém, este não é sempre o caso, como mostra o próximo exemplo.

Exemplo 12 *Considere a fórmula dada pelo seguinte conjunto de cláusulas:*

$$\langle\ [P(x_0, x_1), P(x_1, x_2), P(x_2, x_3)], [P(a, b), R(c)]\ \rangle$$

a menor forma normal disjuntiva é obtida quando escolhemos proibir a subsunção $P(x_1, x_2) \succ P(a, b)$, *porque esta quebra a ligação entre as variáveis de* $P(x_0, x_1)$ *e* $P(x_2, x_3)$, *que estão, assim, habilitadas a subsumir livremente* $P(a, b)$. *Neste caso, temos o seguinte conjunto de cláusulas duais a seguinte forma disjuntiva condensada (a):*

$$[\ \langle P(x_0, x_1)\rangle, \langle P(x_2, x_3)\rangle, \langle P(x_1, x_2), P(a, b)\rangle, \langle R(c), P(x_1, x_2)\rangle\ ]$$

Se escolhermos proibir a subsunção $P(x_0, x_1) \succ P(a, b)$, *então temos de proibir também, ou a subsunção* $P(x_1, x_2) \succ P(a, b)$, *obtendo a forma disjuntiva condensada (b):*

$$\begin{aligned}&[\ \langle P(x_0, x_1), P(a, b)\rangle, \langle P(x_2, x_3)\rangle, \langle P(x_1, x_2), P(a, b)\rangle, \langle R(c), P(x_0, x_1)\rangle,\\ &\langle R(c), P(x_1, x_2)\rangle\ ]\end{aligned}$$

ou a subsunção $P(x_2, x_3) \succ P(a, b)$, *obtendo a seguinte forma disjuntiva condensada (c):*

$$\begin{aligned}&[\ \langle P(x_0, x_1), P(a, b)\rangle, \langle P(x_2, x_3), P(a, b)\rangle, \langle P(x_1, x_2)\rangle, \langle R(c), P(x_0, x_1)\rangle,\\ &\langle R(c), P(x_2, x_3)\rangle\ ]\end{aligned}$$

onde, claramente, as duas últimas formas disjuntivas condensadas calculadas (b e c) são maiores que a primeira (a).

4.3.3 A simplificação propriamente dita

Foram mostradas as dificuldades envolvidas na simplificação da forma disjuntiva. De fato, todas estas dificuldades advêm da não distributividade do operador $\forall$ sobre o operador $\vee$. Como já dito, nada encontramos na literatura relativo à simplificação desta forma. Porém, podemos apresentar alguns resultados obtidos até o momento.

Subsunção externa na forma disjuntiva

Devido à relação existente entre as formas normais, é razoável dizer que o problema da subsunção externa (entre cláusulas duais) na forma disjuntiva é similar ao problema da subsunção interna (condensação) na forma conjuntiva. Sendo assim, podemos estender a definição de condensação para a forma disjuntiva:

> uma fórmula em forma normal disjuntiva é dita condensada se possui apenas cláusulas duais condensadas não contraditórias e não subsume nenhum de seus subconjuntos de cláusulas duais.

O problema aqui consiste em detectar, em um dado W_d, subconjuntos de cláusulas duais subsumidos pela fórmula W_d. Uma definição mais algorítmica é dada a seguir:

- a *condensação* de uma fórmula em forma normal disjuntiva W_d é o subconjunto W'_d de W_d de mínima cardinalidade, tal que, $W_d \succ W'_d$;
- seja $Conds(W_d)$ o conjunto de todas as condensações de W_d, W_d é condensada se $Conds(W_d) = \{W_d\}$.

Definimos que uma fórmula W_d, representada em sua forma disjuntiva, subsume uma fórmula W'_d, também representada em sua forma disjuntiva, com a substituição θ – notado $W_d \succ_\theta W'_d$ ou, simplesmente, $W_d \succ W'_d$ – quando toda a cláusula dual de $W_d\theta$ é subsumida por alguma cláusula dual de W'_d, isto é

$$\forall D \in W_d\theta,\ \exists D' \in W'_d\ (D' \subseteq D)$$

Para esclarecer, apresentamos alguns exemplos:

- $Conds([\langle P(a)\rangle, \langle P(x)\rangle]) = \{[\langle P(a)\rangle]\}$;
- $Conds([\langle P(a)\rangle, \langle P(x), Q(b)\rangle, \langle R(a), R(b)\rangle]) = \{[\langle P(a)\rangle, \langle R(a), R(b)\rangle]\}$;
- $Conds([\langle P(x), R(b)\rangle, \langle Q(x), R(b)\rangle, \langle P(a), R(b)\rangle]) =$ $\{[\langle P(x), R(b)\rangle, \langle Q(x), R(b)\rangle, \langle P(a), R(b)\rangle]\}$ e, portanto a fórmula é condensada.

Toda a condensação W'_d de W_d é equivalente à W_d, uma vez que W'_d é um subconjunto de W_d e toda fórmula disjuntiva é implicada por cada um de seus subconjuntos. Por outro lado, $W_d \succ W'_d$ e, portanto, também implica W'_d. Sendo equivalentes, pode-se substituir W_d pela sua forma condensada W'_d.

Por envolver toda a fórmula, o cálculo da condensação de uma fórmula disjuntiva pode ser bastante caro em termos computacionais. Ele pode, no entanto, ser simplificado. Se considerarmos que:

- as únicas cláusulas duais modificadas pela substituição envolvida na condensação, são aquelas nas quais as variáveis da substituição ocorre;

- uma fórmula disjuntiva é que um conjunto de cláusulas duais e é equivalente à sua forma condensada;

então, uma fórmula disjuntiva W_d é equivalente à fórmula resultante da condensação de apenas um subconjunto de W_d, isto é: seja S um subconjunto de W_d tem-se que $W_d \leftrightarrow (\ W_d - S\ ) \cup Conds(S)$.

Portanto podemos considerar para a condensação, apenas as cláusulas concorrentes à subsunção com uma substituição θ e as demais nas quais as variáveis de θ ocorrem (uma vez que as cláusulas duais, nas quais estas variáveis não ocorrem, não se modificam com a aplicação de θ e, certamente, elas estão contidas em W_d').

Exemplo 13 *Dado* $W_d = [\langle P(a)\rangle, \langle P(x), Q(b)\rangle, \langle R(a), R(b)\rangle]$*, as cláusulas duais concorrentes à subsunção são as duas primeiras com* $\theta = \{x/a\}$*. Como a variável* x *(de* θ*) não ocorre em nenhuma outra cláusula dual, podemos considerar apenas o subconjunto* $S = \{\langle P(a)\rangle, \langle P(x), Q(b)\rangle\}$ *de* W_d *para sua condensação. Neste caso temos que* $Conds(S) = \{\langle P(a)\rangle\}$ *e então,* $W_d \leftrightarrow (\{\langle R(a), R(b)\rangle\}) \cup \{\langle P(a)\rangle\}$*, isto é* $W_d \leftrightarrow [\langle P(a)\rangle, \langle R(a), R(b)\rangle]$.

Subsunção interna na forma disjuntiva

Se seguíssemos o mesmo raciocínio, deveríamos afirmar que a subsunção interna na forma disjuntiva é similar à subsunção externa na forma conjuntiva. "Jein", como diriam os alemães, isto é, sim e não. "Sim" porque, como vimos, o conflito que pode existir na subsunção entre duas cláusulas se reflete na subsunção interna de uma cláusula dual. E "não", porque na forma conjuntiva o conflito pode ser identificado considerando-se apenas as duas cláusulas envolvidas na subsunção e, na forma disjuntiva, como o escopo das variáveis é toda a fórmula, deve-se considerar a fórmula por inteiro ao tratar da subsunção interna de uma de suas cláusulas duais. Considerando estas características definimos:

> uma cláusula dual de W_d é dita condensada se a ocorrência de dois literais l_1 e l_2 tal que $l_1 \succ_{\theta_1} l_2$ implica θ_1 ser não linearmente incompatível com alguma substituição θ_2 utilizada na condensação de outra cláusula dual de W_d.

Cabe esclarecer o que queremos dizer com *não linearmente incompatibilidade* nesta definição:

duas substituições θ_1 e θ_2 são não linearmente incompatíveis se são incompatíveis e as variáveis, cujos valores de substituição são incompatíveis, provêm de literais diferentes.

É claro que duas substituições somente são incompatíveis quando atribuem, para as mesmas variáveis, valores que não unificam. É claro também que, se duas substituições são incompatíveis e as variáveis em comum destas substituições provêm de literais diferentes, estas variáveis são não lineares. Daí o nome para esta incompatibilidade.

Vejamos alguns exemplos. Seja, $Conds(D)$ a condensação da cláusula dual D, temos:

Exemplo 14 $W_d = [\langle P(x), P(a)\rangle, \langle P(x), Q(a)\rangle, \langle Q(x), Q(a)\rangle, \langle Q(x), P(a)\rangle]$

- $Conds(\langle P(x), P(a)\rangle) = \langle P(x)\rangle$, *com* $\theta_1 = \{x/a\}$ *e*
- $Conds(\langle P(x), Q(a)\rangle) = \langle P(x), Q(a)\rangle$ *e*
- $Conds(\langle Q(x), Q(a)\rangle) = \langle Q(x)\rangle$, *com* $\theta_2 = \{x/a\}$ *e*
- $Conds(\langle Q(x), P(a)\rangle) = \langle Q(x), P(a)\rangle$

neste caso, θ_1 *e* θ_2 *atribuem valores para as mesma variável* x *proveniente de literais diferentes (*$P(x)$ *e* $Q(x)$*) o que leva à condição de não linearidade da variável* x*. Porém, os valores atribuídos a este variável são compatíveis e as cláusulas duais podem, então, ser assim condensadas.*

Exemplo 15 $W_d = [\langle P(x), P(a)\rangle, \langle P(x), Q(b)\rangle, \langle Q(x), Q(b)\rangle, \langle Q(x), P(a)\rangle]$

- $Conds(\langle P(x), P(a)\rangle) = \langle P(x)\rangle$, *com* $\theta_1 = \{x/a\}$ *e*
- $Conds(\langle P(x), Q(b)\rangle) = \langle P(x), Q(b)\rangle$ *e*
- $Conds(\langle Q(x), Q(b)\rangle) = \langle Q(x), Q(b)\rangle$ *e*
- $Conds(\langle Q(x), P(a)\rangle) = \langle Q(x), P(a)\rangle$

neste caso, a condensação da terceira cláusula dual com $\theta_2 = \{x/b\}$ *não é permitida porque a variável* x *de* θ_1 *e* θ_2 *vem de literais diferentes (*$P(x)$ *e* $Q(x)$*) e são incompatíveis. Portanto,* θ_1 *e* θ_2 *são não linearmente incompatíveis. Outra possibilidade é efetuar a condensação da terceira cláusula dual antes da primeira, o que leva à uma condensação diferente para a fórmula:*

- $Conds(\langle P(x), P(a)\rangle) = \langle P(x), P(a)\rangle$ *e*

- $Conds(\langle P(x), Q(b)\rangle) = \langle P(x), Q(b)\rangle$ *e*
- $Conds(\langle Q(x), Q(b)\rangle) = \langle Q(x)\rangle$ *e*
- $Conds(\langle Q(x), P(a)\rangle) = \langle Q(x), P(a)\rangle$

Exemplo 16 $W_d = [\langle P(x), P(a), P(b)\rangle, \langle Q(x), P(a), P(b)\rangle]$

- $Conds(\langle P(x), P(a), P(b)\rangle) = \langle P(x)\rangle$, *com* $\theta_1 = \{x/a\}$, $\theta_2 = \{x/b\}$ *e*
- $Conds(\langle Q(x), P(a), P(b)\rangle) = \langle Q(x), P(a), P(b)\rangle$

neste caso, mesmo sendo θ_1 *e* θ_2 *incompatíveis, a condensação é habilitada porque: (i) ambas substituições provêm de mesma cláusula dual e (ii) a variável* x *provém do mesmo literal* $P(x)$. *Na realidade estas duas condições ocorrem sempre juntas, uma vez que jamais dois literais com mesma variável (isto é, literais com variáveis lineares) fazem parte de mesma cláusula dual.*

Exemplo 17 $W_d = [\langle P(x), P(a)\rangle, \langle P(x), P(b)\rangle, \langle Q(x), P(a)\rangle, \langle Q(x), P(b)\rangle]$

- $Conds(\langle P(x), P(a)\rangle) = \langle P(x)\rangle$, *com* $\theta_1 = \{x/a\}$ *e*
- $Conds(\langle P(x), P(b)\rangle) = \langle P(x)\rangle$, *com* $\theta_2 = \{x/b\}$ *e*
- $Conds(\langle Q(x), P(a)\rangle) = \langle Q(x), P(a)\rangle$
- $Conds(\langle Q(x), P(b)\rangle) = \langle Q(x), P(b)\rangle$

também neste caso, mesmo sendo θ_1 *e* θ_2 *incompatíveis e vindo de condensações de diferentes cláusulas duais, elas não são não linearmente incompatíveis, uma vez que a variável* x *de ambas substituições provém do mesmo literal* $P(x)$. *Ambas as condensações podem, assim, ser habilitadas.*

Salientamos que assumimos que as variáveis de W_d encontram-se devidamente renomeadas, isto é, quaisquer literais com variáveis de mesmo nome são provenientes de mesma cláusula da forma conjuntiva W_c equivalente à W_d (isto é, quaisquer variáveis de mesmo nome são, de fato, as mesmas). Se este não for o caso, não podemos dizer que estas definições levam à obtenção de uma forma condensada da fórmula original. Não que a fórmula resultante seja incorreta mas, alguns conflitos de substituições podem ser identificados erroneamente e impedir algumas simplificações que seriam possíveis se variáveis diferentes tivessem nomes diferentes.

4.3.4 Quanto à simplificação da forma disjuntiva

Os resultados da primeira parte desta seção, referente às relações estruturais entre as formas normais de uma fórmula e às sutilezas envolvidas na simplificação da forma disjuntiva, encontram-se no artigo *Forma normal disjuntiva em lógica de primeira ordem* [43]. Neste artigo é proposto um algoritmo de simplificação da forma disjuntiva que considera aquelas relações estruturais. Este artigo, porém, foi escrito anteriormente à extensão das definições de *condensação* para a forma disjuntiva.

O algoritmo apresentado neste artigo não utiliza um hipercubo como estrutura adjacente, como é o caso do algoritmo referido para a simplificação da forma conjuntiva – *A Concurrent Algorithm for Logical Subsumption* [45]. Não foi estudada a adaptação deste algoritmo para realizar a condensação de uma fórmula disjuntiva. Sua estrutura subjacente – um hipercubo – é bastante adequada para a detecção de subsunções entre cláusulas (duais) de fórmulas lógicas. Esta estrutura é apresentada no final deste capítulo, quando abordada a transformação entre as formas canônicas.

Também nenhum estudo de complexidade, ou mesmo de correção, da condensação (interna e externa) da forma disjuntiva foi, até o momento, realizado. O máximo que podemos fazer até agora (por alto) é assumir para a condensação externa a mesma complexidade da condensação de cláusulas: NP-hard. Não esperamos nenhum resultado melhor para a condensação de uma cláusula dual. De qualquer forma, acreditamos que este estudo da relação entre as formas e a extensão da definição da condensação para a forma disjuntiva de uma fórmula possa contribuir para o desenvolvimento de algoritmos que consigam evitar testes de subsunção desnecessários.

O mais importante resultado desta seção é a constatação de que a forma condensada de uma fórmula na forma disjuntiva não é única. Não sendo única, podem existir formas condensadas maiores e menores para a mesma fórmula. Se pensarmos em eficiência de métodos de prova, fórmulas menores podem, a princípio, propiciar provas mais rápidas. Porém nenhuma resultado concreto temos nesta direção até o momento.

4.4 A transformação dual

Até agora falamos das formas normais canônicas conjuntiva e disjuntiva para a representação de uma fórmula e de sua simplificação. Falamos também que, dada uma fórmula W, suas formas conjuntiva W_c e disjuntiva W_d são equivalentes entre si e à fórmula original, isto é $W \leftrightarrow W_c \leftrightarrow W_d$. Nesta seção abordamos a transformação entre as formas conjuntiva e disjuntiva de uma fórmula, isto é, *a transformação dual.*

Vimos também que as formas normais canônicas apresentam uma relação estrutural entre elas, sendo uma obtida pela distributividade dos operadores da outra. Com isto, cada cláusula da forma normal conjuntiva é formada por um único representante de cada cláusula dual da forma normal disjuntiva, e vice-versa. Com isto, também, todas as cláusulas de uma forma normal encontram-se representadas simultaneamente em cada cláusula da outra forma. Cada uma das formas normais é um tipo de representação "holográfica" da outra: cada "parte" de uma contém uma representação do "todo" da outra.

A maneira mais direta de obter W_d dado W_c é construir todas as combinações possíveis de literais de W_c, de modo que em cada combinação apareça apenas um literal de cada cláusula de W_c. O problema com este método é que o número de cláusulas duais resultantes cresce exponencialmente com o número de cláusulas de W_c e, daquelas, muitas são subsumidas. O conjunto de cláusulas duais W_d deve então ser simplificado, ou seja, deve ser calculada a sua versão *condensada.*

O caso dual, ou seja, a transformação inversa, é simétrico. Seja qual for a forma canônica do conjunto original, aplicando-se a transformação dual em um conjunto resultante de uma transformação dual, obtém-se, após a devida simplificação, o conjunto original.

Exemplo 18 *A transformação do conjunto de cláusulas:*

$$W_c = \langle\, [A1, A2, A3],\, [B1, B2, B3]\, \rangle$$

resulta no conjunto de cláusulas duais:

$$\begin{aligned} W_d = [\ & \langle A1, B1\rangle,\ \langle A1, B2\rangle,\ \langle A1, B3\rangle,\ \langle A2, B1\rangle,\ \langle A2, B2\rangle \\ & \langle A2, B3\rangle,\ \langle A3, B1\rangle,\ \langle A3, B2\rangle,\ \langle A3, B3\rangle\,] \end{aligned}$$

cada cláusula dual contém todas as cláusulas representadas, por isto, o número de literais de cada cláusula dual é o mesmo que o número de cláusulas de W_c. Já, na transformação inversa, 512 cláusulas diferentes são geradas. Porém muitas destas não são condensadas ou são subsumidas. Por exemplo, dentre as 512 cláusulas geradas tem-se [A1, A1, A1, A2, A2, A2, A3, A3, A3], [A1, A1, A2, A2, A2, A3, A3, A3, B1] e [A1, A2, A3, B1, B1, B2, B2, B2, B3] com um literal oriundo de cada cláusula dual. Destas, a primeira pode ser simplificada para [A1, A2, A3] que subsume a segunda e terceira cláusulas. Ao calcularmos a versão condensada deste conjunto de 512 cláusulas, obtemos um conjunto igual a W_c com 2 cláusulas. Apesar das cláusulas de W_c não conterem nove literais cada, todas as nove cláusulas duais de W_d estão representadas em cada cláusula de W_c. Ao torná-las condensadas, um mesmo literal destas ou representa mais de uma cláusula dual de W_d ou subsume um literal de uma cláusula dual de W_d.

Esta noção de literais representantes de cláusulas é a base do algoritmo proposto. Dela segue-se a noção de saturação de uma cláusula, utilizada no processo de transformação. Uma cláusula é dita *saturada* se seus literais representam todas as cláusulas do conjunto original.

A transformação dual, independentemente de sua aplicação no método dual, representa um importante objeto de pesquisa (por exemplo, [28],[34]). Mais recentemente, um algoritmo mais eficiente foi proposto [36]. No entanto, este algoritmo gera algumas cláusulas subsumidas, o que torna obrigatória a verificação de subsunção para cada nova cláusula gerada.

4.4.1 Um algoritmo para a transformação dual

A grande dificuldade encontrada no problema da transformação dual é o crescimento exponencial enfrentado durante a transformação. Em [6] o autor sugere, como alternativa possível à solução de problemas cujo crescimento é exponencial, o uso de computadores paralelos. O uso de computadores paralelos implica que algoritmos devem ser redesenhados a fim de explorar eficientemente a capacidade de paralelismo da máquina. A adaptação de algoritmos seqüencias para este fim nem sempre sempre é uma tarefa fácil ou possível. Uma solução alternativa é projetar algoritmos específicos para arquiteturas paralelas.

O algoritmo proposto vem ao encontro da minimização destes dois problemas:

- ele gera, a partir de um conjunto de cláusulas (duais), diretamente a versão condensada de seu conjunto dual e,
- sua estrutura o torna naturalmente concorrente e, por isto, facilmente paralelizável.

Nele são levadas em conta as informações sobre o contexto de cada cláusula (dual) e de cada literal do conjunto original. Estas informações são armazenadas em uma estrutura de dados na forma de um *hipercubo*[14] onde dá-se a *propagação* das informações que formam as novas cláusulas (duais).

As explicações e exemplos são apresentados para a transformação de um conjunto de cláusulas para o conjunto de cláusulas duais equivalente. Isto foi adotado para evitar a constante repetição de notas indicando que o caso dual é simétrico. Esclarece-se, no entanto, que exceto o tratamento dos conflitos de subsunção que podem ocorrer

[14]O mesmo hipercubo utilizado no algoritmo apresentado no artigo *A Concurrent Algorithm for Logical Subsumption*[44, 45].

em fórmulas de primeira ordem, o algoritmo é simétrico para os dois casos de transformação. A formalização e mais detalhes sobre o algoritmo são encontradas no artigo *An algorithm for dual transformation in first-order logic* [10].

4.4.2 A dinâmica do algoritmo

Antes de falar da dinâmica do algoritmo, é necessário definir a estrutura de dados básica na qual se dá esta dinâmica. Esta estrutura de dados é definida na forma de um hipercubo de dimensão n onde n é igual ao número de predicados do conjunto original de cláusulas. Cada eixo do hipercubo representa um destes predicados; assim, cada vértice de dimensão n representa o cruzamento de n predicados. A cada vértice é dado uma *tarja* que é um vetor de tamanho n onde cada posição representa um eixo do hipercubo. Uma posição i deste vetor é igual a 1 se o eixo i compõe o vértice da tarja. Assim, a tarja de um vértice de dimensão n contém n posições iguais a 1. Vamos, então acompanhar, passo a passo, a transformação de um conjunto condensado de cláusulas para o equivalente conjunto condensado de cláusulas duais.

Exemplo 19 *O hipercubo associado à uma fómula cujas formais normais conjuntiva e disjuntiva são, respectivamente:*

$$\begin{aligned}
W_c &= \langle [P(a), Q(a)],\ [P(a), R(x_1)],\ [P(x_2), R(a)] \rangle \\
W_d &= [\langle P(a), P(a), P(x_2)\rangle, \langle P(a), P(a), R(a)\rangle, \langle P(a), R(x_1), P(x_2)\rangle \\
&\quad \langle P(a), R(x_1), R(a)\rangle, \langle Q(a), P(a), P(x_2)\rangle, \langle Q(a), P(a), R(a)\rangle \\
&\quad \langle Q(a), R(x_1), P(x_2)\rangle, \langle Q(a), R(x_1), R(a)\rangle]
\end{aligned}$$

e cuja forma disjuntiva condensada é:

$$W_{d'} = [\langle P(x_2)\rangle, \langle P(a), R(a)\rangle, \langle Q(a), R(x_1)\rangle]$$

é mostrado na figura 4.1. Neste exemplo o conjunto de cláusulas a ser transformado, W_c, é composto por 3 predicados, sendo pois associado, a um hipercubo de 3 dimensões. O eixo 1 representa o predicado P e o vértice de primeira dimensão composto por este eixo possui a tarja 100. O mesmo acontece com os demais vértices de primeira dimensão cujas tarjas são 010, no eixo 2 com o predicado Q e 001, no eixo 3 com o predicado R. Já os vértices de segunda dimensão - 101, 011 e 110 - representam o cruzamento de dois eixos ou seja, de dois predicados. Por conseguinte, o vértice final, de terceira dimensão, cuja tarja é 111, representa o cruzamento dos três eixos do hipercubo (todos os predicados).

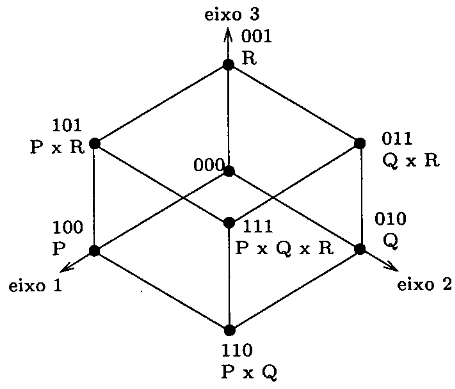

Figura 4.1: Predicados dos exemplos 19 no hipercubo

Continuando a definição de nosso hipercubo, associamos cada literal do conjunto a ser transformado ao vértice (de primeira dimensão) do eixo de mesmo predicado. Estendendo este conceito, podemos dizer que o vértice que representa o cruzamento de n eixos, representa literais de n predicados diferentes (ou ainda, como veremos, cláusulas duais com literais de n predicados diferentes).

Para efetuar a transformação dual o algoritmo utiliza esta estrutura da seguinte maneira. Na primeira dimensão, o algoritmo combina os literais de um mesmo vértice gerando cláusulas duais possivelmente incompletas, verifica se alguma cláusula dual completa foi gerada. Uma cláusula dual D é completa[15] se, para cada cláusula C de W_c, D contém um, e apenas um, literal de C ou um literal de D subsume um literal de C. As cláusulas duais completas são deixadas no vértice e as incompletas são propagadas para os vértices sucessores. A seguir, para cada dimensão a partir da segunda, em cada vértice desta dimensão, as cláusulas duais incompletas recebidas de vértices de menor dimensão são combinadas gerando novas cláusulas duais. Destas, as completas permanecem no vértice e as incompletas são propagados para os vértices sucessores. Este ciclo de combinação, verificação e propagação repete-se até o vértice de última dimensão do hipercubo ser alcançado, ou até que nenhuma cláusula dual incompleta seja propagada.

Vejamos como se dá a transformação do exemplo 19. Primeiramente associamos os literais de W_c a seus vértices de primeira dimensão, conforme mostrado na figura 4.2. As setas no interior do hipercubo representa os caminhos da propagação.

[15]Na formalização do algoritmo no artigo citado anteriormente, uma cláusula completa é dita "saturada".

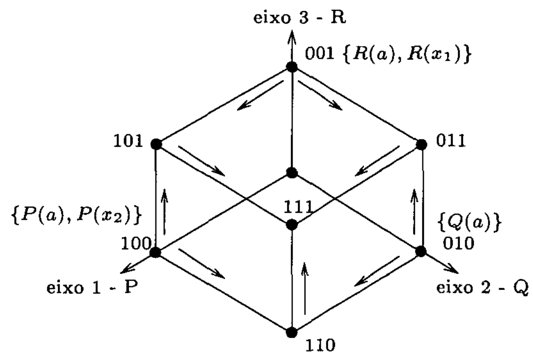

Figura 4.2: Literais das cláusulas do exemplo 19 em seus vértices

Em cada um destes vértices (100, 010 e 001), os literais são combinados entre eles de forma que somente sejam combinados literais que pertençam a cláusulas diferentes de W_c. Desta combinação obtemos cláusulas duais (possivelmente incompletas) com literais de mesmo predicado. Isto feito, verifica-se se alguma destas cláusulas duais é completa podendo, então, fazer parte do W_d final. Em nosso exemplo, este é o caso da cláusula dual $\langle P(x_2)\rangle$ do vértice 100. Esta cláusula é completa por conter (ou subsumir) literais das três cláusulas de W_c: o literal $P(x_2)$, da terceira que, or sua vez, subsume o literal $P(a)$ da primeira e segunda. Assim, esta cláusula dual permanece em seu vértice do qual apenas é propadada, para os vértices 110 e 101, a cláusula dual incompleta $\langle P(a)\rangle$. Nos demais vértices de primeira dimensão, nenhuma cláusula dual completa é gerada dando-se apenas a propagação de cláslulas duais incompletas para os respectivos vértices sucessores.

A cada vértice de segunda dimensão, as cláusulas duais incompletas recebidas de vértices predecessores são combinadas resultando em novas cláusulas duais, desta vez, de literais de dois predicados diferentes. Esta combinação, não se deve esquecer, é efetuada apenas entre cláusulas duais incompletas que contenham literais oriundos de diferentes cláusulas de W_c. Novamente é verificado se alguma cláusula dual completa foi gerada. Em nosso exemplo, no vértice 111 nenhuma cláusula dual incompleta pode ser combinada. No vértice 101 é gerada a cláusula dual completa $\langle P(a), R(a)\rangle$ e nada é propagado. Já no vértice 011, é gerada a cláusula dual completa $\langle Q(a), R(x_1)\rangle$ e propagada a cláusula dual incompleta $\langle Q(a), R(a)\rangle$.

No vértice 111, de terceira dimensão, por receber apenas uma cláusula dual incompleta, o processo é encerrado sem a geração de nenhuma cláusula dual. Tem-se então, como resultado deste processo, um hipercubo com cláusulas duais completas guardadas em alguns de seus vértices, como pode ser visto na figura 4.3. Resta somente

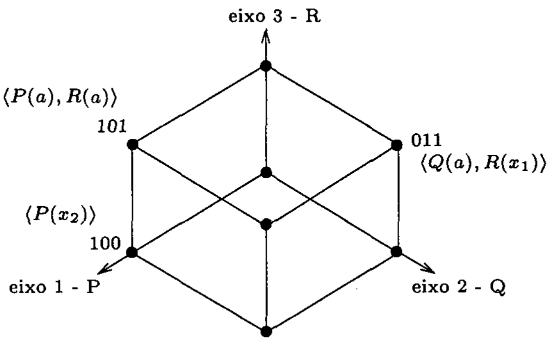

Figura 4.3: Cláusulas duais do exemplo 19 em seus vértices associados

Dim	Vert	Recebidas	Completas	Propagadas
1	100	$\langle P(a)\rangle, \langle P(x_2)\rangle,$	$\langle P(x_2)\rangle$	$\langle P(a)\rangle$
	010	$\langle Q(a)\rangle$	-	$\langle Q(a)\rangle$
	001	$\langle R(a)\rangle, \langle R(x_1)\rangle$	-	$\langle R(a)\rangle, \langle R(x_1)\rangle$
2	110	$\langle P(a)\rangle$, $\langle Q(a)\rangle$	-	-
	101	$\langle P(a)\rangle$, $\langle R(a)\rangle, \langle R(x_1)\rangle$	$\langle P(a), R(a)\rangle$	-
	011	$\langle Q(a)\rangle$, $\langle R(a)\rangle, \langle R(x_1)\rangle$	$\langle Q(a), R(x_1)\rangle$	$\langle Q(a), R(a)\rangle$
3	111	$\langle Q(a), R(a)\rangle$	-	-

Tabela 4.2: Cláusulas duais no hipercubo na transformação do exemplo 19

recuperá-las e compor o o conjunto de cláusulas duais condensado $W_{d'}$, resultante de nossa transformação. As cláusulas duais recebidas, completas e propagadas a cada vértice do hipercubo são mostradas na tabela 4.2.

Pela estrutura de dados representada pelo hipercubo, vê-se que este processo é facilmente paralelizável. Porém, esta não é a única justificativa para o uso do hipercubo. O algoritmo possui uma outra característica importante: gera diretamente a versão condensada do conjunto transformado, isto quer dizer que não são geradas cláusulas duais subsumidas entre si. A relação de subsunção entre cláusulas define uma ordem parcial para um conjunto de cláusulas. Esta ordem parcial está diretamente relacionada aos símbolos de predicado que compõem cada cláusula do conjunto. Ao ser realizada através de um hipercubo, a propagação das informações dá-se segundo uma ordem de precedência das mesmas. Como pode ser observado na figura 4.4, a ordem parcial estabelecida para as cláusulas duais do exemplo 19 é representada pelos caminhos da propagação do hipercubo.

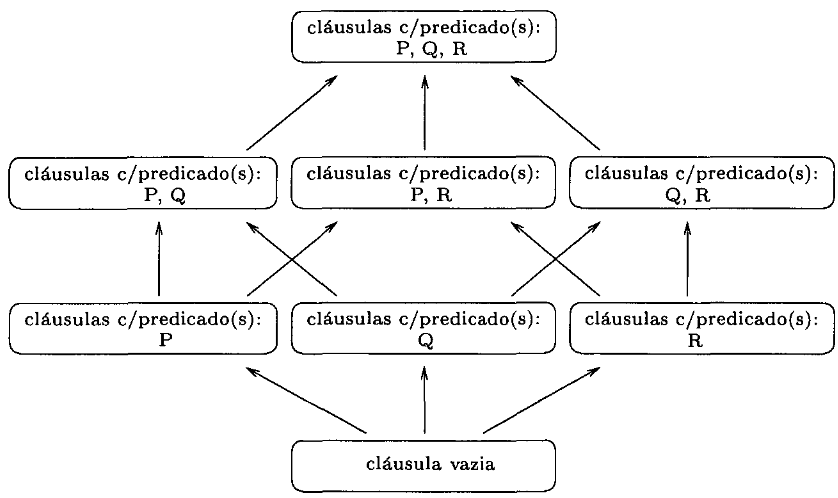

Figura 4.4: Ordem parcial para cláusulas duais do exemplo 19

Com isto tem-se garantido que se uma cláusula dual D_1 subsume uma cláusula D_2, D_2 pertence a mesmo vértice de D_1 ou sucessor deste[16]. Também, pela propagação dar-se da menor dimensão para a maior, tem-se garantido que D_1 é gerada antes de D_2. Com isto, reconhecendo-se D_1, a geração de D_2 pode ser evitada.

A subsunção entre cláusulas duais é evitada pelo uso da estrutura de dados na forma de um hipercubo. Já a subsunção interna às cláusulas duais é evitada no instante da combinação entre os literais de mesmo predicado que ocorre nos vértices de primeira dimensão.

A definição formal do algoritmo propriamente dito encontra-se em [10] e sua dinâmica e implementação em [42, 46] e não cabe ao escopo deste trabalho apresentá-los novamente. No entanto, a maneira como uma fórmula é representada na formalização do algoritmo é completamente pertinente apresentar. Chamamos esta representação de *representação cruzada* e dela muito faz uso o algoritmo na detecção de subsunções e verificaçõa de cláusulas completas durante a transformação. Esta representação é largamente utilizada no decorrer deste trabalho e, devido à sua importância, preferiu-se apresentá-la isoladamente do algoritmo. A representação cruzada e as relações entre os literais de uma fórmula, explicitadas por esta representação, são tratadas no próximo capítulo.

[16]No caso proposicional, D_2 pertence garantidamente a vértice sucessor ao vértice de D_1.

4.5 Conclusão

Neste capítulo apresentamos as formas normais canônicas conjuntiva e disjuntiva para representação de uma fórmula lógica. Estas não são as únicas, mas são as pertinentes ao escopo deste trabalho (e são, também as mais largamente utilizadas).

Primeiramente apresentamos as formas propriamente ditas, depois procedimentos para sua simplificação e, por fim, a transformação entre elas. Quando do início do estudo do método dual não pensávamos em estudar tão profundamente estas formas. Pensávamos que nosso enfoque se resumiria na relação existente entre as mesmas. No entanto, as sutilezas apresentadas na simplificação da forma disjuntiva, a detecção da existência de mais de uma forma condensada, a falta de literatura referente à esta forma nos fez voltar nossa atenção a este assunto e estudá-lo mais profundamente.

Na realidade este assunto, a princípio secundário, tornou-se o mais amplamente divulgado em nosso estudos. Uma publicação internacional no *Journal of Automated Reasoning* [10], o principal na área de dedução automática, outra internacional no livro *Frontiers in Artificial Intelligence and its Applications* [45], uma no *Semish - Seminário de Software e Hardware* [42] e também nas duas primeiras edições do *Laptec - Congresso de Lógica Aplicado à Tecnologia* [43, 44]. Podemos considerar um bom resultado para este assunto, mas muito mais ainda tem para ser pesquisado.

No entanto nosso principal assunto é o método dual, o qual é totalmente baseado nas definições das formas normais de representação de uma fórmula lógica. Achamos que o material aqui apresentado é suficiente para o entendimento das características deste método, assunto do próximo capítulo.

Alice laughed: "There's no use trying," she said: "one can't believe impossible things." "I daresay you haven't had much practice," said the Queen. "When I was younger, I always did it for half an hour a day. Why, sometimes I've believed as many as six impossible things before breakfast."

Lewis Carroll, Alice's Adventures in Wonderland, 1865

Capítulo 5

Eliminando o impossível

Em 1998, no artigo *Concurrent Inference through Dual Transformation* [6], Guilherme Bittencourt define o Método Dual, um método de prova correto e completo para refutação para lógica de primeira ordem. Este método, embora aparentemente bastante ineficiente, apresenta interessantes características, tal como a não existência de regras de inferência explícitas: a inferência dá-se apenas pela exploração da dualidade das representações, conjuntiva e disjuntiva, e suas semânticas.

Nosso interesse é apresentar o método apenas o suficiente para o entendimento do problema do desenvolvimento de estratégias para o mesmo.

5.1 Elementar, meu caro Watson

Antes de apresentar o método dual formalmente, vamos entender qual é o raciocínio a ele subjacente. Para tal, considere o resumo de uma passagem encontrada no primeiro capítulo do livro "The Sign of Four", de Conan Doyle, onde Sherlock Holmes exemplifica a Dr. Watson a diferença entre observação e dedução:

> "Por exemplo, a observação me mostra que você esteve no correio da rua Wigmore nesta manhã, mas a dedução me faz saber que você enviou um telegrama. A lama vermelha em seu sapato somente é encontrada, na vizinhança, em frente a este correio, onde a calçada foi removida. Isto é observação, o resto é dedução. Eu sei que você não escreveu nenhuma carta e vejo vários selos em sua escrivaninha. O que você foi fazer no correio, senão enviar um telegrama?"

Consideremos que Holmes tem razões para acreditar que Watson foi a algum correio, que foi na vizinhança e que o motivo para ir-se ao correio seja, unicamente, enviar uma carta ou um telegrama ou comprar selos, e que Watson não tinha nenhuma carta para enviar e que não compraria selos sobressalentes. Com isto pode-se representar a dedução acima na seguinte fórmula:

$$\{(Lama\_Sapato \rightarrow Foi\_Correio\_Wigmore),\ (Foi\_Correio\_Wigmore \rightarrow Enviou\_Carta \vee Enviou\_Telegrama \vee Comprou\_Selos),\ Lama\_Sapato,\ \neg Enviou\_Carta,\ \neg Comprou\_Selos\ \} \vdash Enviou\_Telegrama$$

que pode ser reescrita:

$$\{(Lama \rightarrow Correio),\ (Correio \rightarrow Carta \vee Telegrama \vee Selo),\ Lama,\ \neg Carta,\ \neg Selo\ \} \vdash Telegrama$$

a qual é uma tautologia, demonstrando que a dedução de Holmes está correta. Este fato pode facilmente ser comprovado com qualquer método de prova apresentado anteriormente. Vamos porém analisar esta dedução mais a fundo.

5.1.1 O Caso da Rua Wigmore - proposicional

hipótese 1 - prova afirmativa

A forma disjuntiva de uma fórmula pode ser vista como a enumeração de todas as possibilidades de "verdade" da fórmula, isto é, de todos os "fatos" (literais) que, quando simultaneamente verdadeiros, satisfazem as premissas da forma conjuntiva desta mesma fórmula. Então, conhecendo-se a forma disjuntiva das premissas pode-se, aplicar o famoso lema de Sherlock Holmes, contido no mesmo livro:

> "Quando se eliminar o impossível, o que resta, mesmo que improvável, tem de ser a verdade."[1]

e eliminar "o impossível" contido na fórmula. Por exemplo, considere a forma conjuntiva da fórmula:

$$W_c = \langle\ [\neg Lama, Correio], [\neg Correio, Carta, Telegrama, Selo], [Lama], [\neg Carta], [\neg Selo]\ \rangle$$

[1] "When you have eliminated the impossible, whatever remains, however improbable, must be the truth.", Sir Arthur Conan Doyle, The Sign of Four, 1890.

e sua forma disjuntiva:

$$
\begin{aligned}
W_d = [\; & \langle \neg Lama, \neg Correio, Lama, \neg Carta, \neg Selo \rangle, \\
& \langle \neg Lama, Carta, Lama, \neg Carta, \neg Selo \rangle, \\
& \langle \neg Lama, Telegrama, Lama, \neg Carta, \neg Selo \rangle, \\
& \langle \neg Lama, Selo, Lama, \neg Carta, \neg Selo \rangle, \\
& \langle Correio, \neg Correio, Lama, \neg Carta, \neg Selo \rangle, \\
& \langle Correio, Carta, Lama, \neg Carta, \neg Selo \rangle, \\
& \langle Correio, Telegrama, Lama, \neg Carta, \neg Selo \rangle, \\
& \langle Correio, Selo, Lama, \neg Carta, \neg Selo \rangle \;]
\end{aligned}
$$

Para eliminar "o impossível" contido nesta fórmula, considera-se o que segue. Por exemplo, pela primeira cláusula dual, uma possibilidade das premissas serem verdadeiras é quando "$\neg Lama$ e $\neg Correio$ e $Lama$ e $\neg Carta$ e $\neg Selo$" são simultaneamente verdadeiros. Sabe-se, no entanto, que em nenhuma interpretação $\neg Lama$ e $Lama$ podem ser simultaneamente verdadeiros. Portanto esta possibilidade de verdade não pode ser considerada por ser contraditória, esta conjunção de fatos é "impossível". Além desta, outras cláusulas duais expressam possibilidades contraditórias. Como já visto, cláusulas duais contraditórias podem ser removidas de um conjunto sem alterar o valor verdade do mesmo. Removendo de W_d todas as "conjunções impossíveis" temos a fórmula:

$$W_{d'} = [\; \langle Correio, Telegrama, Lama, \neg Carta, \neg Selo \rangle \;]$$

na qual, pela única cláusula dual, verifica-se apenas uma "possibilidade de verdade", uma única conjunção de fatos que devem ser simultaneamente verdadeiros para as premissas serem verdadeiras. Estes fatos são explicitados com a transformação desta última fórmula para sua forma conjuntiva:

$$W_{c'} = \langle\; [Correio], [Telegrama], [Lama], [\neg Carta], [\neg Selo] \;\rangle$$

ou seja, Watson foi ao correio na rua Wigmore e enviou um telegrama e, fatos já conhecidos, Watson tinha lama no sapato, não enviou uma carta e não comprou selos. Os dois primeiros fatos $Correio$ e $Telegrama$ são conseqüências lógicas das premissas, isto é são teoremas da fórmula expressa em W_c. Considerando-se o teorema $Telegrama$, expresso na fórmula original, observa-se que o mesmo foi gerado das premissas. Portanto esta dedução efetuada é prova para este teorema.

O mecanismo aqui utilizado para dedução não utilizou explicitamente nenhuma regra de inferência. A inferência foi efetuada apenas pela exploração das diferentes representações de uma fórmula e suas semânticas: eliminação de contradições explicitadas na forma disjuntiva e posterior transformação para a forma conjuntiva, na qual os teoremas são, de certa forma, explicitados.

Outras características podem ser observadas:

1. $W_{c'}$ contém premissas que já existiam em W_c. Este mecanismo de inferência gera, não apenas teoremas "isolados" como é comum nos demais métodos de prova, mas uma subfórmula $W_{c'}$ decorrente da fórmula original W_c, tal que $W_c \leftrightarrow W_{c'}$;

2. ao retirar-se de um conjunto todas as cláusulas duais nas quais um literal ocorre, retira-se da fórmula, na realidade, este literal propriamente dito. Semanticamente, este fato indica que este literal é irrelevante para a satisfação da fórmula e pode ser desconsiderado. Por irrelevante quer-se dizer que o literal não pode ser verdadeiro e, por isto, a valor verdade da fórmula depende dos demais literais nela contidos;

3. a conclusão final "Telegrama" bem como a intermediária "Correio" foram obtidas em apenas um ciclo de inferência. O silogismo hipotético $Lama \rightarrow Correio \rightarrow (Carta \vee Telegrama \vee Selo)$ foi resolvido juntamente com as demais inferências. De fato, com este mecanismo, silogismos hipotéticos, mesmo quando em cadeia, são resolvidos simultaneamente.

hipótese 2 - prova por refutação

Vamos considerar a prova do teorema *Telegrama* por refutação. Sabe-se que primeiramente deve-se incluir sua negação ao conjunto de premissas. Com isto tem-se como forma conjuntiva e disjuntiva deste novo conjunto de premissas:

$$\begin{aligned} W_c = \langle\ & [\neg Lama, Correio], [\neg Correio, Carta, Telegrama, Selo], [Lama], \\ & [\neg Carta], [\neg Selo], [\neg Telegrama]\ \rangle \end{aligned}$$

$$
\begin{aligned}
W_d = [\ & \langle \neg Lama, \neg Correio, Lama, \neg Carta, \neg Selo, \neg Telegrama \rangle, \\
& \langle \neg Lama, Carta, Lama, \neg Carta, \neg Selo, \neg Telegrama \rangle, \\
& \langle \neg Lama, Telegrama, Lama, \neg Carta, \neg Selo, \neg Telegrama \rangle, \\
& \langle \neg Lama, Selo, Lama, \neg Carta, \neg Selo, \neg Telegrama \rangle, \\
& \langle Correio, \neg Correio, Lama, \neg Carta, \neg Selo, \neg Telegrama \rangle, \\
& \langle Correio, Carta, Lama, \neg Carta, \neg Selo, \neg Telegrama \rangle, \\
& \langle Correio, Telegrama, Lama, \neg Carta, \neg Selo, \neg Telegrama \rangle, \\
& \langle Correio, Selo, Lama, \neg Carta, \neg Selo, \neg Telegrama \rangle \ ]
\end{aligned}
$$

Como anteriormente cada cláusula dual representa uma possibilidade de verdade. Ao verificar a forma disjuntiva W_d, no entanto, descobre-se que todas as cláusulas duais são contraditórias, isto é, não existe nenhuma possibilidade de verdade para esta fórmula. Como era de se esperar, pois uma vez que sabemos que o teorema é uma conseqüência lógica do conjunto de premissas, a introdução de sua negação neste conjunto tem de torná-lo contraditório.

A detecção de contradição em todas as cláusulas duais comprova a insatisfazibilidade da fórmula: somente uma fórmula disjuntiva contraditória tem todas suas cláusulas duais contraditórias. Então, esta detecção é condição suficiente para a prova por refutação de um teorema.

5.1.2 O Caso da Rua Wigmore - primeira ordem

Para considerar a lógica de primeira ordem, precisamos transformar as premissas do exemplo anterior em uma fórmula lógica de primeira ordem[2]:

$$
\begin{aligned}
\{\ \forall x,y\ (\ & (Lama\_Sapato(x) \rightarrow Foi\_Correio(x, wigmore)),\ (Foi\_Correio(x,y) \rightarrow \\
& Enviou\_Carta(x,y) \vee Enviou\_Telegrama(x,y) \vee Comprou\_Selos(x,y)), \\
& Lama\_Sapato(watson),\ \neg Enviou\_Carta(watson, x), \\
& \neg Comprou\_Selos(watson, x)\ )\ \} \vdash Enviou\_Telegrama(watson, wigmore)
\end{aligned}
$$

[2]Os predicados dá fórmula expressam: $Lama\_Sapato(x)$ "x tem lama no sapato", $Foi\_Correio(x,y)$ "x foi ao correio y", $Enviou\_Carta(x,y)$ "x enviou uma carta no correio y", $Enviou\_Telegrama(x,y)$ "x enviou um telegrama no correio y" e $Comprou\_Selos(x,y)$ "x comprou selos no correio y".

cujos nomes de predicados e constantes, devido ao tamanho da fórmula, podem ser reescritas de forma reduzida, resultando na fórmula:

$$\{\forall x, y\ (\ (Lama(x) \to Correio(x, wi)),\ (Correio(x, y) \to Carta(x, y)\ \vee\ Teleg(x, y)$$
$$\vee\ Selo(x, y)), Lama(wa),\ \neg Carta(wa, x),\ \neg Selo(wa, x)\ )\ \}$$
$$\vdash Teleg(wa, wi)$$

hipótese 1 - prova afirmativa

Como anteriormente, considere as formas normais conjuntiva e disjuntiva das premissas:

$$\begin{aligned} W_c\ = \langle\ & [\neg Lama(v), Correio(v, wi)], \\ & [\neg Correio(w, x), Carta(w, x), Teleg(w, x), Selo(w, x)], \\ & [Lama(wa)], [\neg Carta(wa, y)], [\neg Selo(wa, z)]\ \ \rangle \end{aligned}$$

$$\begin{aligned} W_d\ = [\ \ & \langle \neg Lama(v), \neg Correio(w, x), Lama(wa), \neg Carta(wa, y), \neg Selo(wa, z)\rangle, \\ & \langle \neg Lama(v), Carta(w, x), Lama(wa), \neg Carta(wa, y), \neg Selo(wa, z)\rangle, \\ & \langle \neg Lama(v), Teleg(w, x), Lama(wa), \neg Carta(wa, y), \neg Selo(wa, z)\rangle, \\ & \langle \neg Lama(v), Selo(w, x), Lama(wa), \neg Carta(wa, y), \neg Selo(wa, z)\rangle, \\ & \langle Correio(v, wi), \neg Correio(w, x), Lama(wa), \neg Carta(wa, y), \neg Selo(wa, z)\rangle, \\ & \langle Correio(v, wi), Carta(w, x), Lama(wa), \neg Carta(wa, y), \neg Selo(wa, z)\rangle, \\ & \langle Correio(v, wi), Teleg(w, x), Lama(wa), \neg Carta(wa, y), \neg Selo(wa, z)\rangle, \\ & \langle Correio(v, wi), Selo(w, x), Lama(wa), \neg Carta(wa, y), \neg Selo(wa, z)\rangle\ \ ] \end{aligned}$$

Por conter variáveis, algumas cláusulas duais são apenas "potencialmente" contraditórias. Para torná-las contradições explícitas deve-se, nelas, aplicar uma substituição. Por exemplo, a substituição $\{(v/wa)\}$ torna a primeira cláusula dual explicitamente contraditória. É certo que para este valor da variável v nenhuma interpretação pode satisfazer esta cláusula dual que passa a ser:

$$\langle \neg Lama(wa), \neg Correio(w, x), Lama(wa), \neg Carta(wa, y), \neg Selo(wa, z)\rangle$$

Esta pode então ser retirada de W_d, contanto que todas as ocorrências da variável v, nas demais cláusulas duais, sejam substituídas pelo valor que tornou aquela cláusula dual contraditória. Isto é, contanto que a substituição $\{(v/wa)\}$ seja aplicada no restante da fórmula, pois foi com este valor da variável v que a contradição tornou-se explícita. Com isto, especializa-se a instância de W_c que pode ser satisfazível. Da mesma forma:

- as substituições $\{(v/wa)\}$ e $\{(w/wa),(x/y)\}$ tornam a segunda cláusula dual contraditória,
- a substituição $\{(v/wa)\}$ torna a terceira cláusula dual contraditória,
- as substituições $\{(v/wa)\}$ e $\{(w/wa),(x/z)\}$ tornam a quarta cláusula dual contraditória,
- a substituição $\{(v/w),(x/wi)\}$ torna a quinta cláusula dual contraditória,
- a substituição $\{(w/wa),(x/y)\}$ torna a sexta cláusula dual contraditória,
- a substituição $\{(w/wa),(x/z)\}$ torna a oitava cláusula dual contraditória.

Segundo o lema de Sherlock Holmes, eliminando-se o impossível encontra-se a verdade. Em lógica de primeira ordem, por envolver a noção de instâncias de uma fórmula, a noção de "impossível" é um pouco mais sutil. "Impossível" implica "impossível em todas interpretações". Como uma interpretação somente é atribuída a instâncias fechadas de uma fórmula, substituições compatíveis são necessárias, na realidade, para o cálculo da mesma instância fechada da fórmula[3]. Por isto, por "eliminar o impossível", deve-se entender "eliminar o impossível na mesma instância".

No exemplo todas as substituições que tornam as cláusulas duais contraditórias são compatíveis e podem ser combinadas na substituição $\{(v,w/wa),(x,y,z/wi)\}$. Portanto, aplicando esta substituição em W_d obtemos uma instância de W_d na qual todas as cláusulas duais, exceto a sétima, são (explicitamente) contraditórias e podem ser removidas, resultando a fórmula $W_{d'}$ contendo uma única cláusula dual:

$$\langle Correio(wa,wi), Teleg(wa,wi), Lama(wa), \neg Carta(wa,wi), \neg Selo(wa,wi)\rangle$$

Como anteriormente, ao transformarmos $W_{d'}$ para a forma conjuntiva

$$\begin{aligned} W_{c'} = \langle\ & [Correio(wa,wi)], [Telegr(wa,wi)], [Lama(wa)], \\ & [\neg Carta(wa,wi)], [\neg Selo(wa,wi)]\ \rangle \end{aligned}$$

[3]Por exemplo, dado a fórmula $(P(x) \vee Q(y))$ e as substituições $\theta_1 = \{(x/a)\}$, $\theta_2 = \{(y/a)\}$ e $\theta_3 = \{(y/b)\}$, podemos calcular as instâncias fechadas da fórmula: $(P(a) \vee Q(a))$ e $(P(a) \vee Q(b))$; em ambas as substituições compatíveis (θ_1,θ_2) e (θ_1,θ_3) foram necessárias ao cálculo da mesma instância fechada da fórmula.

temos explicitados os teoremas gerados. Vê-se que o teorema desejado

$$Telegr(wa, wi)]$$

foi gerado constituindo sua prova.

O mecanismo utilizado foi o mesmo e as características, em geral, são as mesmas observadas no caso proposicional, com algumas diferenças:

1. $W_{c'}$ contém premissas e *instâncias* de premissas que já existiam em W_c. Por exemplo $[\neg Carta(wa, wi)]$ de $W_{c'}$ é uma instância da premissa $[\neg Carta(wa, wi)]$ de W_c. Esta instância também é um teorema gerado: "Watson não enviou uma carta no correio da rua Wigmore". Este mecanismo de inferência, como no caso proposicional, gera não apenas teoremas "isolados" como é comum nos demais métodos de prova, mas uma subfórmula $W_{c'}$ decorrente da fórmula original W_c. A diferença é que no caso da lógica de primeira ordem $W_{c'}$ é uma conseqüência lógica de W_c, isto é, $W_c \to W_{c'}$;

2. como veremos posteriormente, na definição formal do método dual, pode ser necessário mais de um ciclo de inferência para a prova de um teorema.

hipótese 2 - prova por refutação

Como anteriormente, para considerar a prova do teorema por refutação devemos incluir sua negação nas premissas e calcular a forma conjuntiva

$$\begin{aligned} W_c = \langle\ & [\neg Lam(v), Corr(v, wi)], \\ & [\neg Corr(w, x), Cart(w, x), Tel(w, x), Sel(w, x)], \\ & [Lam(wa)], [\neg Cart(wa, y)], [\neg Sel(wa, z)], [\neg Tel(wa, wi)] \ \rangle \end{aligned}$$

e disjuntiva

$$\begin{aligned} W_d = [& \\ & \langle \neg Lam(v), \neg Corr(w, x), Lam(wa), \neg Cart(wa, y), \neg Sel(wa, z), \neg Tel(wa, wi)\rangle, \\ & \langle \neg Lam(v), Cart(w, x), Lam(wa), \neg Cart(wa, y), \neg Sel(wa, z), \neg Tel(wa, wi)\rangle, \\ & \langle \neg Lam(v), Tel(w, x), Lam(wa), \neg Cart(wa, y), \neg Sel(wa, z), \neg Tel(wa, wi)\rangle, \\ & \langle \neg Lam(v), Sel(w, x), Lam(wa), \neg Cart(wa, y), \neg Sel(wa, z), \neg Tel(wa, wi)\rangle, \\ & \langle Corr(v, wi), \neg Corr(w, x), Lam(wa), \neg Cart(wa, y), \neg Sel(wa, z), \neg Tel(wa, wi)\rangle, \\ & \langle Corr(v, wi), Cart(w, x), Lam(wa), \neg Cart(wa, y), \neg Sel(wa, z), \neg Tel(wa, wi)\rangle, \\ & \langle Corr(v, wi), Tel(w, x), Lam(wa), \neg Cart(wa, y), \neg Sel(wa, z), \neg Tel(wa, wi)\rangle, \\ & \langle Corr(v, wi), Sel(w, x), Lam(wa), \neg Cart(wa, y), \neg Sel(wa, z), \neg Tel(wa, wi)\rangle \ ] \end{aligned}$$

da fórmula resultante. Como era de se esperar, todas as cláusulas duais são potencialmente contraditórias. Procedendo-se a mesma busca das substituições que tornam cada cláusula dual contraditória, verifica-se que todas são compatíveis e podem ser combinadas resultando na substituição encontrada anteriormente $\{(v, w/wa), (x, y, z/wi)\}$. Aplicando-se esta substituição em W_d obtém-se uma instância de W_d na qual todas as cláusulas duais são (explicitamente) contraditórias. Como anteriormente, este fato é suficiente para a prova por refutação do teorema, uma vez que uma instância contraditória somente pode ser conseqüência lógica de uma fórmula contraditória: $W_c \vdash F$ se e somente se W_c é contraditório.

5.1.3 O Caso da Rua Wigmore - conclusão

De quatro formas diferentes foi comprovado que, naquela manhã, Dr. Watson foi ao correio na rua Wigmore enviar um telegrama, confirmando que Sherlock Holmes estava correto. (É claro!) No entanto, o resultado principal desta seção é a apresentação do mecanismo de inferência utilizado para esta confirmação.

Este mecanismo é a base do método dual. O exemplo apresentado envolve uma prova bastante simples, que pode ser obtida em um único ciclo. Normalmente, para fórmulas mais complexas, este não é o caso e a cada ciclo, a retenção dos teoremas intermediários pode tornar a obtenção da prova impraticável, problema comum dos métodos de prova para lógica de primeira ordem. Outro problema, que parece desconsiderável mas cujo custo computacional pode ser bastante elevado, é a combinação de substituições compatíveis. Ou seja, como nos demais métodos, o método dual apresenta boas e más características a serem apresentadas na seqüência.

5.2 O Método Dual propriamente dito

O método dual, como visto, baseia-se em sucessivas transformações de uma fórmula lógica de uma forma normal para a outra, a transformação dual, e conseqüentes retiradas de contradições explicitadas na forma disjuntiva. Para obter melhor eficiência, operando em fórmulas menores, a forma condensada de cada forma é calculada a cada transformação.

5.2.1 A definição

Dado um conjunto de cláusulas W_c, considere o conjunto de cláusulas duais W_d definido por:

$$W_d = \lfloor Dual(\lfloor W_c \rfloor) \rfloor$$

onde a função *Dual* transforma um conjunto de uma forma normal para a outra e $\lfloor X \rfloor$ indica a forma condensada da fórmula X. A transformação dual, bem como a condensação das formas normais de uma fórmula, foram tratados em capítulo anterior. Cabe lembrar, no entanto, que a condensação das formas inclui a retirada de tautologias e contradições das formas conjuntiva e disjuntiva, respectivamente.

Seja $W_d = [D_1, D_2, ..., D_n]$, onde cada cláusula dual D_i pode ser uma contradição. Seja θ_{ij} uma substituição tal que sua aplicação é a *j-ésima* maneira de tornar a *i-ésima* cláusula dual explicitamente contraditória. Seja Θ o conjunto de todas as substituições θ_{ij} e *Contra* uma função tal que:

$$\Theta = Contra(W_d)$$

Como já visto, o objetivo é a eliminação do máximo número de cláusulas duais contraditórias, evitando o uso simultâneo de duas substituições incompatíveis, isto é, que substituam a mesma variável por termos não unificáveis. Para tal, seja S_i um subconjunto de $\theta_{ij} \in \Theta$ cujas substituições são compatíveis. Seja ω_i o par $\langle \theta, N \rangle$, onde θ é a substituição gerada pela combinação das substituições de todos $\theta_{ij} \in S$, e N o conjunto resultante da união dos i de todos $\theta_{ij} \in S$. Isto é, cada ω_i é um par formado por uma substituição e o conjunto de cláusulas duais que ela elimina[4].

Seja $\Omega = \{\omega_1, \omega_2, ..., \omega_m\}$ o conjunto de todos ω_i de Θ. Para construir o conjunto Ω é necessário, a princípio, testar a compatibilidade de cada substituição associada com os membros do conjunto $\mathcal{P}(\Theta)$, onde $\mathcal{P}$ representa o conjunto de todos subconjuntos de um conjunto (isto é, o "conjunto potência" de Θ).

Agora, para cada substituição $\omega_i \in \Omega$, temos uma subfórmula V_c (que pode ser vista como um conjunto de *teoremas inferidos*) cuja forma conjuntiva é:

$$V_{c_i} = \lfloor Dual(W_d\, \omega_i) \rfloor$$

Finalmente, o conjunto:

$$\mho_c = \lfloor \bigcup_{\omega_i \in \Omega} V_{c_i} \rfloor$$

[4]Por "eliminar" queremos dizer "tornar explicitamente contraditória".

contém todas cláusulas inferidas obtidas pela aplicação das substituições de Ω. Uma vez obtido o conjunto de cláusulas $\mho_c$, podemos repetir este procedimento inúmeras vezes. Existem três possibilidades:

- (i) a fórmula inicial W_c é *contraditório* e após um número finito de ciclos o procedimento encontrará um $V_{c_i} = \emptyset$. Neste caso $V_{c_i} = \lfloor Dual(\emptyset) \rfloor$, isto é W_d ω_i é contraditório;

- (ii) após um número finito de ciclos o procedimento obtém um conjunto $\mho_c \subseteq W_c$. Nesta caso dizemos que a fórmula W_c é uma *teoria fechada*; ou

- (iii) a repetição do procedimento dá-se infinitamente devido à semi-decidibilidade da lógica de primeira ordem.

O método de inferência descrito acima pode ser usado como um provador de teoremas por refutação (caso i) e afirmativo (casos ii e iii). No caso proposicional, temos somente as possibilidades (i) e (ii).

5.2.2 Exemplos

Exemplo 20 *Associado ao seguinte conjunto insatisfazível de cláusulas:*

$$W_c = \langle [P], [\neg P, Q], [\neg Q, R], [\neg R] \rangle$$

temos o seguinte conjunto de cláusulas duais:

$$\begin{aligned} W_d &= [\langle \neg R, R, Q, P \rangle,\ \langle \neg R, \neg Q, Q, P \rangle, \\ &\quad \langle \neg R, \neg Q, \neg P, P \rangle,\ \langle \neg R, R, \neg P, P \rangle] \end{aligned}$$

onde cada cláusula dual é contraditória. Isto implica que, como esperado:

$$\mho_c = \emptyset$$

Exemplo 21 *Associado ao seguinte conjunto satisfazível de cláusulas:*

$$W_c = \langle [P], [\neg P, Q], [\neg Q, R], [\neg R, S, T] \rangle$$

temos o seguinte conjunto de cláusulas duais:

$$\begin{aligned} W_d \;=\; & [\langle P,\neg P,\neg Q,\neg R\rangle,\langle P,\neg P,\neg Q,S\rangle,\langle P,\neg P,\neg Q,T\rangle,\langle P,\neg P,R,\neg R\rangle,\\ & \langle P,\neg P,R,S\rangle,\langle P,\neg P,R,T\rangle,\langle P,Q,\neg Q,\neg R\rangle,\langle P,Q,\neg Q,S\rangle,\\ & \langle P,Q,\neg Q,T\rangle,\langle P,Q,R,\neg R\rangle,\langle P,Q,R,S\rangle,\langle P,Q,R,T\rangle] =\\ & [\langle P,Q,R,S\rangle,\langle P,Q,R,T\rangle] \end{aligned}$$

$$\Theta = \Omega = (\{\ \})$$

$$V_{c_1} = \langle [P],[Q],[R],[S,T]\rangle$$

e, finalmente:

$$\mho_c = \langle [P],[Q],[R],[S,T]\rangle$$

onde o conjunto de cláusulas $\mho_c$ é uma teoria fechada. É interessante observar em $\mho_c$ que, além da conclusão – Q – obtida por Modus Ponens, todas conclusões obtidas por Silogismo Hipotético – R and $[S,T]$ – foram obtidos em apenas um ciclo de inferência. De fato, para fórmulas da lógica proposicional o método de inferência definido sempre termina em apenas um ciclo.

Exemplo 22 *Considerando a fórmula:*

$$\{\forall x.(Homem(x) \to Mortal(x)), Homem(socrates)\} \vdash Mortal(socrates)$$

para provarmos o teorema por refutação, temos o seguinte conjunto insatisfazível de cláusulas:

$$W_c = \langle [\neg H(x), M(x)], [H(s)], [\neg M(s)]\rangle$$

e temos o seguinte conjunto de cláusulas duais:

$$W_d = [\langle \neg H(x), H(s), \neg M(s)\rangle, \langle M(x), H(s), \neg M(s)\rangle]$$

onde a primeira cláusula dual pode tornar-se contraditória com $\theta_{1,1} = \{x/s\}$ e a segunda cláusula dual com $\theta_{2,1} = \{x/s\}$. Como estas são as únicas maneiras de tornar aquelas cláusulas duais contraditórias, e como as duas substituições são iguais, temos $\theta_1 = \langle \{x/s\}, \{1,2\}\rangle$, $\Theta = \{\theta_1\}$ e $\Omega = \{\omega_1\}$ onde $\omega_1 = \langle \theta_1, \{1,2\}\rangle$.

Se aplicarmos ω_1 em W_d e retirarmos as cláusulas duais contraditórias, temos $W_{d3} = \emptyset$ o que implica $\mho_c = \emptyset$ como esperado.

Exemplo 23 *Considerando a fórmula do exemplo 22, porém com a retirada do teorema, temos o seguinte conjunto satisfazível de cláusulas:*

$$W_{c1} = \langle [\neg H(x), M(x)], [H(s)] \rangle$$

e o seguinte conjunto de cláusulas duais:

$$W_{d1} = [\langle \neg H(x), H(s) \rangle, \langle M(x), H(s) \rangle]$$

onde apenas a primeira cláusula dual é potencialmente contraditória com $\theta_{1,1} = \{x/s\}$. *Temos, então,* $\theta_1 = \langle \{x/s\}, \{1\} \rangle, \Theta = \{\theta_1\}$ *e* $\Omega = \{\omega_1\}$ *onde* $\omega_1 = \langle \theta_1, \{1\} \rangle$. *Ao aplicarmos* ω_1 *em* W_d *e retirarmos a cláusula contraditória, temos:*

$$W_{d1'} = [\langle M(s), H(s) \rangle]$$

o que implica

$$\mho_c = \langle [M(s)], [H(s)] \rangle$$

que são os teoremas inferidos pela fórmula original. Unindo-se estes à fórmula original obteremos um novo conjunto de cláusulas

$$W_{c2} = \langle [\neg H(x), M(x)], [H(s)], [M(s)] \rangle$$

do qual calculamos novamente a forma disjuntiva:

$$W_{d2} = [\langle \neg H(x), H(s) \rangle, \langle \neg H(x), M(s) \rangle, \langle M(x), H(s) \rangle, \langle \neg M(x), M(s) \rangle]$$

onde a primeira e a quarta cláusulas duais são potencialmente contraditórias com a mesma substituição $\{x/s\}$. *Temos então* $\Omega = \{\omega_1\}$ *onde* $\omega_1 = \langle \{x/s\}, \{1, 4\} \rangle$. *Ao aplicarmos* ω_1 *em* W_d *e retirarmos a cláusula contraditória, temos:*

$$W_{d2'} = [\langle \neg H(s), M(s) \rangle, \langle M(s), H(s) \rangle]$$

que produz o conjunto de teoremas gerados:

$$\mho_c = \langle [M(s)] \rangle$$

e, como $\mho_c \subseteq W_{c2}$, *a fórmula* W_{c2} *é uma teoria fechada.*

O método de inferência definido pode ser resumido na rotina *Pulsar*:

$Pulsar(W_c)$

1. Change variable names
2. $W_d \leftarrow \lfloor Dual(\lfloor W_c \rfloor) \rfloor$
3. $\Theta \leftarrow Contra(W_d)$
4. $\Omega \leftarrow \mathcal{P}(\Theta)$
5. $\mho_c \leftarrow \emptyset$
6. **for all** $\omega_i \in \Omega$ **do**
   $V_c \leftarrow \lfloor Dual(W_d\ \omega_i) \rfloor$
   if $V_c = \emptyset$
   then Return(W_c is contradictory)
   else $\mho_c \leftarrow \mho_c \cup \lfloor V_c \rfloor$
7. $\mho_c \leftarrow \lfloor \mho_c \cup W_c \rfloor$
8. **if** $\mho_c = W_c$
   then Return(W_c is a closed theory)
   else $Pulsar(\mho_c)$

A rotina *Pulsar* é um método de inferência correto e completo para refutação para lógica de primeira ordem. Em [6] é mostrado que esta versão determinística da rotina *Pulsar* é equivalente à estratégia de saturação de níveis da resolução. Encontramos também neste artigo uma versão não determinística desta rotina. Nesta versão introduz-se a escolha de determinados $\omega_i \in \Omega$ para os procedimentos do passo 5. Os critérios utilizados para esta escolha, dentre outros, contribuem na definição de estratégias de refinamento da prova para o método dual.

5.3 Conclusão

Foi apresentado o *Método Dual* um método de inferência correto e completo para refutação para lógica de primeira ordem. A versão determinística deste método foi definida na rotina denominada *Pulsar*. Pode-se então, começar a apresentar a estratégia desenvolvida para o *Método Dual*, iniciando-se pelo estudo de uma representação para as formas conjuntiva e disjuntiva de uma fórmula, assunto do próximo capítulo.

"Would you tell me, please, which way I ought to walk from here?" "That depends a good deal on where you want to get to," said the Cat. "I don't much care where" said Alice. "Then it doesn't matter which way you walk," said the Cat. "so long as I get somewhere," Alice added as an explanation. "Oh, you're sure to do that," said the Cat, "if you only walk long enough."

Lewis Carroll, Alice's Adventures in Wonderland, 1865

Capítulo 6

Caminhos e coordenadas

Muitas das relações existentes entre literais de uma fórmula advêm de sua disposição nas cláusulas destas fórmulas. A representação cruzada, isto é, a representação, através de coordenadas, da disposição dos literais nas formas conjuntiva e disjuntiva de uma fórmula lógica, torna explícita algumas destas relações. Nesta representação, a cada literal é atribuído um conjunto de coordenadas. Esta representação é utilizada no algoritmo citado no capítulo 4 (subseção 4.4.1) onde as coordenadas são calculadas durante a transformação dual e utilizadas para minimizar o cálculo desta transformação. Como estas coordenadas representam relações existentes entre os literais em ambas formas normais de uma fórmula, elas podem ser utilizadas para direcionar o processo de busca pela prova, auxiliando, portanto, o desenvolvimento de estratégias para o método dual.

6.1 A representação cruzada

Por definição, uma forma normal é calculada pela distribuição dos conectivos lógicos $\wedge$ (e) e $\vee$ (ou) da outra, ou seja, uma forma normal é calculada pela distribuição dos literais da outra forma, de modo que cada cláusula da forma resultante seja composta por um, e apenas um, literal de cada cláusula da forma original. Sendo uma fórmula em forma conjuntiva $W_c = \langle C_1, C_2, \ldots, C_n \rangle$, pela distributividade dos conectivos, a forma disjuntiva $W_d = [D_1, D_2, \ldots, D_n]$ não simplificada é o resultado da distribuição das cláusulas de W_c, i.e., $W_d = [C_1 \times C_2 \times \ldots \times C_n]$[1] e cada cláusula dual é uma tupla de literais resultantes desta distribuição.

Exemplo 24 *Sendo* $W_c = \langle [a, b, c], [d, e], [f] \rangle$ *a forma conjuntiva de uma fórmula,*

[1]Onde $\times$ é usado para representar a aplicação da distributividade dos conectivos lógicos.

$W_d = [\langle a,d,f\rangle, \langle a,e,f\rangle, \langle b,d,f\rangle, \langle b,e,f\rangle, \langle c,d,f\rangle, \langle c,e,f\rangle]$ *é sua forma disjuntiva.*

É interessante ressaltar que, apesar das formas conjuntiva e disjuntiva de uma fórmula serem equivalentes, como a distribuição é uma operação que "espalha" os literais, se a fórmula A é o resultado da distribuição da fórmula B, B não é o resultado da distribuição da fórmula A, mesmo que sejam consideradas suas formas condensadas. Por exemplo, na fórmula do exemplo anterior, W_d é o resultado da distribuição de W_c e é, também, uma forma condensada. Mesmo assim, W_c não é o resultado da distribuição de W_d. É fácil observar esta propriedade lembrando-se que, pela distributividade, o número de cláusulas da fórmula resultante é igual à multiplicação do número de literais de cada cláusula da fórmula de origem. Assim, o resultado da distribuição de W_d (que contém 6 cláusulas de 3 literais cada) é uma fórmula em forma conjuntiva contendo 3^6 (729) cláusulas e que pode ser simplificada para a fórmula W_c, com 3 cláusulas.

Seja l um literal, C e D cláusulas de W_c e W_d respectivamente, $F(l_{\in C}) = \{D \mid l \in D\}$ e $F(l_{\in D}) = \{C \mid l \in C\}$. O conjunto resultante de $F(l)$ é denominado *conjunto de coordenadas fixas*, ou simplesmente *coordenadas fixas*, do literal l. Por exemplo, considerando-se as fórmulas W_c e W_d do exemplo 24, as coordenadas fixas dos literais da cláusula $[a,b,c]$ são: $F(a_{\in[a,b,c]}) = \{\langle a,d,f\rangle, \langle a,e,f\rangle\}$, $F(b_{\in[a,b,c]}) = \{\langle b,d,f\rangle, \langle b,e,f\rangle\}$ e $F(c_{\in[a,b,c]}) = \{\langle c,d,f\rangle, \langle c,e,f\rangle\}$; e as coordenadas fixas dos literais da cláusula dual $\langle a,d,f\rangle$ são: $F(a_{\in\langle a,d,f\rangle}) = \{[a,b,c]\}$, $F(d_{\in\langle a,d,f\rangle}) = \{[d,e]\}$ e $F(f_{\in\langle a,d,f\rangle}) = \{[f]\}$.

Na representação cruzada, a cada literal é associado seu conjunto de coordenadas fixas. Nela, para simplificar a visualização, as cláusulas das formas conjuntiva e disjuntiva são numeradas, possibilitando que o conjunto de coordenadas fixas seja representado com o número de cada cláusula ao invés da cláusula propriamente dita.

Exemplo 25 *A fórmula do exemplo 24 na representação cruzada:*

$$\begin{aligned}
W_c &= \langle\ 1:[a^{\{1,2\}}, b^{\{3,4\}}, c^{\{5,6\}}], 2:[d^{\{1,3,5\}}, e^{\{2,4,6\}}], 3:[f^{\{1,2,3,4,5,6\}}]\ \rangle \\
W_d &= [\ 1:\langle a^{\{1\}}, d^{\{2\}}, f^{\{3\}}\rangle, 2:\langle a^{\{1\}}, e^{\{2\}}, f^{\{3\}}\rangle, 3:\langle b^{\{1\}}, d^{\{2\}}, f^{\{3\}}\rangle, \\
&\quad\ 4:\langle b^{\{1\}}, e^{\{2\}}, f^{\{3\}}\rangle, 5:\langle c^{\{1\}}, d^{\{2\}}, f^{\{3\}}\rangle, 6:\langle c^{\{1\}}, e^{\{2\}}, f^{\{3\}}\rangle\ ]
\end{aligned}$$

Por este exemplo vê-se que as coordenadas fixas de um literal de uma forma expressam exatamente sua distribuição, ou *ocorrência*, na outra forma da fórmula: as coordenadas fixas $\{1,2\}$ do literal "a" da cláusula 1 indicam que este ocorre (foi distribuído) nas cláusulas duais 1 e 2; já as coordenadas fixas $\{1\}$ do literal "a" da cláusula dual 1 indica que este ocorre (origina-se) unicamente na cláusula 1.

$$
\begin{array}{l}
1{:}[\ a^{\{1,2\}},\ b^{\{3,4\}},\ c^{\{5,6\}}\ ] \\
2{:}[\ d^{\{1,3,5\}},\ e^{\{2,4,6\}}\ ] \\
3{:}[\ f^{\{1,2,3,4,5,6\}}\ ]
\end{array}
$$

Figura 6.1: Caminho representado pela coordenada 1 na forma conjuntiva.

Além disto, sendo um *caminho* uma coleção[2] de literais composto por um, e apenas um, literal de cada cláusula (dual), vê-se que cada coordenada fixa representa um caminho. Por exemplo, a coordenada fixa 1, na forma conjuntiva, representa o caminho $\{a, d, f\}$, conforme mostrado na figura 6.1. Já, na forma disjuntiva esta coordenada representa o caminho $\{a, a, b, b, c, c\}$.

Da definição de caminho segue que cada caminho de uma forma representa uma cláusula da outra. Por exemplo, o caminho 1 de W_c, mostrado na figura 6.1, representada a cláusula dual 1 de W_d. Da mesma forma, por exemplo, o caminho 1 de W_d representa a cláusula 1 de W_c. Pelos caminhos confirma-se o que foi dito anteriormente: como W_d é o resultado da distribuição de W_c, a cada caminho possível de W_c corresponde uma coordenada (isto é, cláusulas de W_d); já em W_d isto não se verifica, o que confirma que W_c não é o resultado da distribuição de W_d.

6.2 As relações estruturais

Como já visto, as coordenadas fixas de uma forma conjuntiva refletem a distribuição dos literais na outra forma. Pode-se também dizer que estas coordenadas refletem a nova organização dos literais decorrente da distribuição. Baseados nisto, pode-se estabelecer relações estruturais refletidas pelas coordenadas fixas. Por estruturais, entende-se as relações advindas simplesmente da distribuição, independentemente da interpretação lógica estabelecida. Apesar da representação ser simétrica, o significado semântico de cada cláusula na forma conjuntiva não é o mesmo de cada cláusula na forma disjuntiva. Naquela, os literais representam uma conjunção de proposições e nesta uma disjunção, mudando completamente sua interpretação lógica. Assim, muda

[2] Usa-se o termo coleção para deixar explícito que propriedades de idempotência e comutatividade, típicas dos conjuntos, não são consideradas.

também o significado das relações expressas pelas coordenadas fixas de cada forma normal.

Para o estudo destas relações, inicialmente, assumimos a representação de uma fórmula na qual a forma disjuntiva é o resultado da distribuição da forma conjuntiva condensada. Devido à relevância no escopo deste trabalho, é dada ênfase às relações expressas pelas coordenadas da forma conjuntiva.

Nesta seção pretende-se abstrair o fato da distribuição ser de literais de uma fórmula lógica. (Poderia ser a distribuição de conjuntos quaisquer – números, pessoas, cartas de baralho, ..., tanto faz – de tal forma que para todos literais l_1 e l_2 da fórmula, $l_1 \neq l_2$. O importante é estabelecer quem relaciona-se com quem, antes e após a distribuição.) Para tal define-se:

- W_c e W_d as formas conjuntiva e disjuntiva de uma fórmula;
- N_c o conjunto dos números de cláusulas de W_c,
- N_d o conjunto dos números de cláusulas de W_d,
- C_i a i-ésima cláusula de W_c,$i = 1, 2, \ldots, N_c$,
- D_j a j-ésima cláusula de W_d,$j = 1, 2, \ldots, N_d$,
- $L(x)$ a coleção de literais de x;
- l um literal (e l_n o enésimo literal),
- $C(l)$ a cláusula à qual o literal l pertence,
- coordenada fixa *conjuntiva* refere-se à coordenada fixa de literal de W_c,
- coordenada fixa *disjuntiva* refere-se à coordenada fixa de literal de W_d,
- $F_c(l)$ o conjunto de coordenadas fixas conjuntivas do literal l,
- $F_d(l)$ o conjunto de coordenadas fixas disjuntivas do literal l,
- $T(x)$ o conjunto de tuplas resultantes da distribuição de x, e
- a expressão *fórmula dada* refere-se à fórmula apresentada no exemplo 25.

Considerando-se a representação das formas conjuntiva e disjuntiva não simplificada de uma fórmula pode-se estabelecer as seguintes relações estruturais (r.e.)[3]:

[3]As relações cujo texto inicia-se por "(*)" são referenciadas no decorrer do trabalho para futuras explicações.

1. como a distribuição de W_c é dada por $C_1 \times C_2 \times \ldots \times C_n$, um caminho de W_c é uma tupla de sua distribuição. Então, $T(W_c)$ é a coleção de todos os caminhos de W_c. Por exemplo, o caminho $\{a, d, f\}$ da fórmula dada é uma tupla da distribuição de W_c;

2. (*) como as coordenadas fixas conjuntivas de um literal são as cláusulas duais nas quais o literal ocorre, sendo n uma coordenada fixa conjuntiva, a coleção dos literais cujas coordenadas fixas conjuntivas contêm n é a coleção dos literais da cláusula dual de número n e, por conseguinte, um caminho de W_c: $\forall n \in N_d$ $(\{l \mid n \in F_c(l)\} = L(D_n))$ e $\forall n \in N_d$ $(\{l \mid n \in F_c(l)\} \in T(W_c))$;

   Por exemplo, na fórmula dada, (a, d, f) é a coleção dos literais cujas coordenadas fixas conjuntivas contém a coordenada 1. Então (a, d, f) é a coleção dos literais da cláusula dual número 1 $(\langle a^{\{1\}}, d^{\{2\}}, f^{\{3\}} \rangle)$ e é também um caminho de W_c;

3. como W_d é o resultado direto da distribuição de W_c, toda tupla desta distribuição é uma cláusula dual de W_d. E, como a toda cláusula dual corresponde uma coordenada fixa conjuntiva, a toda tupla da distribuição de W_c corresponde uma coordenada;

4. (*) decorrente da definição de distribuição, um literal de cláusula unitária é literal de todo caminho de W_c e, portanto, ocorre em todas cláusulas duais de W_d. Assim, toda coordenada $n \in N_d$ é coordenada fixa conjuntiva deste literal: $\forall l \in L(W_c), D_n \in W_d$ $(\{\, l \,\} = L(C) \rightarrow (l \in L(D_n)) \wedge (D_n \in F_c(l))$;

5. também decorrente da definição de distribuição, nenhum par de literais de mesma cláusula ocorre na mesma cláusula dual, ou seja, os conjuntos de coordenadas fixas de literais de mesma cláusula são disjuntos:
   $\forall C \in W_c, i \neq j, l_i \in C, l_j \in C$ $(F_c(l_i) \cap F_c(l_j) = \emptyset)$;

6. (*) como cada literal de uma cláusula C ocorre em cláusula dual diferente e toda cláusula dual contém um literal oriundo desta cláusula C, então a união dos conjuntos de coordenadas fixas conjuntivas dos literais de uma cláusula é igual ao conjunto de de cláusulas duais: $\forall C \in W_c$ $(\bigcup\{F_c(l) \mid l \in C\} = W_d)$;

7. (*) como todo o literal de uma mesma cláusula é distribuído com os mesmos literais das outras cláusulas, as tuplas da distribuição de W_c nas quais os literais de uma cláusula ocorrem diferem apenas por estes literais: $\forall C \in W_c, i \neq j, l_i \in C, l_j \in C, t \in T(W_c)$ $(l_i \in t \rightarrow ((t - \{l_i\} \cup \{l_j\}) \in T(W_c)))$.

Por exemplo, considerando-se a cláusula $C = [a, b, c]$ da fórmula dada, o caminho (a, d, f) é um caminho de W_c. Então os caminhos (b, d, f) e (c, d, f) também o são;

8. decorrente do item anterior, o número de caminhos de W_c nas quais o literal ocorre é o mesmo para todos os literais de mesma cláusula. E, como a cada um destes caminhos corresponde uma coordenada fixa conjuntiva, o número de coordenadas fixas conjuntivas dos literais de uma mesma cláusula é igual: $\forall C \in W_c, i \neq j, l_i \in C, l_j \in C \; (\mid F_c(l_i) \mid = \mid F_c(l_j) \mid)$.

   Por exemplo, o número de coordenadas fixas de cada literal da cláusula $[d^{\{1,3,5\}}, e^{\{2,4,6\}}]$ da fórmula dada é 3;

9. (*) como toda coleção de literais oriundos de cláusulas diferentes é um subconjunto de pelo menos um caminho de W_c: $\forall L = (l_1, \ldots, l_x), i \neq j, m \neq n, l_i \in L, l_j \in L, l_i \in C_m, l_j \in C_n, \; \exists t \; (t \in T(W_c) \wedge L \subseteq t)$. E como a todos estes caminhos corresponde uma coordenada fixa conjuntiva, as coordenadas fixas conjuntivas destes literais não são disjuntas: $\bigcap\{F_c(l) \mid l \in L\} \neq \emptyset$.

   Por exemplo, a intersecção das coordenadas fixas conjuntivas dos literais $(a^{\{1,2\}}, e^{\{2,4,6\}})$, oriundos de cláusulas diferentes da fórmula dada, é $\{2\}$ (de fato $2 : \langle a^{\{1\}}, e^{\{2\}}, f^{\{3\}}\rangle$ já que as coordenadas fixas conjuntivas são cláusulas). Da mesma forma, a intersecção das coordenadas fixas conjuntivas dos literais $(c^{\{5,6\}}, e^{\{2,4,6\}}, f^{\{1,2,3,4,5,6\}})$ é o conjunto $\{6\}$ (isto é $6 : \langle c^{\{1\}}, e^{\{2\}}, f^{\{3\}}\rangle$);

10. (*) dadas duas cláusulas, C_1 e C_2, L_1 a coleção de um ou mais literais de C_1 e L_2 a coleção de um ou mais literais de C_2, a união dos literais de $C_1 - L_1$ e $C_2 - L_2$ é uma tupla de literais da distribuição das cláusulas duais representadas pelas coordenadas fixas conjuntivas de $C_1 - L_1$ e $C_2 - L_2$ (ou todas as cláusulas duais exceto aquelas representadas pela intersecção das coordenadas fixas conjuntivas de L_1 e L_2): seja $C_1 = [l_i, \ldots, l_j, l_{j+1}, \ldots, l_k]$, $C_2 = [l_m, \ldots, l_n, l_{n+1}, \ldots, l_o]$, $L_1 = (l_i, \ldots, l_j)$, $L_2 = l_m, \ldots, l_n$, $t = (l_{j+1}, \ldots, l_k, l_{n+1}, \ldots, l_o)$

    - $t \in T(\{D_x \mid x \in \bigcup F_c(l \in t)\})$ que é o mesmo que
    - $t \in T(W_d - \{ \; D_x \mid D_x \in (\bigcup F_c(l \in L_1) \cap \bigcup F_c(l \in L_2)) \; \})$.

    Por exemplo, seja $C_1 = [a^{\{1,2\}}, b^{\{3,4\}}, c^{\{5,6\}}]$ e $C_2 = [d^{\{1,3,5\}}, e^{\{2,4,6\}}]$, cláusulas da fórmula dada, tem-se:

    - para $L_1 = (a)$ e $L_2 = (d)$, (b, c, e) é uma tupla de

- $D_2 \times D_3 \times D_4 \times D_5 \times D_6$ pois
  $\bigcup\{F_c(b^{\{3,4\}}), F_c(c^{\{5,6\}}), F_c(e^{\{2,4,6\}})\} = \{2,3,4,5,6\}$, e
- $W_d - \{D_1\}$ pois $F_c(a^{\{1,2\}}) \cap F_c(d^{\{1,3,5\}}) = \{1\}$;

- para $L_1 = (b, c)$ e $L_2 = (e)$, (a, d) é uma tupla de
  - $D_1 \times D_2 \times D_3 \times D_5$ pois
    $\bigcup\{F_c(a^{\{1,2\}}), F_c(d^{\{1,3,5\}})\} = \{1,2,3,5\}$, e
  - $W_d - \{D_4, D_6\}$ pois
    $\bigcup\{F_c(b^{\{3,4\}}), F_c(c^{\{5,6\}})\} \cap F_c(e^{\{2,4,6\}}) = \{4,6\}$;

entre outras.

Esta relação pode ser assim explicada: como toda cláusula contém todas as coordenadas fixas conjuntivas (r.e., 6), as únicas coordenadas fixas conjuntivas que a união de $C_1 - L_1$ com $C_2 - L_2$ não contém são aquelas que ocorrem somente em L_1 e em L_2 (i.e. que não ocorrem em $C_1 - L_1$ nem em $C_2 - L_2$), que com certeza são as da intersecção das coordenadas fixas conjuntivas de L_1 e L_2. As demais coordenadas fixas de L_1, que não interseccionam com L_2, ocorrem em $C_2 - L_2$. Da mesma forma, as demais coordenadas fixas de L_2 ocorrem em $C_1 - L_1$. Pode-se então afirmar que a união de $C_1 - L_1$ com $C_2 - L_2$ contém todas coordenadas fixas conjuntivas de W_d exceto aquelas da intersecção. De onde se conclui que aquela união é uma tupla da distribuição da fórmula disjuntiva resultante da eliminação das cláusulas duais representadas por estas coordenadas da intersecção, já que cada literal da união ocorre em pelo menos uma cláusula dual da fórmula resultante. Esta propriedade pode pode ser estendida para um conjunto de cláusulas e, mais especificamente, para uma única cláusula, conforme mostrado na próxima relação;

11. (*) dada uma cláusula C e uma coleção de literais L de C, a coleção de literais de $C' = C - L$ é tupla da distribuição das cláusulas duais representadas pelas coordenadas fixas conjuntivas dos literais de C': seja $C = [l_i, \ldots, l_j, l_{j+1}, \ldots, l_k]$, $L = (l_i, \ldots, l_j)$, $t = (l_{j+1}, \ldots, l_k)$

    - $t \in T(\{D_x \mid x \in \bigcup F_c(l \in t)\})$ que é o mesmo que
    - $t \in T(W_d - \{\, D_x \mid x \in (\bigcup F_c(l \in L))\, \})$.

Por exemplo, seja $C = [a^{\{1,2\}}, b^{\{3,4\}}, c^{\{5,6\}}]$ uma cláusula da fórmula dada, tem-se:

- para $L = (a)$, (b, c) é uma tupla de

- $D_3 \times D_4 \times D_5 \times D_6$ pois
  $\bigcup\{F_c(b^{\{3,4\}}), F_c(c^{\{5,6\}})\} = \{3,4,5,6\}$ $\{D_3, D_4, D_5, D_6\}$), e
- $W_d - \{D_1, D_2\}$ pois $\bigcup\{F_c(a^{\{1,2\}})\} = \{1,2\}$;

• para $L = (a,b)$, (c) é uma tupla de

- $D_5 \times D_6$ pois $\bigcup\{F_c(c^{\{5,6\}})\} = \{5,6\}$, e
- $W_d - \{D_1, D_2, D_3, D_4\}$ pois
  $\bigcup\{F_c(a^{\{1,2\}}), F_c(b^{\{3,4\}})\} = \{1,2,3,4\}$;

Estas relações são válidas para este domínio: coordenadas fixas de uma fórmula lógica, proposicional ou de primeira ordem, cuja forma disjuntiva é o resultado da distribuição dos literais da forma conjuntiva, sem simplificação alguma. Neste caso vemos que as coordenadas são bem distribuídas entre cada literal de uma cláusula, que todos os caminhos da forma conjuntiva são representados pelas coordenadas fixas conjuntivas e que cada um destes caminhos/coordenadas coincide com uma cláusula dual da forma disjuntiva.

6.3 O caso proposicional

6.3.1 Coordenadas e inferência

É interessante observar como uma inferência lógica pode ser vista em termos das coordenadas da forma conjuntiva. Normalmente, nos métodos de prova automática, a inferência baseia-se nas contradições explícitas existentes na fórmula, i.e., na existência de literais complementares oriundos de cláusulas diferentes. Este é o caso, por exemplo, no método de Resolução onde *resolvem-se* cláusulas, contendo um par complementar, gerando-se um *resolvente*, i.e., um novo teorema implicado por aquelas cláusulas.

Exemplo 26 *Considerando-se a fórmula contraditória*

$$\begin{array}{rcll} W_c & = & \langle & 1: \ [\neg p^{\{1,2\}}, q^{\{3,4\}}\ ] \\ & & & 2: \ [\neg q^{\{1,3\}}, r^{\{2,4\}}\ ] \\ & & & 3: \ [\, p^{\{1,2,3,4\}}\ ] \\ & & & 4: \ [\neg r^{\{1,2,3,4\}}\ ]\ \rangle \\ W_d & = & [& 1: \ \langle \neg p^{\{1\}}, \neg q^{\{2\}}, p^{\{3\}}, \neg r^{\{4\}}\ \rangle \\ & & & 2: \ \langle \neg p^{\{1\}}, r^{\{2\}}, p^{\{3\}}, \neg r^{\{4\}}\ \rangle \\ & & & 3: \ \langle\, q^{\{1\}}, \neg q^{\{2\}}, p^{\{3\}}, \neg r^{\{4\}}\ \rangle \\ & & & 4: \ \langle\, q^{\{1\}}, r^{\{3\}}, p^{\{3\}}, \neg r^{\{4\}}\ \rangle\ ] \end{array}$$

- *a resolução das cláusulas 1 e 3, que contêm o par* $(\neg p, p)$*, produzindo o resolvente* $[\, q\,]$*, e*

- *a resolução das cláusulas 1 e 2, que contêm o par* $(q, \neg q)$*, produzindo o resolvente* $[\neg p, r\,]$*,*

são possíveis inferências no método de Resolução.

No método de Resolução, uma inferência é feita pela *regra da Resolução.* No entanto, ela pode também ser vista em termos das coordenadas do par complementar das cláusulas resolvidas. Como nenhum par de literais oriundos de cláusulas diferentes tem conjuntos de coordenadas disjuntos (r.e., 9), todo par de literais oriundos de cláusulas diferentes faz parte de pelo menos uma mesma cláusula dual, inclusive os pares de literais complementares escolhidos para inferência. No exemplo 26,

- pela intersecção dos conjuntos de coordenadas do par de literais complementares $(\neg p^{\{1,2\}}, p^{\{1,2,3,4\}})$, o conjunto $\{1,2\}$, sabe-se que os literais ocorrem simultaneamente nas cláusulas duais 1 e 2; e

- pela intersecção dos conjuntos de coordenadas do par de literais complementares $(q^{\{3,4\}}, \neg q^{\{1,3\}})$, o conjunto $\{3\}$, sabe-se que este par de literais ocorre simultaneamente somente na cláusula dual 3.

Por definição, cláusulas duais contraditórias podem ser removidas de uma fórmula disjuntiva sem alterar sua satisfazibilidade. Assim, sempre que um par de literais complementares ocorrer na mesma cláusula dual, esta pode ser removida. Sendo W_d a forma disjuntiva de uma fórmula e D uma cláusula dual contraditória de W_d e $W_{d'} =$

$W_d - \{D\}$, tem-se que $W_d \leftrightarrow W_{d'}$. Sendo $W_{c'}$ a forma conjuntiva de $W_{d'}$, $W_{d'} \leftrightarrow W_{c'}$. Assim, $W_d \leftrightarrow W_{c'}$ e como $W_d \leftrightarrow W_c$, tem-se que $W_c \leftrightarrow W_{c'}$. Finalmente, como $W_c \leftrightarrow W_{c'} \rightarrow \forall C \in W_{c'}(W_c \rightarrow C)$, o conjunto $I = W_{c'} - W_c$ pode ser visto como o conjunto de teoremas inferidos pela fórmula, obtidos tão somente pela eliminação de contradições explícitas de sua forma disjuntiva.

Exemplo 27 *Por exemplo, as cláusulas duais 1 e 2 do exemplo 26 são contraditórias devido à ocorrência simultânea dos literais complementares* $\neg p^{\{1,2\}}$ *e* $p^{\{1,2,3,4\}}$. *Com a remoção destas cláusulas duais tem-se:*

$$\begin{aligned} W_{d'} = [\ & 3: \ \langle\, q^{\{1\}}, \neg q^{\{2\}}, p^{\{3\}}, \neg r^{\{4\}} \,\rangle \\ & 4: \ \langle\, q^{\{1\}}, r^{\{2\}}, p^{\{3\}}, \neg r^{\{4\}} \,\rangle\ ] \end{aligned}$$

cuja forma conjuntiva é $W_{c'} = \langle [q], [\neg q, r], [p], [\neg r] \rangle$ *e o conjunto de teoremas inferidos é* $I = \langle\ [q]\ \rangle$. *Já a cláusula dual 3 é contraditória pela ocorrência simultânea dos literais* $q^{\{3,4\}}$ *e* $\neg q^{\{1,3\}}$. *Com a remoção desta cláusula tem-se:*

$$\begin{aligned} W_{d'} = [\ & 1: \ \langle\, \neg p^{\{1\}}, \neg q^{\{2\}}, p^{\{3\}}, \neg r^{\{4\}} \,\rangle \\ & 2: \ \langle\, \neg p^{\{1\}}, r^{\{2\}}, p^{\{3\}}, \neg r^{\{4\}} \,\rangle \\ & 4: \ \langle\, q^{\{1\}}, r^{\{2\}}, p^{\{3\}}, \neg r^{\{4\}} \,\rangle\ ] \end{aligned}$$

cuja forma conjuntiva é $W_{c'} = \langle [\neg p, q], [\neg p, r], [\neg q, r], [p], [\neg r] \rangle$ *e o conjunto de teoremas inferidos é* $I = \langle\ [\neg p, r]\ \rangle$.

Como pode-se verificar, os teoremas inferidos no exemplo anterior são os mesmos do exemplo 26, inferidos por resolução. Para inferi-los foram necessários os seguintes procedimentos: (i)remoção das cláusulas duais contraditórias (cálculo de $W_{d'}$), (ii)cálculo da forma conjuntiva ($W_{c'}$) e (iii)cálculo do conjunto de novos teoremas (I). Esta é praticamente a mesma forma adotada para explicar a inferência para o método dual, apresentada no capítulo 5. Porém, para explicar esta inferência pelas coordenadas adota-se outra explicação, baseada na r.e. 10.

Por esta relação é possível determinar tuplas da distribuição de subconjuntos de cláusulas duais sem efetuar a distribuição propriamente dita. Por tratar-se de uma relação estrutural da distribuição de conjuntos quaisquer, nada foi dito sobre equivalência ou implicação destes. Porém, sendo W_d a forma disjuntiva de uma fórmula lógica e $W_{d'}$ o conjunto resultante da remoção de cláusulas duais contraditórias de W_d, então $W_d \leftrightarrow W_{d'}$ e, como visto anteriormente, $W_c \leftrightarrow W_{c'}$. Então, no caso de W_d e

$W_{d'}$ serem equivalentes, W_c implica toda a cláusula cujos literais formem uma tupla da distribuição de $W_{d'}$.

Para ilustrar o que foi dito, considera-se a fórmula do exemplo 26 e os mesmos pares de literais complementares utilizados anteriormente:

- para o par $(\neg p^{\{1,2\}}, p^{\{1,2,3,4\}})$ tem-se $C_1 = [\neg p^{\{1,2\}}, q^{\{3,4\}}]$, $L_1 = \{\neg p^{\{1,2\}}\}$, $C_2 = [p^{\{1,2,3,4\}}]$, $L_2 = \{p^{\{1,2,3,4\}}\}$, $(q) \in T(W_d - \{D_1, D_2\})$ e $W_c \to [q]$. Portanto $[q]$ é um teorema inferido por W_c;

- para o par $(q^{\{3,4\}}, \neg q^{\{1,3\}})$ tem-se $C_1 = [\neg p^{\{1,2\}}, q^{\{3,4\}}]$, $L_1 = \{q^{\{3,4\}}\}$, $C_2 = [\neg q^{\{1,3\}}, r^{\{2,4\}}]$, $L_2 = \{\neg q^{\{1,3\}}\}$, $(\neg p, r) \in T(W_d - \{D_3\})$ e $W_c \to [\neg p, r]$. Portanto $[\neg p, r]$ é um teorema inferido por W_c.

Os teoremas inferidos no exemplo anterior, baseados na r.e. 10, também são os mesmos dos anteriormente inferidos por resolução no exemplo 26. De fato, pela r.e. 10 pode-se concluir que a inferência pela resolução de duas cláusulas que contém um par de literais complementares é equivalente à inferência pela remoção das cláusulas duais que contém a ocorrência simultânea deste par.

6.3.2 Eliminação de literais

Chamamos *eliminação de literais* a determinação de fórmulas (ou instâncias de fórmulas, na lógica de primeira ordem) nas quais um literal, por ser semanticamente irrelevante, pode ser eliminado. Este é o caso da fórmula $W_{c'}$ na primeira inferência do exemplo 27, na qual o literal $\neg p$, considerado para a inferência, não ocorre. Nesta fórmula, uma das cláusulas eleitas para inferência não está contida ($W_c \not\subset W_{c'}$). No entanto, na segunda inferência daquele exemplo, a fórmula $W_{c'}$ gerada contém, além do novo teorema, todas as cláusulas originais de W_c ($W_c \subset W_{c'}$). Isto pode ser explicado pelas coordenadas dos literais envolvidos naquelas inferências:

- como $F_c(\neg p^{\{1,2\}}) \subset F_c(p^{\{1,2,3,4\}})$, em todas cláusulas duais nas quais o literal $\neg p$ ocorre, ocorre também o literal p. Por serem contraditórias estas cláusulas duais são removidas eliminando, com isto, todas ocorrências de $\neg p$ da fórmula. Diz-se que o literal $\neg p$ foi *eliminado* da fórmula. (Se toda ocorrência de $\neg p$ na forma disjuntiva torna a cláusula dual contraditória, então sua própria ocorrência na fórmula é contraditória e não pode ser assumida verdadeira deixando a determinação do valor verdade da fórmula na dependência dos demais literais.);

- já no par $(q^{\{3,4\}}, \neg q^{\{1,3\}})$ nenhum conjunto de coordenadas é subconjunto do outro. Então, nenhuma ocorrência de literal é completamente eliminada durante a remoção da cláusula dual contraditória. Portanto o conjunto $W_{c'}$ gerado deve conter ocorrências de ambos literais envolvidos na inferência.

No caso onde ocorre a eliminação de um literal, o conjunto $W_{c'}$ produzido é uma fórmula, equivalente à W_c, cuja determinação do valor verdade independe do literal eliminado. Sendo irrelevante, o literal pode ser removido de sua cláusula de origem sem afetar o valor verdade da mesma. No exemplo, como $\langle\ [\neg p, q], [p]\ \rangle \equiv \langle\ [q], [p]\ \rangle^4$, pode-se dizer que o teorema gerado $[q]$ é o resultado da eliminação do literal irrelevante $\neg p$ de sua cláusula de origem $[\neg p, q]$ (na fórmula $W_{c'}$ resultante, esta cláusula foi substituída pelo novo teorema).

Portanto, com a eliminação de um literal, a cláusula contendo os demais literais não eliminados *é teorema de* W_c. A r.e. 11 confirma esta inferência: se um literal é eliminado é porque todas as cláusulas duais nas quais ele ocorre são contraditórias e podem ser removidas de W_d gerando $W_{d'}$; e, por esta r.e., sabe-se que a coleção dos literais remanescentes é tupla da distribuição $W_{d'}$ e é, portanto uma cláusula do conjunto de teoremas inferidos $W_{c'}$.

Pela r.e. 4, sabe-se que um literal de uma cláusula unitária ocorre em todas cláusulas duais. Então todo literal de uma fórmula proposicional conjuntiva que é complementar à um literal de uma cláusula unitária, é irrelevante e pode ser removido de sua cláusula. Isto não significa que um literal é irrelevante somente quando contradiz um literal de uma cláusula unitária. Um literal é irrelevante quando todas as cláusulas duais, nas quais ele ocorre, são contraditórias (independentemente do literal fazer parte do par complementar que torna a cláusula dual contraditória). No exemplo, o literal $q^{\{3,4\}}$ contradiz o literal $\neg q^{\{1,3\}}$, que não pertence a uma cláusula unitária. Neste caso, apenas a cláusula dual 3 é contraditória devido a ocorrência destes literais. No entanto, o par de literais $(r^{\{2,4\}}.\neg r^{\{1,2,3,4\}})$ ocorre simultaneamente nas cláusulas duais 2 e 4, tornando-as contraditórias. Assim, como as cláusulas duais 3 e 4 são contraditórias, o literal $q^{\{3,4\}}$ é irrelevante e pode ser eliminado de sua cláusula de origem.

Pelo que foi demonstrado, ambos literais da cláusula $[\neg p^{\{1,2\}}, q^{\{3,4\}}]$ são irrelevantes e podem ser eliminados, dando origem (ou inferindo) uma cláusula vazia que, sabe-se, pode somente ser inferida de uma fórmula contraditória (que é o caso, conforme enunciado do exemplo 26). Conforme a r.e. 6 todas as cláusulas duais estão representadas nas coordenadas dos literais de uma cláusula; então se todos literais podem ser eliminados é porque todas as cláusulas duais da fórmula são contraditórias, fato que também confirma a insatisfazibilidade da fórmula. Por fim, como toda coordenada

---

4 $((\neg p \vee q) \wedge p) \equiv ((\neg p \wedge p) \vee (p \wedge q)) \equiv (p \wedge q)$

representa uma tupla da distribuição de W_c (r.e. 2) e todos pares complementares tem pelo menos uma coordenada comum (r.e. 9), estas coordenadas comuns ao par de literais complementares representam tuplas contraditórias.

A simples constatação de todas as cláusulas duais de uma fórmula serem contraditórias é prova de sua insatisfazibilidade ou, é sua refutação. Da mesma forma o é, a comprovação da irrelevância de todos os literais de uma cláusula. Esta busca pode parecer um pouco "descabida" para o caso proposicional, mas pode auxiliar bastante o processo de refutação de fórmulas da lógica de primeira ordem.

6.4 O caso de primeira ordem

Na seção anterior as relações estruturais foram estabelecidas para fórmulas cuja forma disjuntiva é o resultado da distribuição da forma conjuntiva, sem considerar nenhuma simplificação. Neste caso, as relações estruturais estabelecidas para fórmulas proposicionais mantêm-se para fórmulas de primeira ordem. (Lembre-se que, quando as relações estruturais foram introduzidas, afirmou-se poder tratar-se de conjuntos quaisquer. Então pode ser uma fórmula da lógica de primeira ordem.) No exemplo 28 tem-se uma fórmula de primeira ordem representada na representação cruzada.

Exemplo 28 *Uma fórmula de primeira ordem contraditória:*

$$\begin{array}{rclll}
W_c & = & \langle & 1 & [\neg P(x_1)^{\{1,2\}}, Q(x_1)^{\{3,4\}}] \\
 & & & 2 & [\neg Q(x_2)^{\{1,3\}}, R(x_2)^{\{2,4\}}] \\
 & & & 3 & [P(a)^{\{1,2,3,4\}}] \\
 & & & 4 & [\neg R(a)^{\{1,2,3,4\}}] \ \rangle \\
W_d & = & [& 1 & \langle \neg P(x_1)^{\{1\}}, \neg Q(x_2)^{\{2\}}, P(a)^{\{3\}}, \neg R(a)^{\{4\}} \rangle \\
 & & & 2 & \langle \neg P(x_1)^{\{1\}}, R(x_2)^{\{2\}}, P(a)^{\{3\}}, \neg R(a)^{\{4\}} \rangle \\
 & & & 3 & \langle Q(x_1)^{\{1\}}, \neg Q(x_2)^{\{2\}}, P(a)^{\{3\}}, \neg R(a)^{\{4\}} \rangle \\
 & & & 4 & \langle Q(x_1)^{\{1\}}, R(x_2)^{\{2\}}, P(a)^{\{3\}}, \neg R(a)^{\{4\}} \rangle \ ]
\end{array}$$

Se as relações estruturais não sofrem alteração alguma, este não é o caso com a inferência baseada nas coordenadas. A diferença básica vem do conceito de contradição que, na lógica de primeira ordem, considera instâncias da fórmula. Uma fórmula da lógica proposicional, quando contraditória, o é *explicitamente*. Já uma fórmula da lógica de primeira ordem pode ser contraditória explicitamente, analogamente às fórmulas proposicionais, ou potencialmente, quando uma de suas instâncias o for.

Exemplo 29 *Considerando-se a fórmula do exemplo 28,*

- *a resolução das cláusulas 1 e 3 que contêm o par complementar* $(\neg P(x_1), P(a))$ *que produz o resolvente* $[Q(a)]$, *e*

- *a resolução das cláusulas 1 e 2 que contêm o par complementar* $(Q(x_1), \neg Q(x_2))$ *que produz o resolvente* $[\neg P(x_2), R(x_2)]$,

são inferências possíveis no método de Resolução. Já pelas coordenadas,

- *a intersecção das coordenadas fixas conjuntivas do par complementar* $(\neg P(x_1)^{\{1,2\}}, P(a)^{\{1,2,3,4\}})$ *é o conjunto* $\{1, 2\}$. *Portanto estes literais ocorrem simultaneamente nas cláusulas duais 1 e 2; e*

- *como a intersecção das coordenadas fixas conjuntivas do par de literais complementares* $(Q(x_1)^{\{3,4\}}, \neg Q(x_2)^{\{1,3\}})$ *é o conjunto* $\{3\}$, *este par de literais ocorre simultaneamente na cláusula dual 3.*

Sempre que um par de literais complementares ocorre na mesma cláusula dual D de W_d, pode-se dizer que, na instância θ de W_d $(W_d\theta)$ D é explicitamente contraditória e pode ser removida de $W_d\theta$. Então, sendo $W_{d'}\theta = W_d\theta - \{D\}$, tem-se que $W_d\theta \leftrightarrow W_{d'}\theta$ e, como $W_d \rightarrow W_d\theta$, $W_d \rightarrow W_{d'}\theta$. Sendo $W_{c'}\theta$ a forma conjuntiva de $W_{d'}\theta$, $W_{d'}\theta \leftrightarrow W_{c'}\theta$. Assim $W_d \rightarrow W_{c'}\theta$ e, finalmente, como $W_d \leftrightarrow W_c$, tem-se que $W_c \rightarrow W_{c'}\theta$. O conjunto $W_{c'}\theta$ pode ser visto como uma instância, ou um conjunto de teoremas, inferido por W_c pela aplicação da substituição θ. Por sua vez, o conjunto $I\theta = W_{c'}\theta - W_c$, pode ser visto como um conjunto de *novos* teoremas inferidos. Observa-se que estas definições diferem levemente das apresentadas para a lógica proposicional, na seção anterior. A razão para tal é a existência de instâncias da fórmula fazendo com que algumas equivalências de fórmulas proposicionais tornem-se implicações de fórmulas de primeira ordem.

Exemplo 30 *O par* $(\neg P(x_1)^{\{1,2\}}, P(a)^{\{1,2,3,4\}})$, *da fórmula do exemplo anterior, torna-se explicitamente contraditório com a aplicação da substituição* $\theta_1 = \{x_1/a\}$. *Então, a instância* θ_1 *do conjunto de cláusulas duais:*

$$
\begin{array}{rcll}
W_d\theta_1 & = & [& 1 \quad \langle \neg P(a)^{\{1\}}, \neg Q(x_2)^{\{2\}}, P(a)^{\{3\}}, \neg R(a)^{\{4\}} \rangle \\
& & & 2 \quad \langle \neg P(a)^{\{1\}}, R(x_2)^{\{2\}}, P(a)^{\{3\}}, \neg R(a)^{\{4\}} \rangle \\
& & & 3 \quad \langle Q(a)^{\{1\}}, \neg Q(x_2)^{\{2\}}, P(a)^{\{3\}}, \neg R(a)^{\{4\}} \rangle \\
& & & 4 \quad \langle Q(a)^{\{1\}}, R(x_2)^{\{2\}}, P(a)^{\{3\}}, \neg R(a)^{\{4\}} \rangle \ ]
\end{array}
$$

tem suas primeira e segunda cláusulas duais explicitamente contraditórias e podem ser removidas produzindo o conjunto:

$$\begin{aligned} W_{d'}\theta_1 = [\ & 3 \ \ \langle Q(a)^{\{1\}}, \neg Q(x_2)^{\{2\}}, P(a)^{\{3\}}, \neg R(a)^{\{4\}} \rangle \\ & 4 \ \ \langle Q(a)^{\{1\}}, R(x_2)^{\{2\}}, P(a)^{\{3\}}, \neg R(a)^{\{4\}} \rangle] \end{aligned}$$

cuja forma conjuntiva é $W_{c'}\theta_1 = \langle [Q(a)], [\neg Q(x_2), R(x_2)], [P(a)], [\neg R(a)] \rangle$*, e o conjunto de novos teoremas inferidos é* $I\theta_1 = \langle [Q(a)] \rangle$*.*

Já o par $(Q(x_1)^{\{3,4\}}, \neg Q(x_2)^{\{1,3\}})$ *torna-se explicitamente contraditório com a aplicação da substituição* $\theta_2 = \{x_1/x_2\}$*. Pelas coordenadas sabe-se que a instância* θ_2 *do conjunto de cláusulas duais tem a terceira cláusula dual explicitamente contraditória. Usando o mesmo raciocínio anterior, os seguintes conjuntos são produzidos:*

$$\begin{aligned} W_d\theta_2 = [\ & 1 \ \ \langle \neg P(x_2)^{\{1\}}, \neg Q(x_2)^{\{2\}}, P(a)^{\{3\}}, \neg R(a)^{\{4\}} \rangle \\ & 2 \ \ \langle \neg P(x_2)^{\{1\}}, R(x_2)^{\{2\}}, P(a)^{\{3\}}, \neg R(a)^{\{4\}} \rangle \\ & 3 \ \ \langle Q(x_2)^{\{1\}}, \neg Q(x_2)^{\{2\}}, P(a)^{\{3\}}, \neg R(a)^{\{4\}} \rangle \\ & 4 \ \ \langle Q(x_2)^{\{1\}}, R(x_2)^{\{2\}}, P(a)^{\{3\}}, \neg R(a)^{\{4\}} \rangle \ ] \end{aligned}$$

$$\begin{aligned} W_{d'}\theta_2 = [\ & 1 \ \ \langle \neg P(x_2)^{\{1\}}, \neg Q(x_2)^{\{2\}}, P(a)^{\{3\}}, \neg R(a)^{\{4\}} \rangle \\ & 2 \ \ \langle \neg P(x_2)^{\{1\}}, R(x_2)^{\{2\}}, P(a)^{\{3\}}, \neg R(a)^{\{4\}} \rangle \\ & 4 \ \ \langle Q(x_2)^{\{1\}}, R(x_2)^{\{2\}}, P(a)^{\{3\}}, \neg R(a)^{\{4\}} \rangle \ ] \end{aligned}$$

$$\begin{aligned} W_{c'}\theta_2 = \langle \ \ [& \ \neg P(x_2), Q(x_2)], [\neg P(x_2), R(x_2)], \\ [& \ \neg Q(x_2), R(x_2)], [P(a)], [\neg R(a)] \rangle \end{aligned}$$

$$I\theta = \langle \ [\ \neg P(x_2), R(x_2)] \ \rangle.$$

6.4.1 Eliminação de literais

A eliminação de literais em fórmulas de primeira ordem deve também considerar as instâncias que tornam os pares de literais explicitamente contraditórios. Assim, considerando-se o que foi dito anteriormente sobre eliminação de literais e os exemplos 28 e 30 tem-se que:

- pelas coordenadas fixas conjuntivas de $(\neg P(x_1)^{\{1,2\}}, P(a)^{\{1,2,3,4\}})$, vê-se que em todas as cláusulas duais nas quais $\neg P(x_1)$ ocorre, $P(a)$ também ocorre, ou seja, que $F(\neg P(x_1)) \subset F(P(a))$. Portanto o literal $\neg P(x_1)$ foi eliminado da instância $W_{c'}\theta_1$;

- o par $(Q(x_1)^{\{3,4\}}, \neg Q(x_2)^{\{1,3\}})$ não tem nenhum conjunto de coordenadas fixas conjuntivas incluso no outro. Então nenhuma ocorrência de literal é completamente excluída. Então o conjunto $W_{c'}\theta_2$ gerado deve conter ambos literais.

Sabe-se também que a eliminação de um literal cujas coordenadas fixas conjuntivas não estejam inclusas nas coordenadas fixas conjuntivas de seu literal complementar, como é o caso de $Q(x_1)^{\{3,4\}}$, pode dar-se pela eliminação simultânea de cláusulas duais que contêm diferentes pares complementares. Na fórmula do exemplo, pelo par $(R(x_2)^{\{2,4\}}, \neg R(a)^{\{1,2,3,4\}})$, sabe-se que as cláusulas duais 2 e 4 são explicitamente contraditórias na instância $\theta_3 = \{x_2/a\}$. Então, se na instância $\theta_2 = \{x_1/x_2\}$ a cláusula dual 3 é explicitamente contraditória e na instância $\theta_3 = \{x_2/a\}$ as cláusulas duais 2 e 4 são explicitamente contraditórias, então na instância $\theta_4 = \{x_1/x_2, x_2/a\}$ as cláusulas duais 2,3 e 4 são explicitamente contraditórias e podem ser removidas. Assim seguindo o mesmo raciocínio, tem-se os seguintes conjuntos para esta instância:

$$\begin{aligned} W_d = [\ & 1\ \langle \neg P(a)^{\{1\}}, \neg Q(a)^{\{2\}}, P(a)^{\{3\}}, \neg R(a)^{\{4\}} \rangle \\ & 2\ \langle \neg P(a)^{\{1\}}, R(a)^{\{2\}}, P(a)^{\{3\}}, \neg R(a)^{\{4\}} \rangle \\ & 3\ \langle Q(a)^{\{1\}}, \neg Q(a)^{\{2\}}, P(a)^{\{3\}}, \neg R(a)^{\{4\}} \rangle \\ & 4\ \langle Q(a)^{\{1\}}, R(a)^{\{2\}}, P(a)^{\{3\}}, \neg R(a)^{\{4\}} \rangle\ ] \end{aligned}$$

$$W_{d'}\theta_4 = [\ 1\ \langle \neg P(a)^{\{1\}}, \neg Q(a)^{\{2\}}, P(a)^{\{3\}}, \neg R(a)^{\{4\}} \rangle$$

$$W_{c'}\theta_4 = \langle\ [\ \neg P(a)], [\neg Q(a)], [P(a)], [\neg R(a)] \rangle$$

$$I\theta = \langle\ [\ \neg P(a)], [\neg Q(a)]\ \rangle.$$

Como pode-se observar, de fato $Q(x_1)^{\{3,4\}}$ foi eliminado de $W_{c'}\theta_4$. No entanto, observa-se também que a cláusula dual 1 de $W_{d'}\theta_4$ e, por conseguinte $W_{c'}\theta_4$, são também explicitamente contraditórias. Isto ocorre porque ao considerar a instância $\theta_4 = \{x_1/x_2, x_2/a\}$ considerou-se implicitamente a instância $\theta_1 = \{x_1/a\}$ que torna as cláusulas duais 1 e 2 explicitamente contraditórias. Assim, a instância θ_4 torna todas as cláusulas duais explicitamente contraditórias e $W_{c'}\theta_4$ poderia ser substituído por $W_{c''}\theta_4 = \langle\ \rangle$. Como $W_c \rightarrow W_{c''}\theta_4$, pode-se afirmar que W_c é uma fórmula contraditória.

6.4.2 Os ciclos de inferência

Viu-se, então, que a noção de eliminação de literais se mantém mesmo para fórmulas de primeira ordem quando expressas na representação cruzada. Mantém-se também a noção de refutação destas fórmulas com a identificação de uma substituição cuja aplicação torna todas as cláusulas duais, simultaneamente, explicitamente contraditórias. Estas noções são (praticamente) as mesmas requeridas para a inferência na lógica proposicional. No entanto nesta, uma vez verificada alguma cláusula dual não contraditória, pode-se garantir a satisfazibilidade da fórmula. O mesmo somente é válido para fórmulas de primeira ordem se for verificada a existência de pelo menos uma cláusula dual não contraditória (e nem potencialmente contraditória), isto é, cláusulas que não contêm nenhum par de literais complementares. Porém, considerando-se fórmulas cujas cláusulas duais potencialmente contraditórias, a inexistência de uma substituição que as torne simultaneamente contraditórias não indica a insatisfazibilidade da fórmula. Muitas vezes, para o cálculo desta substituição, seria necessário ou mais de uma ocorrência de determinadas cláusulas da fórmula ou o cálculo de mais de um *ciclo de inferência*.

As duas formas são equivalentes e podem ser utilizadas para a refutação de qualquer fórmula contraditória. A utilização de mais uma instância de determinadas cláusulas da fórmula permite que sejam calculadas instâncias diferentes de um mesmo literal no mesmo ciclo de inferência. Já, na inferência em ciclos, a refutação dá-se pela sucessão de cálculo, e união na fórmula original, de várias instâncias de mesmo literal, porém uma de cada vez. Cada forma tem vantagens e desvantagens. A primeira é normalmente adotada no Método das Conexões e tem a vantagem de evitar a geração explícita de teoremas intermediários, preocupando-se em calcular substituições. A segunda é normalmente adotada no Método da Resolução e tem a vantagem de facilmente descartar teoremas (gerados ou oriundos da fórmula) subsumidos, contendo, muitas vezes, o crescimento excessivo da fórmula durante o processo de refutação.

Normalmente fórmulas que exemplificam o que foi dito são aquelas que representam teorias cuja definição envolve recursividade. Uma fórmula, muito simples, deste tipo é apresentada a seguir:

Exemplo 31 *Uma fórmula contraditória com recursividade:*

$$\begin{aligned} W_c = \langle \quad & [\neg P(x_1), P(f(x_1))] \\ & [P(a)] \\ & [\neg P(f(f(a)))] \ \rangle \end{aligned}$$

Pelo Método das Conexões, uma possível refutação desta fórmula é:

- do par $(P(a), \neg P(x_{1_1}))$[5] tem-se a substituição $\theta_1 = \{x_{1_1}/a\}$,
- então, aplicando-se θ_1 em $P(f(x_{1_1}))$, tem-se $P(f(a))$,
- do par $(P(f(a)), \neg P(x_{1_2}))$ tem-se $\theta_2 = \{x_{1_2}/f(a)\}$,
- então, aplicando-se θ_2 em $P(f(x_{1_2}))$, tem-se $P(f(f(a))$,
- e finalmente, pelo par $(P(f(f(a)), \neg P(f(f(a))))$ tem-se a refutação da fórmula.

E, pelo Método da Resolução, uma possível refutação desta fórmula é:

- da resolução das cláusulas que contêm o par $(P(a), \neg P(x_1))$ tem-se o teorema $[P(f(a))]$,
- o teorema gerado é adicionado à fórmula, $W_{c'} = W_c \cup \{[P(f(a))]\}$, e um novo ciclo de inferência com $W_{c'}$ é iniciado,
- da resolução das cláusulas que contêm o par $(P(f(a)), \neg P(x_1))$ tem-se o teorema $[P(f(f(a)))]$,
- o teorema gerado é adicionado à fórmula, $W_{c''} = W_{c'} \cup \{[P(f(f(a)))]\}$, e um novo ciclo de inferência com $W_{c''}$ é iniciado,
- da resolução das cláusulas que contêm o par $(P(f(f(a))), \neg P(f(f(a))))$ tem-se, finalmente, a refutação da fórmula.

Para proceder-se a refutação desta fórmula baseados nas coordenadas, deve-se considerar a sua representação cruzada:

[5] Adicionou-se um índice indicando a instância do literal, e portanto da variável, considerada.

Exemplo 32 *A fórmula do exemplo 31 na representação cruzada:*

$$
\begin{array}{rclll}
W_c & = & \langle & 1 & [\neg P(x_1)^{\{1\}}, P(f(x_1))^{\{2\}}] \\
 & & & 2 & [P(a)^{\{1,2\}}] \\
 & & & 3 & [\neg P(f(f(a)))^{\{1,2\}}] \ \rangle \\
W_d & = & [& 1 & \langle \neg P(x_1)^{\{1\}}, P(a)^{\{2\}}, \neg P(f(f(a)))^{\{3\}} \rangle \\
 & & & 2 & \langle P(f(x_1))^{\{1\}}, P(a)^{\{2\}}, \neg P(f(f(a)))^{\{3\}} \rangle \ ]
\end{array}
$$

Nesta fórmula vê-se que a primeira cláusula dual torna-se contraditória com a substituição $\theta_1 = \{x_1/a\}$ e a segunda por $\theta_2 = \{x_1/f(a)\}$. Como não são compatíveis, θ_1 e θ_2 não podem ser combinadas. Assim, nenhuma substituição torna todas as cláusulas duais explicitamente contraditórias simultaneamente. Apesar disto pode-se, como apresentou-se anteriormente com os demais métodos, proceder a refutação em ciclos. Uma possível refutação baseada nas coordenadas para esta fórmula é a seguinte:

- pelo par $(P(a)^{\{1,2\}}, \neg P(x_1))^{\{1\}}$, vê-se que o literal $\neg P(x_1)^{\{1\}}$ pode-ser eliminado com $\theta_1 = \{x_1/a\}$, o que dá origem ao teorema $[P(f(a))]$,
- o teorema gerado é adicionado à W_c e a representação da fórmula $W_{c'}$ resultante é calculada

$$
\begin{array}{rclll}
W_{c'} & = & \langle & 1 & [\neg P(x_1)^{\{1\}}, P(f(x_1))^{\{2\}}] \\
 & & & 2 & [P(a)^{\{1,2\}}] \\
 & & & 3 & [\neg P(f(f(a)))^{\{1,2\}}] \\
 & & & 4 & [P(f(a))^{\{1,2\}}] \ \rangle \\
W_{d'} & = & [& 1 & \langle \neg P(x_1)^{\{1\}}, P(a)^{\{2\}}, \neg P(f(f(a)))^{\{3\}}, P(f(a))^{\{4\}} \rangle \\
 & & & 2 & \langle P(f(x_1))^{\{1\}}, P(a)^{\{2\}}, \neg P(f(f(a)))^{\{3\}}, P(f(a))^{\{4\}} \rangle \ ]
\end{array}
$$

  e um novo ciclo de inferência com $W_{c'}$ é iniciado,

- pelo par $(P(f(a)^{\{1,2\}}, \neg P(x_1))^{\{1\}}$, vê-se que o literal $\neg P(x_1)^{\{1\}}$ pode-ser eliminado com $\theta_2 = \{x_1/f(a)\}$, e
- pelo par $(P(f(x_1))^{\{2\}}, \neg P(f(f(a)))^{\{1,2\}})$, vê-se que o literal $P(f(x_1))^{\{2\}}$ também pode ser eliminado com θ_2,
- então, com a mesma substituição, todos os literais da primeira cláusula podem ser eliminados e todas as cláusulas duais tornam-se explicitamente contraditórias,

caracterizando a refutação da fórmula.

6.5 Eliminando caminhos

Pode-se agora resumir o que foi apresentado no conceito de *eliminação de caminhos* da forma conjuntiva de uma fórmula. Para tal pode-se considerar apenas a forma conjuntiva de uma fórmula na representação cruzada. Por exemplo, considerando-se a fórmula proposicional satisfazível e seus caminhos mostrada na figura 6.2, tem-se:

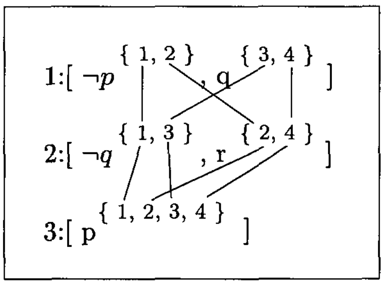

Figura 6.2: Uma fórmula satisfazível e seus caminhos.

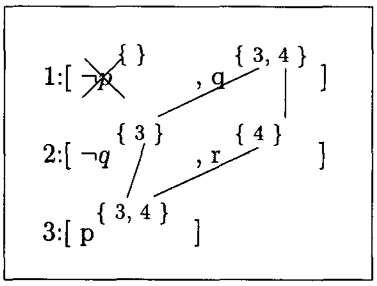

Figura 6.3: Eliminação dos caminhos contraditórios 1 e 2.

- os caminhos 1 e 2 são contraditórios porque contêm o par $(\neg p, p)$, o caminho 3 porque contém o par $(\neg q, q)$ e o caminho 4 não é contraditório;
- como um caminho não é contraditório e a fórmula é proposicional, a fórmula é satisfazível e a conjunção de literais do caminho, no caso $q \wedge r \wedge p$ é um modelo da fórmula;
- a retirada de uma cláusula dual coincide com a retirada de uma coordenada, ou de um caminho da representação. Por exemplo, na figura 6.3 é mostrada a retirada dos caminhos contraditórios 1 e 2. Assim, diz-se que um caminho em uma fórmula proposicional *é eliminado* se ele for contraditório;
- como a eliminação de caminhos coincide com a retirada de cláusulas duais, a eliminação de um literal coincide com a eliminação, de todos os caminhos que passam por este literal, pois se todos os caminhos dos quais um literal faz parte são contraditórios, este literal é irrelevante à prova (se a fórmula for satisfazível, não é por causa deste literal). Por exemplo, com a retirada dos caminhos 1 e 2, como mostrado na figura 6.3, nenhum caminho passa pelo literal $\neg p$ da primeira

cláusula, habilitando sua retirada da cláusula. Com sua retirada, pela r.e. 11, tem-se a inferência do teorema $[q]$;

- com a eliminação de todos os caminhos contraditórios, os literais não eliminados fazem parte de um modelo da fórmula. Na fórmula da figura 6.3, com a eliminação do caminho 3, o literal $\neg q$ da segunda cláusula é também eliminado, restando apenas os literais que contém a coordenada 4 em suas coordenadas fixas conjuntivas, gerando os teoremas $[q]$, $[r]$ e $[p]$ (ou $q \wedge r \wedge p$).

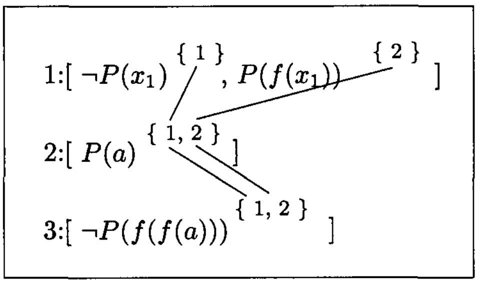

Figura 6.4: Uma fórmula de primeira ordem contraditória e seus caminhos.

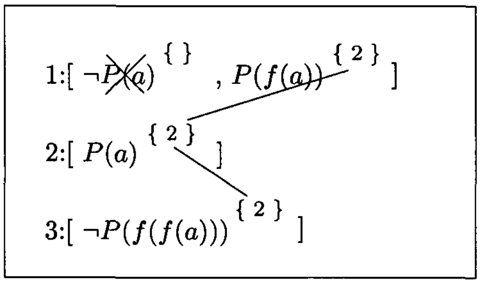

Figura 6.5: Eliminação do caminho 1 contraditório na instância $x_1 = a$.

Pode-se, da mesma forma, estender o conceito de eliminação de caminhos para fórmulas de primeira ordem. Considerando-se a fórmula de primeira ordem contraditória e seus caminhos, mostrada na figura 6.4 tem-se:

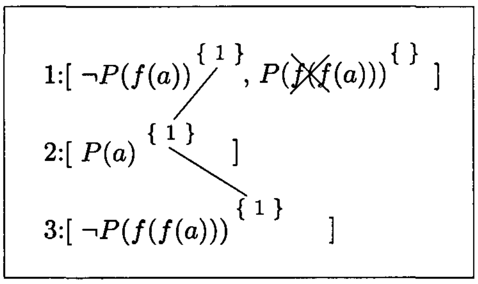

Figura 6.6: Eliminação do caminho 2 contraditório na instância $x_1 = f(a)$.

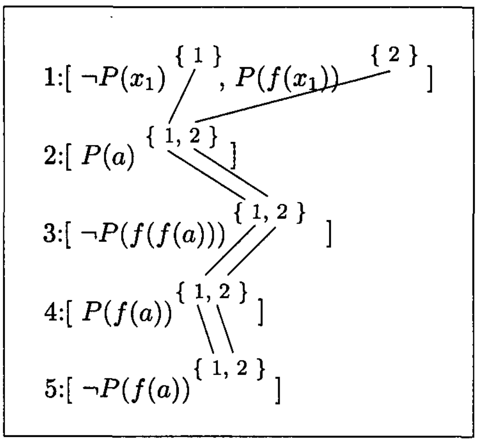

Figura 6.7: Fórmula original e teoremas gerados.

- o caminho 1 é contraditório com a substituição $\theta_1 = \{x_1/a\}$ e o caminho 2 é contraditório com a substituição $\theta_2 = \{x_1/f(a)\}$. Diz-se que um caminho em uma fórmula de primeira ordem *é eliminado* na instância θ se ele for contraditório nesta instância;

- como as substituições não são compatíveis, os caminhos não podem ser eliminados simultaneamente, isto é, não se pode considerar uma instância da fórmula na qual ambos caminhos são contraditórios. Pode-se, porém, considerar, paralelamente, as instâncias na qual cada caminho é contraditório;

- na instância onde $x = a$, conforme mostrado na figura 6.5, o caminho 1 é eliminado, eliminando o literal $\neg P(x_1)$ (que nesta instância é $\neg P(a)$) e dando origem, pela r.e. 11, ao teorema $[P(f(a))]$ (isto é, $P(f(x_1))$ nesta instância);

- já na instância onde $x = f(a)$, conforme mostrado na figura 6.6, o caminho 2 é eliminado, eliminando o literal $P(f(x_1))$ (que nesta instância é $P(f(f(a)))$) e dando origem, pela r.e. 11, ao teorema $[\neg P(f(a))]$ (isto é, $\neg P(x_1)$ nesta instância);

- a inclusão dos teoremas gerados na fórmula original dá origem a novo ciclo de inferência. Esta nova fórmula e seus caminhos são mostrados na figura 6.7. Considerando-se esta nova fórmula tem-se:

  - os caminhos 1 e 2 ainda são eliminados nas mesmas instâncias θ_1 e θ_2, respectivamente, e também o são na instância $\theta_3 = \{\}$ (que é a instância da fórmula mostrada na figura 6.7), proveniente do par $(P(f(a)), \neg P(f(a)))$;

  - como ambos caminhos são contraditórios na mesma instância ambos podem ser eliminados e, como estes são os únicos caminhos, esta eliminação é condição suficiente para a refutação da fórmula.

6.6 Conclusão

Resumindo, considerando a forma conjuntiva de uma fórmula proposicional ou de primeira ordem na representação cruzada, um caminho é *contraditório* se contém pelo menos um par de literais complementares. Um caminho contraditório é dito *eliminado* na instância da fórmula na qual é contraditório. A eliminação de um caminho dá-se pela retirada, da coordenada que o representa, das coordenadas fixas conjuntivas de todos os literais da forma conjuntiva da instância fórmula. Um caminho *passa* por um literal quando a coordenada que o representa faz parte das coordenadas fixas conjuntivas deste literal. Um literal é dito *eliminado* em uma instância da fórmula quando, nesta instância, nenhum caminho passa por ele. A cláusula resultante da remoção de literais eliminados em determinada instância *é teorema* da fórmula considerada. A eliminação de todos os literais de uma cláusula em uma determinada instância da fórmula considerada é condição suficiente para sua refutação.

A eliminação de caminhos pode ser vista como uma *regra de inferência* baseada na transformação dual e pode dar origem à diferentes estratégias para métodos de prova baseados nesta transformação. Um exemplo de estratégia baseada em eliminação de caminhos é visto no próximo capítulo.

"Well, in out country," said Alice, still panting a little, "you'd generally get to somewhere else if you ran very fast for a long time, as we've been doing." "A slow sort of country!" said the Queen. "Now, here, you see, it takes all the running you can do, to keep in the same place. If you want to get somewhere else, you must run at least twice as fast as that!"

Lewis Carroll, Through the Looking Glass, 1871

Capítulo 7

Acelerando o passo

O desenvolvimento de uma estratégia baseada em caminhos para fórmulas proposicionais não faz muito sentido pois, com a forma disjuntiva calculada, basta uma "varredura" em todas as cláusulas duais e, se (i) todas as cláusulas duais forem contraditórias, a fórmula é contraditória, senão (ii) cada cláusula dual não contraditória é um modelo para a fórmula. Pode-se dizer, então, que, no caso proposicional, a estratégia é o próprio algoritmo de transformação dual. Portanto, nosso escopo para falar de estratégias baseadas em eliminação de caminhos é a lógica de primeira ordem.

7.1 Estratégias baseadas em caminhos

Considerando-se os caminhos, refutar uma fórmula é demonstrar que existe uma instância desta fórmula na qual todos os caminhos são contraditórios. Desenvolver estratégias consiste em tentar encontrar esta instância de uma maneira eficiente. Por eficiente entende-se minimizando tanto o número de teoremas intermediários gerados como o número de ciclos necessários. Este conceito de eficiência é geral a todas estratégias para métodos de prova e sua realização não é trivial, uma vez que menos teoremas gerados em cada ciclo pode levar ao aumento do número de ciclos necessários à prova. Por outro lado, o excesso de teoremas gerados em cada ciclo pode impossibilitar, na prática, a obtenção da prova devido ao crescimento exponencial do espaço de busca.

Uma estratégia eficiente deve, então, preocupar-se em gerar a cada ciclo os "bons" teoremas, ou seja, aqueles que são necessários à prova. Porém, não se pode garantir a eficiência de uma estratégia. Normalmente tem-se estratégias eficientes para determinadas classes de problemas. Mesmo esta eficiência é normalmente comprovada empiricamente.

De uma forma geral, uma estratégia baseada em caminhos deve levar em conta os seguintes fatos:

- em uma mesma instância mais de um caminho pode ser contraditório, isto é, mais de um caminho de uma fórmula pode ser contraditório mediante a aplicação da mesma substituição, e
- mesmo que um mesmo caminho possa ser contraditório em mais de uma instância, para a refutação da fórmula é necessário encontrar-se apenas uma de suas instâncias na qual todos os caminhos sejam simultaneamente contraditórios,

os quais justificam estratégias nas quais:

- em cada instância calculada sejam eliminados, simultaneamente, todos os caminhos contraditórios nesta instância; e
- uma vez eliminado em uma instância, um caminho não seja mais considerado na busca pela refutação nesta instância.

Para exemplificar estes fatos considere a fórmula[1]:

Exemplo 33 *Uma fórmula contraditória:*

$$
\begin{array}{rcll}
W_c & = & \langle & 1 \quad [\neg P(x_1)^{\{1,2\}}, P(f(x_1))^{\{3,4\}}] \\
 & & & 2 \quad [\neg P(f(x_2))^{\{1,3\}}, Q(x_2)^{\{2,4\}}] \\
 & & & 3 \quad [P(f(a))^{\{1,2,3,4\}}] \\
 & & & 4 \quad [\neg Q(a)^{\{1,2,3,4\}}] \ \rangle
\end{array}
$$

Os caminhos desta fórmula e as instâncias nas quais estes caminhos são contraditórios são mostrados na tabela 7.1. Nesta vê-se, por exemplo, que o caminho 3, por conter mais de um par complementar, é eliminado nas instâncias θ_2 e θ_3. Neste caso, apesar de serem compatíveis, estas instâncias não precisam ser consideradas simultaneamente na busca da refutação. Vê-se também que na instância θ_2, os caminhos 1 e 3 são eliminados, portanto, ao considerar esta instância pode-se desconsiderar outras instâncias nas quais estes caminhos são eliminados. Além disto, vê-se que as instâncias

[1] Adotou-se, neste capítulo, apresentar apenas a forma conjuntiva das fórmulas. Considera-se, porém, que as mesmas encontram-se expressas na representação cruzada (isto é, as coordenadas dos literais da forma conjuntiva referem-se às cláusulas duais da forma disjuntiva.)

par complementar	*caminho*	*instância*
$(P(f(a))^{\{1,2,3,4\}}, \neg P(x_1)^{\{1,2\}})$	1,2	$\theta_1 = \{x_1/f(a)\}$
$(P(f(a))^{\{1,2,3,4\}}, \neg P(f(x_2))^{\{1,3\}})$	1,3	$\theta_2 = \{x_2/a\}$
$(P(f(x_1))^{\{3,4\}}, \neg P(f(x_2))^{\{1,3\}})$	3	$\theta_3 = \{x_1/x_2\}$
$(Q(x_2)^{\{2,4\}}, \neg Q(a)^{(\{1,2,3,4\}})$	2,4	$\theta_4 = \{x_2/a\}$

Tabela 7.1: Pares complementares, caminhos eliminados e instância desta eliminação associados à fórmula do exemplo 33.

θ_2 e θ_4 são a mesma instância $\{x_2/a\}$, portanto nesta instância são eliminados os caminhos 1, 3, 2 e 4. Por fim, como nesta instância todos os caminhos são eliminados, a fórmula está refutada, conforme mostra a figura 7.1.

1 [$\neg P(x_1)^{\{1,2\}}$, $P(f(x_1))^{\{3,4\}}$]

2 [$\neg P(f(a))^{\{1,3\}}$, $Q(a)^{\{2,4\}}$]

$\{x_2/a\}$, eliminando 1,3

3 [$P(f(a))^{\{1,2,3,4\}}$]

$\{x_2/a\}$, eliminando 2,4

4 [$\neg Q(a)^{\{1,2,3,4\}}$]

Figura 7.1: A refutação da fórmula do exemplo 33 na instância $x_2 = a$.

As diferentes formas de busca pelas instâncias nas quais os caminhos são eliminados definem estratégias baseadas em caminhos.

7.2 A estratégia Goal

Esta estratégia é inspirada tanto na *redução por objetivo* ("goal reduction" [49]), uma técnica bastante utilizada pra solução de problemas em diversas áreas da Inteligência Artificial, como na estratégia *conjunto de suporte* ("set of support" [11]) uma das estratégias mais eficientes para métodos baseados na resolução. A idéia central da redução por objetivo é a solução de um problema (objetivo) através da solução dos subproblemas que o compõem.

O conjunto de suporte, por sua vez, considera que, em prova por refutação normalmente tem-se uma fórmula sabidamente satisfazível e a negação de um teorema a ser refutada. Como a fórmula sem a negação do teorema é satisfazível, é esta negação que a torna contraditória e portanto, toda a resolução efetuada utiliza a negação do teorema ou um de seus resolventes. Com isto, tem-se sempre a geração de resolventes de instâncias compatíveis com a negação do teorema. A idéia central é: como acredita-se que o teorema é uma conseqüência lógica da fórmula dada e que a fórmula dada é satisfazível, acredita-se ser possível inferi-lo da mesma. Portanto, existe uma instância da fórmula na qual o teorema é inferido e a negação do teorema é "explicitamente contraditória" (isto é, complementar ao teorema inferido com substituição vazia). E esta instância é, certamente, uma instância onde os literais complementares à negação do teorema unificam com esta com substituição vazia.

7.2.1 A idéia

Considerando-se a eliminação de literais, sabe-se que a eliminação de todos os literais de uma cláusula de uma fórmula constitui sua refutação. Unindo-se estes enfoques, pode-se considerar como "objetivo" a eliminação dos literais da negação do teorema considerando apenas as instâncias nas quais estes literais são explicitamente contraditórios. Para tal considere a seguinte fórmula:

Exemplo 34 *Uma fórmula contraditória:*

$$\begin{array}{rcll} W_c & = & \langle & 1 \quad [\neg Equal(x_1, x_2)^{\{1,2,3\}}, Equal(x_2, x_1)^{\{4,5,6\}}] \\ & & & 2 \quad [\neg Equal(x_3, x_4)^{\{1,4\}}, \neg Equal(x_4, x_5)^{\{2,5\}}, Equal(x_3, x_5)^{\{3,6\}}] \\ & & & 3 \quad [Equal(f(x_6), f(f(x_6)))^{\{1,2,3,4,5,6\}}] \\ & & & 4 \quad [\neg Equal(f(f(f(a))), f(a))^{\{1,2,3,4,5,6\}}] \ \rangle \end{array}$$

na qual o teorema negado é a cláusula de número 4 e o restante da fórmula (cláusulas 1 a 3) é satisfazível. Para a refutação do teorema, quer-se encontrar uma instância da fórmula onde a negação do teorema seja explicitamente contraditória. Esta instância tem de ser uma instância onde $Equal(x_2, x_1)$ ou $Equal(x_3, x_5)$ são (pelos valores das variáveis) $Equal(f(f(f(a))), f(a))$, uma vez que estes são os únicos literais complementares à negação do teorema.

Considerando-se a primeira possibilidade tem-se que $x_2 = f(f(f(a)))$ e $x_1 = f(a)$. Então nesta instância, $\neg Equal(x_1, x_2)$ é $\neg Equal(f(a), f(f(f(a))))$ e, sendo a fórmula contraditória, também existe nesta instância um literal explicitamente complementar

par complementar	*caminhos*
$Equal(x_2, x_1)^{\{4,5,6\}}, \neg Equal(x_3, x_4)^{\{1,4\}}$	4
$Equal(x_2, x_1)^{\{4,5,6\}}, \neg Equal(x_4, x_5)^{\{2,5\}}$	5
$Equal(x_2, x_1)^{\{4,5,6\}}, \neg Equal(f(f(f(a))), f(a))^{\{1,2,3,4,5,6\}}$	4,5,6
$Equal(x_3, x_5)^{\{3,6\}}, \neg Equal(x_1, x_2)^{\{1,2,3\}}$	3
$Equal(x_3, x_5)^{\{3,6\}}, \neg Equal(f(f(f(a))), f(a))^{\{1,2,3,4,5,6\}}$	3,6
$Equal(f(x_6), f(f(x_6)))^{\{1,2,3,4,5,6\}}, \neg Equal(x_1, x_2)^{\{1,2,3\}}$	1,2,3
$Equal(f(x_6), f(f(x_6)))^{\{1,2,3,4,5,6\}}, \neg Equal(x_3, x_4)^{\{1,4\}}$	1,4
$Equal(f(x_6), f(f(x_6)))^{\{1,2,3,4,5,6\}}, \neg Equal(x_4, x_5)^{\{2,5\}}$	2,5

Tabela 7.2: Pares complementares e caminhos contraditórios associados à fórmula do exemplo 34.

à $\neg Equal(f(a), f(f(f(a))))$. Deve-se agora procurar qual pode ser este literal. Esta busca continua até que a refutação seja encontrada (até que todos os caminhos tenham sido eliminados) ou até que não haja mais nenhuma possibilidade a ser considerada neste ciclo e deve-se reiniciar a busca em ciclos subseqüentes.

Dada a idéia geral, pode-se explicar a estratégia "mais tecnicamente". Nesta estratégia o objetivo inicial é tentar eliminar os literais da negação do teorema, eliminando os caminhos que por eles passam. No exemplo, o objetivo inicial é eliminar o literal $\neg Equal(f(f(f(a))), f(a))$ eliminando os caminhos 1, 2, 3, 4, 5 e 6.

Na tabela 7.2 estão representados todos os caminhos potencialmente contraditórios da fórmula deste exemplo. Nela vê-se que muitos são os caminhos contraditórios e muitas instâncias podem ser calculadas (e, por conseguinte, muitos são os teoremas que podem ser gerados). Quer-se, porém, encontrar pelo menos uma das instâncias nas quais a negação do teorema é explicitamente contraditória.

Pelos literais complementares ao teorema, mostrados na tabela, vê-se que o literal da negação do teorema é explicitamente complementar ao literal $Equal(x_2, x_1)$ da cláusula 1 nas instâncias onde $x_2 = f(f(f(a)))$ e $x_1 = f(a)$. Nestas instâncias, então, a negação do teorema é contraditória. No entanto, como nestas instâncias apenas alguns caminhos (4, 5 e 6) são contraditórios, este par complementar não é suficiente à refutação. Deve-se então procurar eliminar os demais caminhos com instâncias compatíveis à estas.

Como os demais caminhos não eliminados são os caminhos que passam pelos demais literais da cláusula 1, o objetivo passa a ser eliminar estes literais com instâncias compatíveis. No caso, o objetivo passa a ser eliminar o literal $\neg Equal(x_1, x_2)$. Procede-se para este objetivo da mesmo forma que para o objetivo inicial criando novos objetivos se necessário. Toma-se, no entanto, o cuidado de respeitar as restrições impostas pelo

$\neg Equal(f(f(f(a))), f(a))^{\{1,2,3,4,5,6\}}$ gap: 1,2,3,4,5,6

$\|$ elimina 4,5,6

$Equal(x_2, x_1)^{\{4,5,6\}}$

$|$

$\neg Equal(x_1, x_2)^{\{1,2,3\}}$ gap: 1,2,3

$\|$ elimina 3

$Equal(x_3, x_5)^{\{3,6\}}$

$\neg Equal(x_3, x_4)^{\{1,4\}}$ gap: 1	$\neg Equal(x_4, x_5)^{\{2,5\}}$ gap: 2
$\|$ elimina 1,4	$\|$ elimina 2,5
$Equal(f(x_6), f(f(x_6)))^{\{1,2,3,4,5,6\}}$	$Equal(f(x_6), f(f(x_6)))^{\{1,2,3,4,5,6\}}$

Figura 7.2: Árvore parcial de objetivos para a fórmula do exemplo 34 sem a aplicação das substituições envolvidas.

objetivo superior: (i) a substituição já considerada (no caso $\{x_2/f(f(f(a)), x_1/f(a)\}$) e (ii) os caminhos já eliminados (no caso 4, 5 e 6). Isto feito obtém-se uma *árvore de objetivos*, conforme mostrado na figura 7.2.

A representação de objetivos e subobjetivos em árvore é comum na técnica de redução de objetivos. Esta árvore é chamada de *árvore e/ou* ("and/or trees") pois através dela representa-se objetivos que devem ser resolvidos simultaneamente – ramos "e", ou independentemente, isto é, opções de solução para um mesmo objetivo – ramos "ou". Nossa representação é uma adaptação desta árvore. Os caminhos indicados por *gap* indicam os caminhos ainda não eliminados em ramos superiores. Ramos duplos indicam soluções (literais complementares) consideradas e, se mais de uma ocorrer para um mesmo objetivo, estas são independentes (isto é são ramos "ou"). Ramos simples indicam subobjetivos necessários à realização do objetivo. Se mais de um ocorrer para uma mesma solução, estes são ramos "e" e aparecem unidos por um arco. Neste caso os objetivos destes ramos são *dependentes* e devem ser solucionados simultaneamente para o objetivo superior ser considerado resolvido.

Na figura 7.3 tem-se a mesma árvore de objetivos da figura 7.2 porém com as substituições aplicadas (como de fato deve ocorrer). É interessante observar nesta figura que os objetivos dependentes podem ser resolvidos simultaneamente. Nestes, as substituições envolvidas são $\{x_6/a, x_4/f(f(a))\}$ e $\{x_6/f(a), x_4/f(f(a))\}$ que são, à princípio incompatíveis. Porém, se observarmos as variáveis dos objetivos, vemos que apenas x_4 faz parte do mesmo e o valor atribuído à esta variável, em ambas substituições é o

$$\neg Equal(f(f(f(a))),f(a))^{\{1,2,3,4,5,6\}}$$
$$\| \quad \{x_2/f(f(f(a))), x_1/f(a)\}$$
$$Equal(f(f(f(a))),f(a))^{\{4,5,6\}}$$
$$|$$
$$\neg Equal(f(a),f(f(f(a))))^{\{1,2,3\}}$$
$$\| \quad \{x_3/f(a), x_5/f(f(f(a)))\}$$
$$Equal(f(a),f(f(f(a))))^{\{3,6\}}$$

$$\neg Equal(f(a),x_4)^{\{1,4\}} \qquad \neg Equal(x_4,f(f(f(a))))^{\{2,5\}}$$
$$\| \; \{x_6/a, x_4/f(f(a))\} \qquad \| \; \{x_6/f(a), x_4/f(f(a))\}$$
$$Equal(f(a),f(f(a)))^{\{1,2,3,4,5,6\}} \qquad Equal(f(f(a)),f(f(f(a))))^{\{1,2,3,4,5,6\}}$$

Figura 7.3: Árvore parcial de objetivos para a fórmula do exemplo 34 com a aplicação das substituições envolvidas.

mesmo. Portanto as substituições envolvidas nas soluções, *relativas às (ou projetadas nas)* variáveis dos objetivos são compatíveis e podem ser aplicadas simultaneamente. Assim, como todos os caminhos podem ser eliminados simultaneamente a fórmula está refutada.

Apesar de já termos apresentado a refutação da fórmula, na tabela 7.2 vê-se que o literal da negação do teorema é também complementar ao literal $Equal(x_3, x_5)$ da cláusula 2. Esta é uma outra possibilidade de cumprir o objetivo inicial e, a princípio, seria também considerada pela estratégia. Seguindo o mesmo raciocínio tem-se a árvore de objetivos mostrada nas figuras 7.4 e 7.5. É interessante observar que nesta árvore existe a solução $\{x_4/f(f(f(f(a))))\}$ para o literal $\neg Equal(X_3, x_4)$ (sublinhada na figura 7.5) que não pode ser considerada por ser incompatível com a solução $\{x_4/f(f(a))\}$ do literal $\neg Equal(x_4, x_5)$.

Existem também fórmulas para as quais a refutação não ocorre em um único ciclo. Para exemplificar, considere a seguinte fórmula apresentada no capítulo anterior:

Exemplo 35 *A fórmula do exemplo 31:*

$$\begin{array}{rcll} W_c & = & \langle & 1 \quad [\neg P(x_1)^{\{1\}}, P(f(x_1))^{\{2\}}] \\ & & & 2 \quad [P(a)^{\{1,2\}}] \\ & & & 3 \quad [\neg P(f(f(a)))^{\{1,2\}}] \ \rangle \end{array}$$

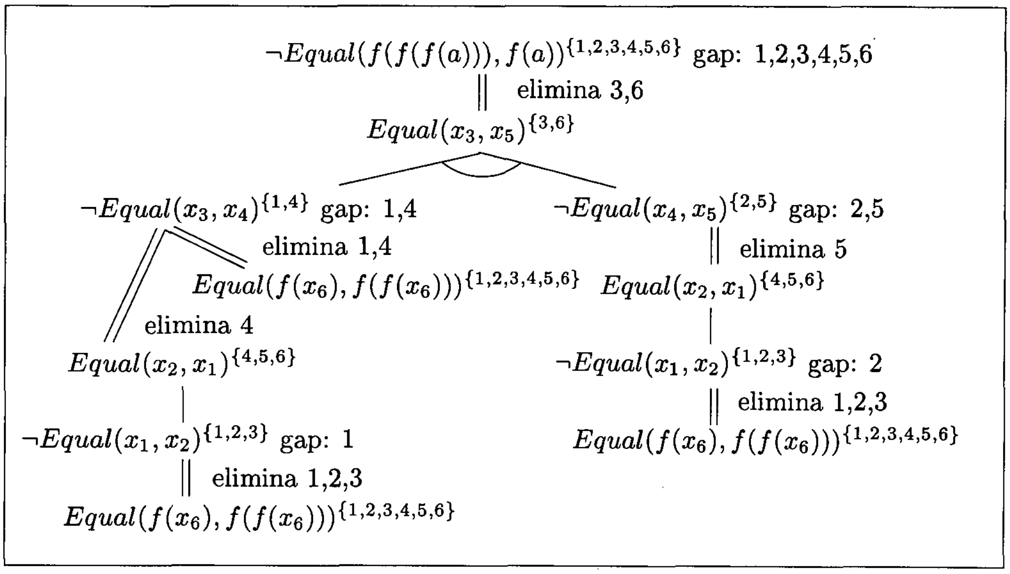

Figura 7.4: Árvore parcial de objetivos para a fórmula do exemplo 34 sem a aplicação das substituições envolvidas.

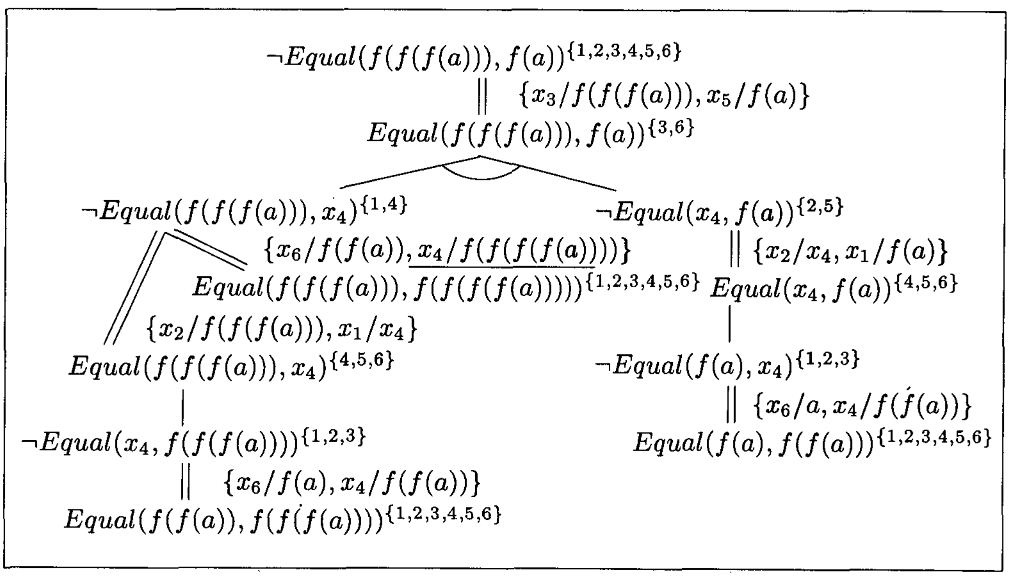

Figura 7.5: Árvore parcial de objetivos para a fórmula do exemplo 34 com a aplicação das substituições envolvidas.

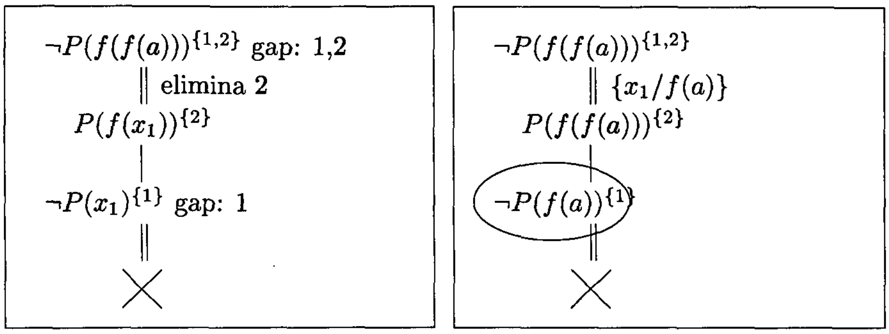

Figura 7.6: Árvore de objetivos do primeiro ciclo para a fórmula do exemplo 35.

A árvore de objetivos para esta fórmula é mostrada na figura 7.6. Nela vê-se que o objetivo $\neg P(f(a))$ não pode ser satisfeito pois nenhum literal complementar da fórmula contribui para a eliminação do *gap*. Neste caso, este literal é teorema gerado e pode ser adicionado à fórmula. Sabe-se que este literal é teorema porque nos ramos superiores da árvore todos os caminhos dos demais literais de mesma cláusula (isto é, o caminho 2) foram eliminados e, portanto, na instância onde $x_1 = f(a)$ o literal $\neg P(x_1)$ é o único literal não eliminado.

Os teoremas gerados representam objetivos não solucionados. São também teoremas inferidos da negação do teorema, isto é, não são teoremas da fórmula sem a negação do teorema. Então, estes teoremas gerados são *potencialmente*[2] a negação de teoremas da parte satisfazível da fórmula, pois considerando-se que a fórmula estaria refutada se estes literais pudessem ter sido eliminados. Assim, pode-se usá-los como objetivos iniciais do próximo ciclo de inferência. Isto feito, inicia-se outro ciclo de inferência com a fórmula:

[2]Depende da prova de $(W_c - T \rightarrow \neg\delta)$, isto é, que a negação do teorema gerado δ é conseqüência lógica da parte satisfazível da fórmula $W_c - T$, pois se isto for verdade, então, $W_c - T \cup \{\delta\}$ é insatisfazível e os teoremas negados contidos em T considerados em um ciclo, poderiam ser desconsiderados em ciclos subseqüentes, o que acarretaria uma redução do espaço de busca considerável no decorrer dos ciclos de inferência. Além disto, se cada teorema gerado é, por si, a negação de teoremas da parte satisfazível da fórmula, então apenas um destes teoremas é necessário à refutação ou, pelo menos, poder-se-ia considerar apenas destes como objetivo inicial. Mas, como não se tem esta prova até o momento, para garantir completude esta estratégia mantém T em ciclos subseqüentes.

$$\begin{array}{l} \neg P(f(a))^{\{1,2\}} \text{ gap: } 1,2 \\ \quad \Vert \text{ elimina } 2 \\ P(f(x_1))^{\{2\}} \\ \quad | \\ \neg P(x_1)^{\{1\}} \text{ gap: } 1 \\ \quad \Vert \text{ elimina } 1 \\ P(a)^{\{1,2\}} \end{array} \qquad \begin{array}{l} \neg P(f(a))^{\{1,2\}} \\ \quad \Vert \ \{x_1/a\} \\ P(f(a))^{\{2\}} \\ \quad | \\ \neg P(a)^{\{1\}} \\ \quad \Vert \ \{\} \\ P(a)^{\{1,2\}} \end{array}$$

Figura 7.7: Árvore de objetivos do segundo ciclo para a fórmula do exemplo 35.

$$W_c = \langle \begin{array}{ll} 1 & [\neg P(x_1)^{\{1\}}, P(f(x_1))^{\{2\}}] \\ 2 & [P(a)^{\{1,2\}}] \\ 3 & [\neg P(f(f(a)))^{\{1,2\}}] \\ 4 & [\neg P(f(a))^{\{1,2\}}] \ \rangle \end{array}$$

no qual a refutação é alcançada conforme mostrado na figura 7.7.

Por fim, como alguns importantes aspectos não foram salientados nos exemplos, acrescenta-se que:

- a disjunção de objetivos dependentes não solucionados é teorema da fórmula uma vez que eles são obtidos da não eliminação de literais de mesma cláusula (isto é, literais disjuntos);
- cláusulas já usadas por ramos superiores não são reutilizadas; e
- uma cláusula pode ser reutilizada em ramos paralelos (ramos com diferentes ramos superiores). Neste caso, quando utilizadas para resolver objetivos dependentes, suas variáveis devem ser renomeadas para evitar incompatibilidades de substituições. Na realidade, cada ocorrência de uma mesma cláusula é independente, não é a mesma cláusula. É como se a cláusula ocorresse várias vezes na fórmula.

7.2.2 O algoritmo

A estratégia apresentada pode ser resumida na rotina *Goal*:

Goal$(W_c, Goals)$

1. Change variable names
2. $new \leftarrow \emptyset$
3. **for all** $C \in Goals$ **do**

   $\Delta \leftarrow W_c - \{C\}$

   $(solutions, theorems) \leftarrow \sum_{l \in C} Solutions(l, F_c(l), \Delta, \{\})$

   $(solutions, theorems) \leftarrow \bigotimes(solutions, theorems, C)$

   if $solutions \neq \emptyset$

   then Return (W_c is contradictory)

   else $new \leftarrow new \cup theorems$
4. $NewW_c \leftarrow \|\lfloor W_c \cup new \rfloor\|$
5. **if** $NewW_c = W_c$

   then Return (W_c is closed)

   else $Goal(NewW_c, new \cap NewW_c)$

a qual recebe, como parâmetros de entrada, uma fórmula (W_c) na representação cruzada e um subconjunto desta fórmula ($Goal \subset W_c$) com as cláusulas que formam a negação do teorema a ser provado. Caso a fórmula seja contraditória, em algum ciclo a rotina encontra a refutação e encerra. Os passos desta rotina podem ser assim explicados:

1. troca do nome das variáveis da fórmula para garantir que variáveis diferentes tenham nomes diferentes;

2. inicialização da variável que acumula cláusulas inferidas no ciclo em processo;

3. para cada cláusula $C \in Goals$:

   - coloca em Δ o espaço de busca da solução ($W_c - \{C\}$);

   - para cada $l \in C$: a eliminação de l é buscada através da rotina *Solutions*, considerando os caminhos a serem eliminados ($F_c(l)$) e o espaço de busca. São retornados dois conjuntos: as soluções encontradas (substituições) e os teoremas gerados (lista de literais não eliminados na busca). Estes conjuntos retornados são colecionados ($\sum$) nas variáveis *solutions* e *theorems* respectivamente.

   - as soluções e teoremas de cada literal da cláusula C são combinadas através da operação $\bigotimes$. Se existe uma solução retornada por esta operação a fórmula está refutada, caso contrário os teoremas retornados são acumulados na variável *new* (variável que acumula os teoremas gerados no ciclo de inferência);

4. o conjunto de novos teoremas é adicionado à fórmula original e a nova fórmula, $NewW_c$ é condensada ($\lfloor \ldots \rfloor$) e a representação cruzada recalculada ($\| \ldots \|$);

5. se nenhum novo teorema foi gerado no ciclo ($NewW_c = W_c$) então a fórmula é dita *fechada* e a busca é encerrada. Caso contrário, a rotina *Goal* é chamada recursivamente com a nova fórmula $NewW_c$ e as cláusulas desta fórmula advindas dos novos teoremas, dando origem a um novo ciclo de inferências.

Durante a execução da rotina *Goal*, no passo 3, para cada literal de uma cláusula do teorema a ser provado é executada a rotina *Solutions*. Esta rotina é responsável pelo cálculo das soluções possíveis para a eliminação de um literal dado, e pode representada como segue:

Solutions$(Lit, Gap, W_c, Prev)$
1. $solutions \leftarrow Previous(Lit, Gap, Prev)$
2. $\Omega \leftarrow Complementaries(Lit, W_c, Gap)$
3. **if** $\Omega = \emptyset$
   then $theorems \leftarrow \{(\{Lit\}, \{\})\}$
   else $(solutions, theorems) \leftarrow \bigcup_{(neg,\theta)\in\Omega} Subgoals(Lit, Gap, neg, \theta, W_c,$
   $Prev)$
4. **if** $solutions \neq \emptyset$
   then $solutions \leftarrow Proj(solutions, Lit \cup Prev)$
5. Return $(solutions, theorems)$

a qual recebe como parâmetros de entrada o literal propriamente dito, os caminhos deste literal ainda não eliminados por ramos superiores da árvore de objetivos, o subconjunto de W_c na qual a solução deve ser buscada e a lista de objetivos prévios (ancestrais). A rotina calcula as soluções possíveis (substituições) para eliminar o literal e retorna estas soluções e, como teoremas, os literais não eliminados durante a busca das soluções. Os passos desta rotina podem ser assim explicados:

1. primeiramente é verificado, através da rotina *Previous*, se o literal do objetivo atual *Lit* contradiz algum dos literais dos objetivos prévios (contidos no parâmetro de entrada *Prev*) com substituição θ. Neste caso, na instância θ, os caminhos da intersecção destes dois literais são caminhos contraditórios. Como *Gap* vêm da intersecção dos caminhos de objetivos prévios com os caminhos do objetivo atual, considerando-se a instância θ tem-se $Gap = \emptyset$. Assim, esta contradição é uma solução para *Lit* e é armazenada na variável *solutions*;

2. considerando-se os parâmetros de entrada, é calculado Ω, o conjunto de todos os pares (neg, θ) tal que: neg é um literal de W_c, é complementar ao literal Lit com substituição θ e os caminhos eliminados por este par complementar de literais está em Gap;

3. se Ω é vazio, o literal Lit não possui nenhuma nova solução e é, portanto, um teorema a ser retornado (associado à substituição vazia, detalhe importante para a operação $\bigotimes$). Senão, para cada par de Ω a solução é calculada através da rotina *Subgoals* que também retorna, por sua vez, soluções e teoremas gerados durante a busca. Estes resultados são acumulados (unidos) nas variáveis *solutions* e *theorems* respectivamente;

4. após considerar todos os pares de Ω para a eliminação de Lit, as substituições contidas em cada solução são projetadas nas variáveis de Lit e de $Prev$ (pois são as variáveis conhecidas neste ramo da árvore de objetivos);

5. a rotina retorna as soluções e os teoremas armazenados nas variáveis *solutions* e *theorems* respectivamente.

Por sua vez, durante a execução da rotina *Solutions*, no passo 2, é executada a rotina *Subgoals*. Esta rotina é responsável pelo cálculo das soluções para a eliminação de um literal dado um literal complementar. Para tal subobjetivos são solucionados se necessário. Esta rotina pode ser representada como segue:

Subgoals$(Lit, Gap, neg, \theta, W_c, Prev)$
1. $Gap \leftarrow Gap - (F_c(Lit) \cap F_c(neg))$
2. **if** $Gap = \emptyset$
    then $solutions \leftarrow \{\theta\}$
    else $L \leftarrow C(neg) - \{neg\}, \Delta \leftarrow [W_c - \{C\}], P \leftarrow Prev \cup \{Lit\}$
      $(solutions, theorems) \leftarrow \sum_{l \in L} Solutions(l\theta, Gap \cap F_c(l), \Delta, P)$
      $(solutions, theorems) \leftarrow \bigotimes(solutions, theorems, L)$
      if $solutions \neq \emptyset$
        then $solutions \leftarrow Proj(Compatible(solutions, \theta), Lit \cup Prev)$
        else $theorems \leftarrow Compatible(theorems, \theta)$
3. Return $(solutions, theorems)$

a qual recebe como parâmetros de entrada o literal a ser eliminado, os caminhos (deste literal) ainda não eliminados, um literal complementar, uma substituição tal que $Lit\theta = \neg neg\theta$, o espaço de busca (no caso de ser necessária a solução de subobjetivos) e a lista de objetivos prévios. A rotina verifica, primeiramente, se o par de literais complementares com a substituição θ elimina todos os caminhos a serem eliminados. Se for o caso, a substituição θ é a solução. Caso contrário, é buscada a eliminação de cada literal da cláusula do literal complementar através da rotina *Solutions*. Como estes são objetivos dependentes, as soluções e teoremas encontrados devem ser combinados para compor a solução do objetivo *Lit*. Os passos desta rotina podem ser assim explicados:

1. os caminhos eliminados pelo par complementar (Lit, neg) são retirados da lista de caminhos a serem eliminados (Gap);

2. se todos os caminhos foram eliminados, então a substituição θ é, por si, uma solução para a eliminação de *Lit*. Senão, a eliminação de *Lit* depende da eliminação dos caminhos que ainda restam ser eliminados, contidos nos demais literais da cláusula do literal complementar *neg*. Para tal:

   - as variáveis L,Δ e P recebem a lista dos literais da cláusula do literal *neg* exceto *neg*, o espaço de busca para os subobjetivos dependentes, com as variáveis renomeadas ($\lceil \ldots \rceil$) e a lista de objetivos prévios acrescida de *Lit*;

   - para cada literal l de L são calculadas as soluções para a instância θ de l, que eliminem os caminhos de *Gap* contidos em l, no espaço de busca indicado em Δ. As soluções e teoremas retornadas são acumulados nas variáveis *solutions* e *theorems*, respectivamente;

   - como as soluções retornadas advêm de objetivos dependentes, elas devem ser combinadas entre si através da operação $\bigotimes$;

   - se existem soluções para todos os literais de L e compatíveis com a substituição da contradição θ, a projeção da combinação de θ com cada uma destas soluções, nas variáveis de *Lit* e *Prev*, é acumulada na variável *solutions*. Se existem teoremas retornados na variável *theorems*, aqueles cuja substituição associada é compatível com a substituição da contradição θ, são mantidos com a substituição θ combinada como nova substituição associada. Os demais são teoremas descartados.

3. a rotina retorna as soluções e os teoremas armazenados nas variáveis *solutions* e *theorems* respectivamente.

Nas rotinas apresentadas foi utilizada uma importante operação – $\bigotimes$ – responsável pela combinação de teoremas e soluções oriundos de objetivos dependentes. Ela pode ser assim explicada:

- os parâmetros de entrada são um conjunto de soluções, um conjunto de teoremas e um conjunto de literais, tal que a posição n do conjunto de soluções e a posição n do conjunto de teoremas são, respectivamente o conjunto das soluções e dos teoremas do literal da posição n do conjunto de literais;

- as soluções dos literais, uma de cada literal, são combinadas de acordo com a compatibilidade das substituições. Toda combinação composta por uma solução de cada literal é uma solução resultante;

- os teoremas são combinados com as soluções respeitando que: cada teorema somente é um teorema se sua substituição associada combina, simultaneamente, com a substituição dos demais literais. Se o caso, seja θ a combinação destas substituições, a instância θ do teorema é um teorema resultante com substituição associada θ;

- para cada solução θ dos literais $L \in literais$ que não combinaram com teoremas de $literais - L$, a instância θ de $literais - L$ é um teorema resultante com substituição associada θ;

- para cada teorema dos literais $L \in literais$ com substituição associada θ que não combinou com soluções para os literais de $literais - L$, a instância θ de $L \cup (literais - L)$ é um teorema resultante com substituição associada θ;

- a operação retorna as soluções e os teoremas resultantes.

Esta é uma operação de combinação bastante complexa por envolver o cálculo do conjunto potência[3] das soluções compatíveis e dos teoremas compatíveis. De acordo com a forma que é implementada pode acarretar um grande volume de teoremas resultantes subsumidos. Este pode ser um fator bastante negativo uma vez que objetivos dependentes combinados são, durante a inferência, novamente combinados com objetivos dependentes superiores da árvore de objetivos.

Em nossa implementação, optamos por realizar esta combinação utilizando um hipercubo como estrutura subjacente. O uso desta estrutura aqui é bastante similar ao seu uso para o cálculo da transformação dual e para a identificação de cláusulas subsumidas, apresentados em capítulos anteriores. Para o cálculo da operação $\bigotimes$, cada eixo do hipercubo representa um literal de $literais$. Nesta estrutura os conjuntos de soluções e teoremas são distribuídos nos vértices da primeira dimensão de acordo com

[3]Na prática, este cálculo pode ser otimizado com a consideração apenas das substituições compatíveis das soluções.

o literal representado. A partir destes vértices, os conjuntos são propagados para os vértices sucessores da próxima dimensão onde são combinados. As soluções e teoremas resultantes desta combinação são novamente propagados e combinados e assim sucessivamente até todos as soluções e teoremas resultantes, dos vértices de dimensões inferiores, serem combinados no sucessor comum, isto é, o vértice de última dimensão do hipercubo.

Em cada vértice do hipercubo de segunda dimensão ou superior é executada a seguinte rotina:

Holo-inference$(packs, literals1, literals2)$

1. $(solutions1, solutions2, theorems1, theorems2) \leftarrow packs$
2. $lit1 \leftarrow (literals1 - literals2), lit2 \leftarrow (literals2 - literals1),$
   $comm \leftarrow (dimension - 2)$
3. $solutions \leftarrow (solutions1 \times solutions2)$
4. $theorems \leftarrow (theorems1 \times solutions2)$
5. $theorems \leftarrow theorems \cup (theorems2 \times solutions1)$
6. $theorems \leftarrow theorems \cup (theorems1 \times theorems2)$
7. $theorems \leftarrow theorems \cup (remain(solutions1))$
8. $theorems \leftarrow theorems \cup (remain(solutions2))$
9. $theorems \leftarrow theorems \cup (remain(theorems1))$
10. $theorems \leftarrow theorems \cup (remain(theorems2))$
11. Return $(solutions, theorems)$

a qual recebe dos vértices de dimensões inferiores os "pacotes" contendo as soluções e teoremas calculados e as listas de literais para os quais as soluções e os teoremas foram calculados. Como um vértice de dimensão n difere de apenas um eixo dos vértices da dimensão $n - 1$, naquela dimensão apenas pacotes oriundos de dois vértices desta são necessários para o cálculo das soluções e teoremas. De uma forma geral, a rotina pode ser assim explicada:

1. as soluções e teoremas de cada pacote recebido são colocados em variáveis (solutions1, solutions2, theorems1 e theorems2);

2. outras variáveis são calculadas: $lit1$ o literal de $literais1$ que não se encontra em $literais2$, $lit2$ o oposto e $comm$ o número de literais em comum;

3. para cada par $(sol1, sol2)$ da distribuição de $solutions1$ e $solutions2$, compatível com substituição θ, e cujo número de membros considerados em comum é igual

a *comm*, a substituição θ é uma solução (cujos membros considerados são *sol*1 e *sol*2 e substituição associada é θ);

4. para cada par $(th1, sol2)$ da distribuição de *theorems*1 e *solutions*2, compatível com substituição θ, e cujo número de membros considerados em comum é igual a *comm*, $th1\theta$ é um teorema;

5. para cada par $(th2, sol1)$ da distribuição de *theorems*2 e *solutions*1, compatível com substituição θ, e cujo número de membros considerados em comum é igual a *comm*, $th2\theta$ é um teorema;

6. para cada par $(th1, th2)$ da distribuição dos teoremas de *theorems*1 que não deram origem à nenhum teorema no item 4 e dos teoremas de *theorems*2 que não deram origem à nenhum teorema no item 5, compatível com substituição θ, e cujo número de membros considerados em comum é igual a *comm*, $(th1 \cup th2)\theta$ é um teorema;

7. para cada solução $sol \in solutions1$, de substituição θ, que não deu origem à nenhuma solução no item 3 e a nenhum teorema no item 5, $lit2\theta$ é um teorema;

8. para cada solução $sol \in solutions2$, de substituição θ, que não deu origem à nenhuma solução no item 3 e a nenhum teorema no item 4, $lit1\theta$ é um teorema;

9. para cada teorema $th \in theorems1$, de substituição θ, que não deu origem à nenhum teorema nos ítens 4 ou 6, $(\{lit2\} \cup th)\theta$ é um teorema;

10. para cada teorema $th \in theorems2$, de substituição θ, que não deu origem à nenhum teorema nos ítens 5 ou 6, $(\{lit1\} \cup th)\theta$ é um teorema;

11. as soluções e teoremas acumulados nas variáveis *solutions* e *theorems* são retornados.

Esta solução tem a vantagem de combinar, isoladamente em cada vértice, apenas soluções e teoremas advindos de dois literais, o que diminui a complexidade desta combinação. Outra vantagem é a facilidade de evitar efetuar a mesma combinação mais de uma vez. E, por fim, outra vantagem é a possibilidade de distribuir o processo de geração de teoremas e soluções.

7.3 Resultados

A estratégia proposta foi implementada no *LOGIK* [41], um ambiente voltado para ensino e desenvolvimento de sistemas em lógica de primeira ordem. O *LOGIK* é um sistema escrito em *Common Lisp* utilizando *Clos*, um módulo de orientação a objeto para Lisp, no qual as entidades da sintaxe lógica (variáveis, funções, predicados, etc.) são implementados como classes e as operações lógicas (substituição, unificação, subsunção, etc.) como métodos.

As mesmas fórmulas foram provadas utilizando esta implementação da estratégia *Goal* e a versão 3.1 do Otter (http://www-unix.mcs.anl.gov/AR/otter/), um dos mais bem sucedidos provadores automáticos de teoremas (que utiliza diversas estratégias da resolução). Os resultados obtidos são mostrados na tabela 7.3.

As fórmulas escolhidas foram retiradas, em grande parte, do artigo *Seventy-Five Problems for Testing Automatic Theorem Provers* [29] de *F.J.Pelletier* e fazem também parte da biblioteca de problemas *TPTP* (http://www.math.miami.edu/~tptp/). O nome das fórmulas no referido artigo e na biblioteca estão indicados nas colunas *Pelletier* e *TPTP* respectivamente. As fórmulas foram selecionados segundo seu tamanho e dificuldade (indicada na tabela pelo número de pontos atribuídos a elas, onde o número de pontos é diretamente proporcional à dificuldade apresentada em sua solução por provadores automáticos). O tamanho das fórmulas pode ser visto nas colunas *conj.* e *disj.* da tabela, que contêm o número de cláusulas (duais) das formas conjuntiva e disjuntiva respectivamente. A escolha de fórmulas de tamanho reduzido deve-se à dificuldade (tempo e recursos computacionais) do cálculo da forma disjuntiva de fórmulas maiores, uma vez que a forma disjuntiva calculada é não condensada. Apesar de não fazer parte do artigo de *Pelletier*, para enriquecer a comparação, foram adicionados os resultados apresentados na solução de algumas outras fórmulas da biblioteca *TPTP*. As fórmulas utilizadas encontram-se no apêndice C.

Foram considerados apenas o número de ciclos necessários à refutação e o número de teoremas intermediários retidos em cada estratégia. Tempos não foram considerados devidos às diferenças de implementação. O Otter é um sistema extremamente bem desenvolvido e utiliza diversas técnicas de engenharia de softwre em sua implementação que lhe conferem eficiência. Nossa implementação é voltada apenas ao teste da estratégia e não se preocupa (muito) com eficiência (linguagem, indexações, etc.). Outro detalhe importante é que o Otter possui integrado diversos mecanismos de avaliação da fórmula de entrada e decisão da estratégia (todas da resolução) a ser usada para a sua refutação. O símbolo "-" nos resultados indica a incapacidade computacional (tempo ou memória) de refutação da fórmula em nossa implementação.

Fórmulas					*Goal*		*Otter*	
Pontos	*Pelletier*	*TPTP*	*conj.*	*disj.*	*ciclos*	*teor.*	*ciclos*	*teor.*
3	22	SYN052-1	5	32	1	0	2	4
3	44	SYN068-1	7	64	1	0	3	7
3	48	SYN071-1	7	24	3	18	8	20
3	58	SYN080-1	7	24	1	0	3	3
5	21	SYN051-1	4	16	1	0	2	3
5	31	SYN061-1	5	8	1	0	1	2
5	49	SYN072-1	9	36	9	54	10	53
6	24	SYN054-1	6	96	1	0	5	7
6	27	SYN057-1	7	24	1	0	4	5
6	30	SYN060-1	7	64	1	0	3	6
6	32	SYN062-1	7	54	1	0	3	3
6	46	SYN070-1	9	4860	1	0	9	37
7	25	SYN055-1	7	216	1	0	6	10
8	28	SYN058-1	9	96	1	0	3	3
8	55	PUZ001-1	12	288	1	0	6	14
9	69	LCL039-1	8	6	-	-	25	274.961
10	47	PUZ031-1	26	enorme	-	-	17	68
		NUM001-1	13	243	1	0	3	31.307
		COM001-1	11	36	1	0	3	9
		COM002-1	19	36	3	4	7	45

Tabela 7.3: Número de ciclos e de teoremas retidos durante refutações em diferentes estratégias.

Pela tabela pode-se constatar que, de uma forma geral, a estratégia *Goal* apresenta um bom desempenho. A redução do número de ciclos de inferência necessários à refutação contribui consideravelmente para a redução do número de teoremas intermediários retidos. A redução destes dois fatores pode ser considerada uma importante característica da estratégia proposta. Porém, este assunto merece um estudo mais aprofundado pois, apesar do número de ciclos necessários à refutação pela estratégia *Goal* ser geralmente menor do que os ciclos necessários à refutação por resolução, o ciclo daquela estratégia é geralmente "maior" do que o ciclo da maioria das estratégias da resolução. Também nas refutações de um único ciclo, cujo número de teoremas retidos é *0*, não se deve esquecer que, mesmo não gerando nenhum teorema intermediário, o esforço computacional (número de possibilidades testadas e conseqüentes substituições combinadas) realizado para a refutação da fórmula pode ser considerável, como é o caso, por exemplo, da fórmula *NUM001-1*.

7.4 Comentários

Esta estratégia pode ser considerada uma mistura entre as duas famílias de provadores de teoremas:

- com a conexão a semelhança é bastante clara: a estratégia busca a refutação montando uma árvore de eliminação de literais, que é exatamente o que a conexão faz, durante a busca da *spanning mating* considerando as conexões. Já as diferenças não são tão claras, mas importantes: em nossa estratégia, os objetivos são eliminados independentemente (e podem, inclusive, ser processados em paralelo) o que introduz *justiça* (fairness) na avaliação de eliminações para um literal. Nas conexões a consideração dos pares complementares é feita seqüencialmente para cada literal da cláusula seguida da aplicação da substituição envolvida. Este mecanismo "força" a busca da solução com substituições compatíveis com as previamente consideradas. Devido a isto a conexão tem de incluir procedimentos de "backtraking" durante sua busca. Outra grande diferença é o processamento em ciclos de inferência. Na conexão este conceito não existe, o que implica normalmente o uso de várias instâncias de uma mesma cláusula, tornando o controle das variáveis envolvidas bem mais complexo. Outro grande problema da ausência de ciclos na conexão é a não ciência, por um ramo de busca, dos demais o que pode causar buscas paralelas para um mesmo objetivo. Em nossa estratégia, objetivos não solucionados geram teoremas. Estes teoremas gerados dão origem a objetivos. Estes objetivos são unidos à fórmula e esta é condensada, o que causa a remoção de muitos destes objetivos gerados (ou por duplicidade ou por serem subsumidos). Além disto, a solução para os objetivos não redundantes é buscada uma única vez.

- com a resolução, a semelhança é a refutação por ciclos, que imputa àquele método a capacidade de considerar novos teoremas (os resolventes) oriundos da resolução de diferentes cláusulas, sem exigir nenhuma estrutura rígida como a de árvores na busca da refutação. As diferenças de nossa estratégia com a resolução, por sua vez, coincidem com as semelhanças apresentadas no item anterior.

Independentemente de semelhanças e diferenças, a estratégia *Goal* apresenta algumas boas características tais quais:

- a possibilidade da busca de soluções para cada objetivo, efetuada pela rotina *Solutions*, poder ser processada de maneira independente. Isto é, cada possibilidade pode ser considerada por processos paralelos, característica pouco comum em estratégias de prova automática de teoremas; e

- a busca da refutação convergir rapidamente para a solução, quando esta pertence ao ciclo em processo. Esta característica deve-se às sucessivas aplicações de substituições (o que define previamente para qual instância se deseja encontrar a solução) e a consideração dos caminhos já eliminados (o que descarta caminhos a percorrer na busca da solução).

Apenas a título de comparação, a mesma fórmula do exemplo 34, que é refutada em apenas um ciclo com esta estratégia através de algumas poucas possibilidades testadas, tem os seguintes resultados quando provada através da estratégia da resolução *Conjunto de Suporte*:

- refutação alcançada no quarto ciclo
- 1028 unificações (pares complementares) testadas
- 93 resolventes (teoremas) gerados

No entanto, apresenta uma característica que pode ser uma desvantagem para fórmulas cuja refutação exige mais de um ciclo e, a cada ciclo são gerados muitos teoremas. Com estas características, como cada teorema gerado em um ciclo é um objetivo inicial do ciclo subseqüente, o número de objetivos iniciais pode crescer exageradamente a cada ciclo, aumentando o número de soluções a serem calculadas. (Por outro lado, o crescimento do espaço de busca ao longo dos ciclos de inferência é uma desvantagem comum a todos os métodos de prova.)

Este é o caso, por exemplo, de uma fórmula discutida em *Association for Automated Reasoning (AAR) Newsletter "Non-Obviousness"* [16] onde é proposta uma classificação para as provas das fórmulas: óbvias e não óbvias. Não óbvia seria a prova que requer o uso de mais de uma instância de cláusulas da fórmula, caso contrário, mesmo que longa, a prova seria óbvia. Como exemplo de prova não óbvia, é apresentada a seguinte fórmula:

$$W_c = \langle \begin{array}{ll} 1 & [\neg P(x,y), \neg P(y,z), P(x,z)] \\ 2 & [\neg Q(x,y), \neg Q(y,z), Q(x,z)] \\ 3 & [\neg Q(x,y), Q(y,x)] \\ 4 & [P(x,y), Q(x,y)] \\ 5 & [\neg P(a,b)] \\ 6 & [\neg Q(c,d)] \ \rangle \end{array}$$

na qual o teorema a ser provado é composto pelas cláusulas 5 e 6. É dito que, para a prova, foram gerados mais de 1800 teoremas intermediários. Na estratégia proposta são gerados quase 200 teoremas no primeiro ciclo. Isto implica o cálculo da forma dual de uma fórmula com aproximadamente 200 cláusulas (que é enorme!) e a avaliação de quase 200 novos objetivos iniciais no segundo ciclo. Para cada um destes são avaliadas as possibilidades de prova ou novos teoremas são gerados. Se a prova requer mais de 2 ciclos, o problema pode tornar-se grande demais para os recursos computacionais (em nossa implementação não conseguimos calcular a forma dual da fórmula inicial do segundo ciclo).

Uma saída para este problema seria a escolha de um dentre os novo teoremas como objetivo e seguir a prova apenas a partir deste objetivo. Esta é uma possibilidade a ser estudada, principalmente sua completude e como efetuar a escolha pois uma má escolha poderia aumentar o número de ciclos necessários à refutação, o que também não é de todo desejável.

7.5 Conclusão

A estratégia *Goal* é correta uma vez que: (i) a refutação somente é encontrada pela eliminação de todos os caminhos da fórmula, o que indica a insatisfazibilidade da fórmula e (ii) os teoremas através dela gerados são teoremas da fórmula (o que pode ser conferido pela r.e. 10). Já para dizer que a estratégia é completa, é necessário um estudo mais aprofundado. Temos fortes indícios de que a estratégia é completa: devido à sua semelhança com a conexão e a resolução (conjunto de suporte), acreditamos que toda a inferência efetuada por uma ou por outra estratégia é representada na estratégia *Goal.*

Uma variação desta estratégia encontra-se no artigo *A Proof Strategy Based on* a *Dual Representation*[4] [9] apresentado na *AISC'2000 Fifth International Conference Artificial Intelligence and Symbolic Computation Theory, Implementations and Applications* realizada em Madri, Espanha em julho de 2000.

Tem-se ainda um grande problema no tocante à aplicação desta estratégia: o cálculo da forma disjuntiva. Mesmo com o algoritmo *Dual*, proposto em capítulo anterior, este cálculo pode inviabilizar a tentativa de refutação de fórmulas maiores. É bem verdade que não implementamos o paralelismo permitido pelo algoritmo *Dual*. Mesmo assim o crescimento da forma disjuntiva é exponencial ao crescimento do tamanho da fórmula. O problema é que problemas reais, quando representados em fórmulas da lógica de primeira ordem, podem facilmente requerer 100 ou mais cláusulas com 2 ou 3 literais cada uma, e este tamanho, já para o primeiro ciclo de inferência de nossa estratégia, é comprovadamente muito grande. Uma alternativa estudada para contornar este problema é tratada no próximo capítulo.

[4]Neste, porém é adotada uma metáfora na qual um literal e suas coordenadas é denominado *quantum*, uma cláusula (conjunto de quanta) de *particle*, uma substituição e os caminhos eliminados por esta compõem uma estrutura denominada *Gluon* e a coleção de gluons que eliminam os mesmos caminhos é denominada *Glueball*.

"The rule is jam tomorrow and jam yesterday but never jam to-day," the Queen said. "It must come sometimes to 'jam today'," Alice objected. "No it can't," said the Queen. "It's jam every other day: to-day isn't any other day, you know." "I don't understand you," said Alice. "It's dreadfully confusing."

Lewis Carroll, Through the Looking Glass, 1871

Capítulo 8

Simplificando(?)

Na tentativa de contornar o problema que o tamanho da forma disjuntiva representa para a inferência baseada nas coordenadas, estudou-se o uso da forma disjuntiva condensada nesta inferência. A princípio acreditou-se que, tendo a fórmula um número menor de cláusulas duais, a inferência baseada nas coordenadas dar-se-ia de forma mais simples e direta. Porém esta simplificação da forma disjuntiva introduz dificuldades no tratamento da inferência, tornando seu estudo "dreadfully confusing" (assustadoramente confuso).

8.1 A condensação no caso proposicional

Mesmo que a estratégia não tenha sido pensada visando o caso proposicional, literais fechados ocorrem em fórmulas de primeira ordem e a simplificação destas incluem, de certa forma, a simplificação proposicional. Além disto algumas mudanças, relativas à inferência, podem ser observadas nesta simplificação.

Sabe-se que na simplificação de uma fórmula em forma normal, algumas cláusulas podem ter literais removidos ou, ainda, serem removidas totalmente da fórmula.

Por exemplo, com a distribuição dos literais da forma conjuntiva da fórmula:

$$W_c \;=\; \langle\, [p,q],[r,q],[s,r]\,\rangle$$

obtém-se sua forma disjuntiva não condensada

$$W_{d'} \;=\; [\,\langle p,r,s\rangle,\langle p,r,r\rangle,\langle p,q,s\rangle,\langle p,q,r\rangle,\langle q,r,s\rangle,\langle q,r,r\rangle,\langle q,q,s\rangle,\langle q,q,r\rangle\,]$$

que, pelas remoção de duplicidades (devido 'as propriedades de idempotência e comutatividade dos operadores) e de cláusulas subsumidas, pode ser simplificada para sua forma disjuntiva condensada

$$W_d = [\,\langle p, r\rangle, \langle q, s\rangle, \langle q, r\rangle\,].$$

Estas mudanças na fórmula acarretam mudanças nas coordenadas da representação da mesma. Por sua vez, estas mudanças acarretam mudanças nas relações estruturais estabelecidas anteriormente. Para estudar os efeitos da simplificação na representação de um fórmula toma-se por exemplo a fórmula apresentada no início desta seção.

Exemplo 36 *A fórmula apresentada anteriormente é assim representada:*

$$\begin{aligned}
W_c &= \langle\ 1 : [p^{\{1,2,3,4\}}, q^{\{5,6,7,8\}}], 2 : [r^{\{1,2,5,6\}}, q^{\{3,4,7,8\}}], 3 : [s^{\{1,3,5,7\}}, r^{\{2,4,6,8\}}]\ \rangle \\
W_{d'} &= [\ 1 : \langle p^{\{1\}}, r^{\{2\}}, s^{\{3\}}\rangle, 2 : \langle p^{\{1\}}, r^{\{2\}}, r^{\{3\}}\rangle, 3 : \langle p^{\{1\}}, q^{\{2\}}, s^{\{3\}}\rangle, \\
&\quad\ \ 4 : \langle p^{\{1\}}, q^{\{2\}}, r^{\{3\}}\rangle, 5 : \langle q^{\{1\}}, r^{\{2\}}, s^{\{3\}}\rangle, 6 : \langle q^{\{1\}}, r^{\{2\}}, r^{\{3\}}\rangle, \\
&\quad\ \ 7 : \langle q^{\{1\}}, q^{\{2\}}, s^{\{3\}}\rangle, 8 : \langle q^{\{1\}}, q^{\{2\}}, r^{\{3\}}\rangle\ ]
\end{aligned}$$

Pela idempotência dos símbolos proposicionais, a cláusula dual $\langle q, r, r\rangle$ pode ser simplificada para $\langle q, r\rangle$. Por representarem o mesmo símbolo proposicional, nenhuma distinção é feita entre qual dos r é removido (o r oriundo da segunda cláusula ou o r da terceira). Sendo assim, a cláusula dual $6 : \langle q^{\{1\}}, r^{\{2\}}, r^{\{3\}}\rangle$ de $W_{d'}$ poderia ser simplificada para $\langle q^{\{1\}}, r^{\{2\}}\rangle$ ou para $\langle q^{\{1\}}, r^{\{3\}}\rangle$, o que não faz nenhum sentido já que são a mesma cláusula dual. Para representar esta unicidade de um símbolo proposicional, os conjuntos de coordenadas dos literais de mesmo símbolo proposicional devem ser unidos, sendo $\langle q^{\{1\}}, r^{\{2,3\}}\rangle$ a representação da cláusula dual simplificada.

Exemplo 37 *Com a união das coordenadas dos literais de mesmo símbolo proposicional e remoção dos literais duplicados a fórmula é assim representada:*

$$\begin{aligned}
W_c &= \langle\ 1 : [p^{\{1,2,3,4\}}, q^{\{3,4,5,6,7,8\}}], 2 : [r^{\{1,2,4,5,6,8\}}, q^{\{3,4,5,6,7,8\}}], \\
&\quad\ \ 3 : [s^{\{1,3,5,7\}}, r^{\{1,2,4,5,6,8\}}]\ \rangle \\
W_{d''} &= [\ 1 : \langle p^{\{1\}}, r^{\{2,3\}}, s^{\{3\}}\rangle, 2 : \langle p^{\{1\}}, r^{\{2,3\}}\rangle, 3 : \langle p^{\{1\}}, q^{\{1,2\}}, s^{\{3\}}\rangle, \\
&\quad\ \ 4 : \langle p^{\{1\}}, q^{\{1,2\}}, r^{\{2,3\}}\rangle, 5 : \langle q^{\{1,2\}}, r^{\{2,3\}}, s^{\{3\}}\rangle, 6 : \langle q^{\{1,2\}}, r^{\{2,3\}}\rangle, \\
&\quad\ \ 7 : \langle q^{\{1,2\}}, s^{\{3\}}\rangle, 8 : \langle q^{\{1,2\}}, r^{\{2,3\}}\rangle\ ]
\end{aligned}$$

Ainda pela idempotência, a fórmula $\langle q, r\rangle \vee \langle q, r\rangle$ pode ser simplificada para $\langle q, r\rangle$. Esta simplificação permite a remoção de cláusulas duais duplicadas na fórmula. Em

$W_{d''}$ as cláusulas duais 6 e 8 são iguais e uma pode ser removida.

Exemplo 38 *Com a remoção da oitava cláusula dual tem-se:*

$$\begin{aligned}
W_c &= \langle\ 1: [p^{\{1,2,3,4\}}, q^{\{3,4,5,6,7\}}], 2: [r^{\{1,2,4,5,6\}}, q^{\{3,4,5,6,7\}}], \\
& 3: [s^{\{1,3,5,7\}}, r^{\{1,2,4,5,6\}}]\ \rangle \\
W_{d'''} &= [\ 1: \langle p^{\{1\}}, r^{\{2,3\}}, s^{\{3\}}\rangle, 2: \langle p^{\{1\}}, r^{\{2,3\}}\rangle, 3: \langle p^{\{1\}}, q^{\{1,2\}}, s^{\{3\}}\rangle, \\
& 4: \langle p^{\{1\}}, q^{\{1,2\}}, r^{\{2,3\}}\rangle, 5: \langle q^{\{1,2\}}, r^{\{2,3\}}, s^{\{3\}}\rangle, 6: \langle q^{\{1,2\}}, r^{\{2,3\}}\rangle, \\
& 7: \langle q^{\{1,2\}}, s^{\{3\}}\rangle\ ]
\end{aligned}$$

Já, pela definição de subsunção, a fórmula $\langle p, r, s\rangle \vee \langle p, r\rangle$ pode ser simplificada para $\langle p, r\rangle$. Esta simplificação permite a remoção de cláusulas duais subsumidas na fórmula. Em $W_{d'''}$ as cláusulas duais 1, 3, 4 e 5 podem ser removidas por serem subsumidas pelas cláusulas duais 2, 7, 6 e 7 respectivamente.

Exemplo 39 *Com a remoção das cláusulas duais subsumidas, a forma disjuntiva alcança sua forma condensada:*

$$\begin{aligned}
W_c &= \langle\ 1: [p^{\{2\}}, q^{\{6,7\}}], 2: [r^{\{2,6\}}, q^{\{6,7\}}], 3: [s^{\{7\}}, r^{\{2,6\}}]\ \rangle \\
W_d &= [\ 2: \langle p^{\{1\}}, r^{\{2,3\}}\rangle, 6: \langle q^{\{1,2\}}, r^{\{2,3\}}\rangle, 7: \langle q^{\{1,2\}}, s^{\{3\}}\rangle\ ]
\end{aligned}$$

Com a idempotência e subsunção consideradas, e decorrentes remoções efetuadas, a representação da fórmula sofre duas mudanças: acréscimo de coordenadas nos literais de forma a todo literal de mesmo símbolo proposicional ter o mesmo conjunto de coordenadas, i.e, ser o mesmo, e remoção de coordenadas conjuntivas devido à remoção de cláusulas duais. Acréscimo de coordenadas em literais implica a representação de mais de uma tupla da distribuição de W_c por uma coordenada. Por sua vez, a remoção de uma coordenada implica a não representação de uma daquelas tuplas.

Com a inserção de coordenadas conjuntivas, as coordenadas conjuntivas de literais de mesma cláusula deixam de ser disjuntos. Este é o caso da cláusula 2 do exemplo 39, onde a coordenada 6 foi acrescida ao literal q. Por outro lado, a cláusula dual 8 foi removida por ser igual à cláusula dual 6. Na figura 8.1 pode-se verificar que a representação da tupla, antes representada pela coordenada 8, mantém-se na representação. Assim, após a simplificação, a coordenada 6 representa as tuplas (q, r, r) e (q, q, r) equivalentes.

Se a representação de tuplas removidas por duplicidade permanece na representação após a simplificação de uma fórmula, este não é o caso das tuplas representadas por

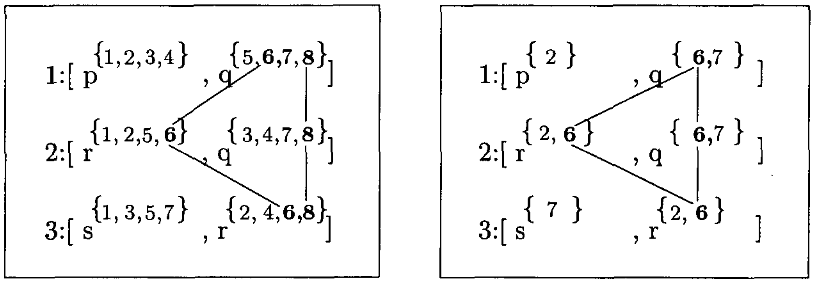

Figura 8.1: Remoção da coordenada 8 por duplicidade.

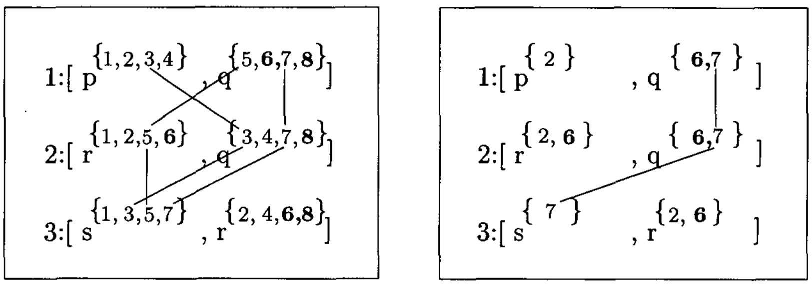

Figura 8.2: Remoção das coordenadas 3 e 5 por subsunção.

cláusulas subsumidas. Na figura 8.2 são mostradas as tuplas representadas pelas coordenadas 3 e 5 subsumidas pela coordenada 7 na simplificação. Pela figura constata-se que estas desaparecem da representação.

Pela r.e. 7, as coordenadas de l_1 e l_2, literais de mesma cláusula, representam tuplas que diferem apenas por l_1 e l_2. Assim, sendo l_3 um literal de cláusula diferente, para toda cláusula dual D_1, que contém o par (l_1, l_3), existe em W_d uma cláusula dual $D_2 = (D_1 - \{l_1\}) \cup \{l_2\}$. Assim $D_1 = \langle \ldots, l_1, l_3, \ldots \rangle$ e $D_2 = \langle \ldots, l_2, l_3, \ldots \rangle$. Então, quando $l_1 = l_3$, $D_1 = \langle \ldots, l_3, l_3, \ldots \rangle$ e, portanto, D_2 é subsumida por D_1. Por exemplo, seja $l_1 = q$ e $l_2 = p$ literais da cláusula 1. Então todos os caminhos que passam por estes literais diferem apenas por q e p. Seja $l_3 = q$ literal da cláusula 2 e L_n uma coleção de literais. Então para cada caminho $\{l_1, l_3\} \cup L_n$ existe um caminho $\{l_2, l_3\} \cup L_n$. De fato, pela figura 8.2 constata-se que para o caminho 7 $\{q, q, s\}$ existe o caminho 3 $\{p, q, s\}$ (com $L_1 = \{s\}$) e para o caminho 8 $\{q, q, r\}$ existe o caminho 4 $\{p, q, r\}$ (com $L_2 = \{r\}$). Como $l_1 = l_3$, então para cada L_n a cláusula dual representada pelo caminho $\{l_1, l_3\} \cup L_n$ subsume a cláusula dual representada pelo caminho $\{l_2, l_3\} \cup L_n$: a cláusula dual 7 subsume a cláusula dual 3 e a cláusula dual 8 subsume a cláusula dual 4.

Seguindo-se o mesmo raciocínio e considerando-se $l_1 = q$ e $l_2 = r$ literais da cláusula 2 e $l_3 = q$ literal da cláusula 1, chega-se a conclusão que a cláusula dual 7 ($\{q, q, s\}$) subsume a cláusula dual 5 ($\{r, q, s\}$) e a cláusula dual 8 ($\{q, q, r\}$) subsume a cláusula dual 6 ($\{r, q, r\}$). Por outro lado, considerando-se $l_1 = r$ e $l_2 = q$ literais da cláusula 2 e $l_3 = r$ literal da cláusula 3, conclui-se que a cláusula dual 2 ($\{r, r, p\}$) subsume a cláusula dual 4 ($\{q, r, p\}$) e a cláusula dual 6 ($\{r, r, q\}$) subsume a cláusula dual 8 ($\{q, r, q\}$). Como a cláusula dual 6 subsume a cláusula dual 8 e vice-versa, vê-se que a igualdade entre cláusulas duais corresponde à dupla subsunção e, neste caso, apenas uma pode ser removida durante a simplificação.

Com a simplificação, ocorrem mudanças nas coordenadas de cada literal e com estas algumas relações estruturais estabelecidas sofrem alterações. Pelo exemplo apresentado constata-se que a remoção de duplicidades de uma fórmula não traz nenhum efeito maior nas coordenadas conjuntivas de sua representação, o maior efeito observado foi a representação de mais de uma tupla por uma mesma coordenada. Com esta mudança as relações estruturais não sofreriam alterações consideráveis. No entanto, a remoção de cláusulas subsumidas causa um "desbalanceamento" na representação da distribuição. O desbalanceamento na representação causado pela remoção de cláusulas subsumidas altera uma importante relação estrutural considerada para a eliminação de literais: a r.e. 9. Por esta r.e. tinha-se garantida a intersecção dos conjuntos de coordenadas de todos literais da fórmula. Por conseguinte, era garantido que todo o literal de uma fórmula contribuía para a eliminação de seu literal complementar. Com a alteração daquela r.e. para o caso da representação de fórmulas condensadas esta garantia é perdida, como pode ser visto no exemplo seguinte:

Exemplo 40 *A forma conjuntiva da representação cruzada de uma fórmula contraditória condensada*[1]*:*

$$
\begin{aligned}
W_c = \langle\ & 1 : [\neg p^{\{1,2\}}, q^{\{2,3\}}], \\
& 2 : [\neg p^{\{1,2\}}, r^{\{3\}}], \\
& 3 : [\neg r^{\{1\}}, q^{\{2,3\}}], \\
& 4 : [p^{\{1,2,3\}}], \\
& 5 : [\neg q^{\{1,2,3\}}]\ \rangle
\end{aligned}
$$

Neste exemplo pode-se observar que as coordenadas do par $(r^{\{3\}}, \neg r^{\{1\}})$ são dis-

[1] *Apesar de ser mostrada apenas a forma conjuntiva, a forma disjuntiva da fórmula foi calculada e consdensada, dando origem às coordenadas da forma conjuntiva.*

juntas e portanto nenhuma cláusula dual é eliminada por este par de literais. Porém, pode-se observar também que ambos literais podem ser eliminados por outros pares de literais complementares: o literal $r^{\{3\}}$ é eliminado com a eliminação da cláusula dual 3 que é contraditória pela ocorrência do par $(q^{\{2,3\}}, \neg q^{\{1,2,3\}})$ e o literal $\neg r^{\{1\}}$ é eliminado com a eliminação da cláusula dual 1 que é contraditória pela ocorrência do par $(\neg p^{\{1,2\}}, p^{\{1,2,3\}})$. Pode ser que a possibilidade de eliminação de cada literal do par complementar, sem considerar os literais do par propriamente dito, signifique que a fórmula do exemplo é contraditória independentemente deste par de literais complementares (já que, nesta fórmula, os literais r e $\neg r$ podem ser substituídos respectivamente por a e b, que a fórmula resultante permanece contraditória). Pode ser, mas isto não foi ainda comprovado.

O mais importante a ser observado neste exemplo é que, mesmo na forma condensada, uma fórmula contraditória apresenta todas suas cláusulas duais contraditórias. Isto é facilmente comprovável: todas as cláusulas duais de uma fórmula não simplificada são contraditórias. Com a simplificação ocorre ou a retirada de literais duplicados de uma cláusula dual ou a retirada de cláusulas duais subsumidas. A retirada de literais duplicados nada afeta a insatisfazibilidade de uma cláusula dual. O mesmo acontece com a retirada de cláusulas subsumidas: se todas as cláusulas duais são contraditórias, então todas as cláusulas duais não subsumidas também o são.

Referente à eliminação de literais baseada nas coordenadas, pode-se dizer que: como cada cláusula dual da forma disjuntiva condensada de uma fórmula contraditória é contraditória, cada uma destas cláusulas duais contém pelo menos um par de literais complementares e estes literais, por sua vez, têm pelo menos uma coordenada conjuntiva em comum. E, como toda cláusula dual pode ser eliminada (e, por conseguinte, toda coordenada pode ser eliminada) todo literal de uma fórmula contraditória condensada pode ser eliminado.

Conclui-se então que, mesmo alterando algumas relações estruturais, a condensação de uma fórmula proposicional não tem nenhum efeito negativo no que antes foi estabelecido para inferência. Mesmo com a eliminação, devida à simplificação, da representação de tuplas da distribuição da forma conjuntiva, as tuplas remanescentes na representação são suficientes para manter as relações necessárias à inferência.

8.2 A condensação no caso de primeira ordem

No caso proposicional a inferência, como mostrada, mantém-se mesmo para fórmulas cuja forma disjuntiva é condensada. Neste caso, a relação de subsunção, utilizada na condensação da fórmula, dá-se pela igualdade de literais e a refutação dá-se sempre em um único ciclo. Isto, no entanto, não é verdade para fórmulas de primeira ordem. Mesmo considerando-se para a condensação apenas a igualdade entre literais fechados, o desbalanceamento das coordenadas acarreta conseqüências para fórmulas cuja refutação exige mais de um ciclo.

Este é o caso da fórmula do exemplo 41 cuja forma disjuntiva foi condensada considerando-se apenas a igualdade dos literais fechados (a forma disjuntiva não condensada contém 1536 cláusulas duais).

Exemplo 41 *Uma fórmula contraditória*[2] *cuja forma disjuntiva foi condensada com a consideração de apenas a igualdade dos literais fechados:*

$$
\begin{array}{lrl}
W_c = \langle & 0: [& Eq(x_0,x_0)^{(\{0,\dots,47\})}] \\
& 1: [& \neg Eq(x_2,x_1)^{(\{6,\dots,11,18,\dots,23,30,\dots,35,42,\dots,47\})}, \\
& & Eq(x_1,x_2)^{(\{0,\dots,5,12,\dots,17,24,\dots,29,36,\dots,41\})}] \\
& 2: [& \neg Eq(x_4,x_3)^{(\{4,5,10,11,16,17,22,23,28,29,34,35,40,41,46,47\})}, \\
& & \neg Eq(x_3,x_5)^{(\{2,3,8,9,14,15,20,21,26,27,32,33,38,39,44,45\})}, \\
& & Eq(x_4,x_5)^{(\{0,1,6,7,12,13,18,19,24,25,30,31,36,37,42,43\})}] \\
& 3: [& Eq(a,b)^{(\{24,\dots,47\})}, Eq(c,d)^{(\{0,\dots,23\})}] \\
& 4: [& Eq(a,c)^{(\{1,\dots,47\ (\acute{\imath}mpares)\})}, Eq(b,d)^{(\{0,\dots,46\ (pares)\})}] \\
& 5: [& \neg Eq(a,d)^{(\{0,\dots,47\})}] \\
& 6: [& \neg Eq(b,c)^{(\{0,\dots,47\})}] \\
& *7: [& \neg Eq(d,a)^{(\{0,\dots,47\})}] \\
& *8: [& Eq(c,a)^{(\{0,\dots,35\})}, Eq(d,b)^{(\{0,\dots,11,24,\dots,47\})}] \\
& *9: [& Eq(b,a)^{(\{12,\dots,35\})}, Eq(d,b)^{(\{0,\dots,11,24,\dots,47\})}] \\
& *10: [& Eq(c,a)^{(\{0,\dots,35\})}, Eq(d,c)^{(\{0,\dots,11,36,\dots,47\})}] \\
& *11: [& Eq(b,a)^{(\{12,\dots,35\})}, Eq(d,c)^{(\{0,\dots,11,36,\dots,47\})}] \\
& *12: [& \neg Eq(c,b)^{(\{0,\dots,47\})}] \\
& *13: [& Eq(a,b)^{(\{24,\dots,47\})}, Eq(c,a)^{(\{0,\dots,35\})}] \\
& *14: [& Eq(d,b)^{(\{0,\dots,11,24,\dots,47\})}, Eq(c,d)^{(\{0,\dots,23\})}]\ \rangle
\end{array}
$$

A refutação desta fórmula é mostrada nas figuras 8.3 e 8.4. A princípio, esta refutação pareceria correta, uma vez que todas as coordenadas da fórmula são eliminadas com substituições compatíveis. A refutação mostrada utiliza as cláusulas 1, 2, 11, 12,

[2] *Esta fórmula é obtida no segundo ciclo de refutação pela estratégia Goal do 48º problema apresentado no artigo Seventy-five Problems for Testing ATP[29].*

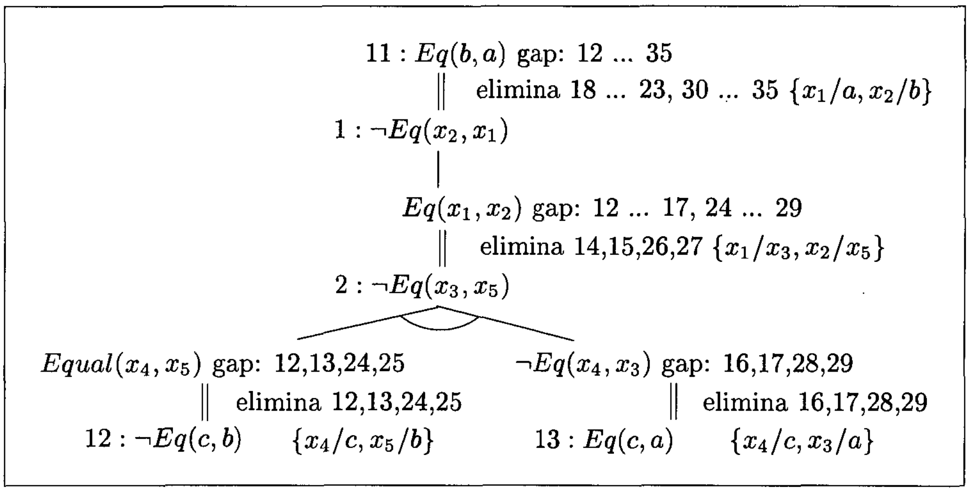

Figura 8.3: Árvore parcial de objetivos para a fórmula do exemplo 41 e as substituições envolvidas.

13 e 14 e, portanto, o conjunto destas cláusulas deveria ser uma fórmula contraditória[3]. No entanto, isto não é verdade: o conjunto destas cláusulas é uma fórmula satisfazível. O problema reside nas cláusulas 13 e 14 utilizadas para a refutação: em ambas apenas um literal contribui para a eliminação do *gap*. As coordenadas dos literais $Eq(a, b)$, da cláusula 13, e $Eq(c, d)$, da cláusula 14, não têm intersecção com *gap*. Seus literais vizinhos são suficientes na cláusula para a eliminação do *gap*. Isto não ocorre quando a cláusula não é simplificada. Todos os literais de uma cláusula contribuem para eliminação de um determinado *gap* durante o processo de inferência. Esta não eliminação é causada pelo "desbalanceamento" das coordenadas devido à simplificação da fórmula.

Em todo o caso, sabe-se de antemão que esta fórmula é contraditória. Mas cabe aqui a pergunta: este desbalanceamento pode gerar refutações incorretas? Não sabemos a resposta. Se o desbalanceamento não torna o método incorreto, então esta simplificação pode ser considerada vantajosa para a refutação uma vez que permite a obtenção de refutações mais simples. E, se este é o caso, como apresentar esta prova? Esta árvore de objetivos apresentada não é "completa" pelo menos (se não incorreta).

8.2.1 A representação cruzada estendida

Para representar fórmulas da lógica de primeira ordem condensada, a representação cruzada foi estendida de forma a representar as relações de subsunção entre os literais

[3]A subfórmula formada pelas cláusulas utilizadas na refutação de uma fórmula deve ser contraditória, uma vez que as mesmas foram suficientes para a refutação da fórmula.

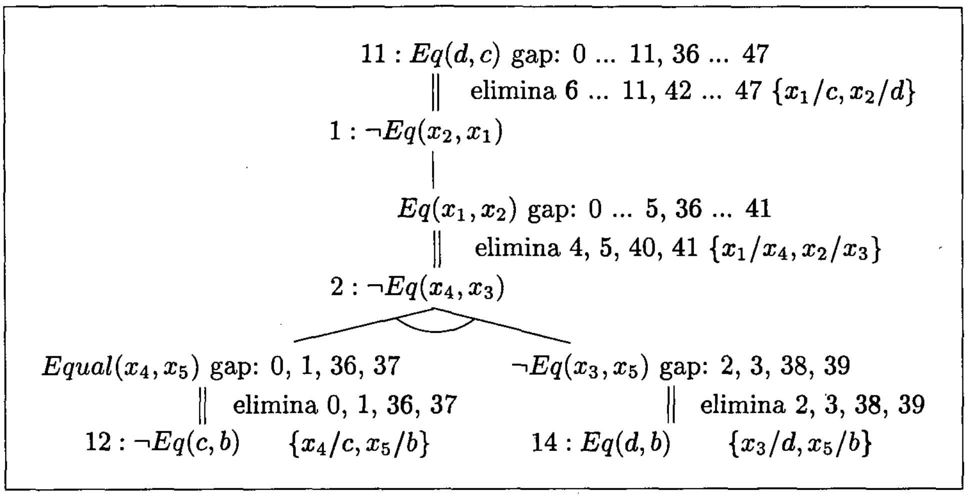

Figura 8.4: Árvore parcial de objetivos para a fórmula do exemplo 41 e as substituições envolvidas.

da fórmula. Para tal, é acrescido um novo conjunto às coordenadas de cada literal da fórmula representada. Este novo conjunto é composto pelos literais subsumidos pelo literal representado. Uma fórmula condensada de primeira ordem, na representação cruzada estendida, é mostrada no próximo exemplo:

Exemplo 42 *A fórmula contraditória do exemplo 33 do capítulo 7:*

$$
\begin{array}{llll}
W_c & = \langle & 1 & [\neg P(x_1)^{(\{3\},\{\})}, P(f(x_1))^{(\{1,2\},\{\})}] \\
 & & 2 & [\neg P(f(x_2))^{(\{2\},\{\neg P(x_1)^{\{3\}}\})}, Q(x_2)^{(\{1\},\{\})}] \\
 & & 3 & [P(f(a))^{(\{3\},\{P(f(x_1))^{\{1,2\}}\})}] \\
 & & 4 & [\neg Q(a)^{(\{1,2,3\},\{\})}] \ \rangle \\
W_d & = [& 1 & \langle P(f(x_1))^{(\{1\},\{P(f(a))^{\{3\}}\})\}}, Q(x_2)^{(\{2\},\{\})}, \neg Q(a)^{(\{4\},\{\})}\rangle \\
 & & 2 & \langle P(f(x_1))^{(\{1\},\{P(f(a))^{\{3\}}\})}, \neg P(f(x_2))^{(\{2\},\{\})}, \neg Q(a)^{(\{4\},\{\})}\rangle \\
 & & 3 & \langle \neg P(x_1)^{(\{1\},\{\neg P(f(x_2))^{\{2\}}\})}, P(f(a))^{(\{3\},\{\})}, \neg Q(a)^{(\{4\},\{\})}\rangle \ ]
\end{array}
$$

Sejam W_c e W_d as formas normais conjuntiva e disjuntiva de uma fórmula representada na representação cruzada estendida, l_1 e l_2 literais desta fórmula, C e D cláusulas de W_c e W_d respectivamente, tem-se:

- as coordenadas de $l_1 \in C$ são o par $(F_c(l_1), Subs_c(l_1))$ onde F_c é o conjunto de coordenadas conjuntivas (já introduzidas nos capítulos anteriores) e $Subs_c(l_1)$

é o *conjunto conjuntivo de subsunção*, possivelmente vazio. Cada elemento de $Subs_c(l_1)$ é um literal l_2 de W_c e suas coordenadas conjuntivas tal que $l_1 \succ_\theta^{or} l_2$;

- as coordenadas de $l_1 \in D$ são o par $(F_d(l_1), S_d(l_1))$ onde F_d é o conjunto de coordenadas disjuntivas (já introduzidas nos capítulos anteriores) e $Subs_d(l_1)$ é o *conjunto disjuntivo de subsunção*, possivelmente vazio. Cada elemento de $Subs_d(l_1)$ é um literal l_2 de W_d e suas coordenadas disjuntivas tal que $l_1 \succ_\theta l_2$;

Na fórmula do exemplo anterior, considerando-se $l_1 = \neg P(f(x_2))$ e $l_2 = \neg P(x_1)$, tem-se que $l_1 \succ_\theta^{or} l_2$ para $\theta = \{x_1/f(x_2)\}$. Portanto, toda ocorrência de l_2 em mesma cláusula de l_1 é subsumida por este literal e, então, l_2 é membro de $Subs_c(l_1)$. Por exemplo, na cláusula 2 de W_c existe uma ocorrência de l_2 subsumida por l_1. Por outro lado, se $l_1 \succ_\theta^{or} l_2$ então $l_2 \succ_\theta l_1$. Então, em toda cláusula dual onde l_2 ocorre, a ocorrência de l_1 é subsumida. Por exemplo, como $F_c(l_2) = \{3\}$, na cláusula dual 3 a ocorrência de l_1 é subsumida. Ambos fatos acima relatados são expressos pelas coordenadas destes literais, em W_c:

- por $F_c(\neg P(f(x_2))) = \{2\}$ sabe-se que $\neg P(f(x_2))$ ocorre apenas na cláusula dual 2 de W_d e, por $Subs_c(\neg P(f(x_2))) = \{\neg P(x_1)^{\{3\}}\}$ sabe-se que $\neg P(f(x_2))$ subsume a ocorrência do literal $\neg P(x_1)$ na cláusula em que ocorre (2) e que, como $F_c(\neg P(x_1)) = \{3\}$, a ocorrência de $\neg P(f(x_2))$ na cláusula dual 3 é subsumida por $\neg P(x_1)$;

- por $F_c(\neg P(x_1)) = \{3\}$ sabe-se que $\neg P(x_1)$ ocorre apenas na cláusula dual 3 de W_d e por $Subs_c(\neg P(x_1)) = \{\}$ sabe-se que $\neg P(x_1)$ não subsume a ocorrência de literal algum e, portanto, não tem nenhuma ocorrência subsumida em nenhuma cláusula dual de W_d.

Em contrapartida, como trata-se de uma representação cruzada, nas coordenadas dos literais em W_d também estão representadas as mesmas relações entre os literais l_1 e l_2 (considerando-se os mesmos valores para l_1 e l_2). As coordenadas destes literais, em W_d, podem ser assim interpretadas:

- por $F_d(\neg P(f(x_2))) = \{2\}$ sabe-se que $\neg P(f(x_2))$ ocorre apenas na segunda cláusula de W_C e, por $Subs_d(\neg P(f(x_2))) = \{\}$ sabe-se que $\neg P(f(x_2))$ não subsume nenhum literal e não tem nenhuma ocorrência subsumida em nenhuma cláusula de W_c;

- por $F_d(\neg P(x_1)) = \{1\}$ sabe-se que $\neg P(x_1)$ ocorre apenas na segunda cláusula de W_c e por $Subs_c(\neg P(x_1)) = \{\neg P(f(x_2))^{\{2\}}\}$ sabe-se que $\neg P(x_1)$ subsume a ocorrência de $\neg P(f(x_2))$ na cláusula dual em que ocorre (3) e que, como $F_d(\neg P(f(x_2))) = \{2\}$, a ocorrência de $\neg P(x_1)$ na cláusula 2 é subsumida por $\neg P(f(x_2))$.

É interessante observar que, com a representação dos literais subsumidos, todas as cláusulas de uma forma normal estão representadas nas coordenadas dos literais de cada cláusula da outra. Toma-se por exemplo a segunda cláusula de W_c, a saber $[\neg P(f(x_2))^{(\{2\},\{\neg P(x_1)^{\{3\}}\})}, Q(x_2)^{(\{1\},\{\})}]$. Nesta tem-se:

- como representante da cláusula dual 1, o literal $Q(x_2)$,
- como representante da cláusula dual 2, o literal $\neg P(f(x_2))$, e
- como representante da cláusula dual 3, o literal subsumido $\neg P(x_1)$.

8.2.2 Eliminação de literais - o caso estendido

Como a eliminação de um literal l de uma fórmula, da forma anteriormente descrita, dá-se pela eliminação das cláusulas duais da fórmula nas quais o literal ocorre, esta considera apenas as coordenadas conjuntivas deste literal, isto é, apenas $F_c(l)$. Assim, na fórmula do exemplo anterior tem-se as seguintes eliminações de literais:

- como $F_c(Q(x_2)^{(\{1\},\{\})}) \subset F_c(\neg Q(a)^{(\{1,2,3\},\{\})})$, o literal $Q(x_2)$ pode ser eliminado com $\theta_1 = \{x_2/a\}$, que torna a primeira cláusula dual explicitamente contraditória;
- por sua vez, como $F_c(\neg P(f(x_2))^{(\{2\},\{\neg P(x_1)^{\{3\}}\})}) \subset F_c(P(f(x_1))^{(\{1,2\},\{\})})$, o literal $\neg P(f(x_2))$ pode ser eliminado com $\theta_2 = \{x_2/x_1\}$, que torna a segunda cláusula dual explicitamente contraditória;
- por fim, como $F_c(\neg P(x_1)^{(\{3\},\{\})}) \subset F_c(P(f(a))^{(\{3\},\{P(f(x_1))^{\{1,2\}}\})})$, o literal $\neg P(x_1)$ pode ser eliminado com $\theta_3 = \{x_1/f(a)\}$, que torna a terceira cláusula dual explicitamente contraditória.

Por serem as substituições incompatíveis, as cláusulas duais não podem tornar-se explicitamente contraditórias simultaneamente. Mesmo assim, os literais $\neg P(f(x_2))$ e $Q(x_2)$, da segunda cláusula, podem ser eliminados simultaneamente com a substituição $\theta_4 = \{(x_2/x_1), (x_2/a)\}$ (resultado da combinação de θ_1 e θ_2). Neste caso, procedendo a eliminação das cláusulas duais contraditórias e cálculo de novo conjunto de cláusulas tem-se $W_{c'} = \langle\ [\neg P(a)], [P(f(a))], [\neg Q(a)]\ \rangle$ e $[\neg P(a)]$ como teorema inferido por esta eliminação. Atendo-se ao novo teorema gerado $[\neg P(a)]$ vê-se que este é a instância θ_4 do literal $\neg P(x_1)$ subsumido pelo literal $\neg P(f(x_2))$. Isto pode ser explicado da seguinte forma: a substituição θ_4 elimina as coordenadas 1 e 2 e não elimina a coordenada 3. Assim, nesta instância, com a eliminação do literal $\neg P(f(x_2))^{(\{2\},\{\neg P(x_1)^{\{3\}}\})}$ da segunda cláusula, o literal $\neg P(x_1)^{(\{3\},\{\})}$ deixa de ser subsumido e pode "voltar" a ocorrer nesta cláusula. Com este exemplo vê-se, então, que as coordenadas subjuntivas são necessárias à correta inferência de teoremas oriundos de cláusulas nas quais ocorre a subsunção de algum literal.

A fim de considerar as coordenadas subsumidas para a inferência, deve-se estender a noção de eliminação de um literal:

> um literal é *completamente eliminado* quando todas as suas coordenadas (inclusive as subsumidas) forem eliminadas.

Desta definição segue que, sendo C uma cláusula de uma fórmula W_c na representação cruzada estendida (e W_d sua forma disjuntiva) e l um literal de C, se l é completamente eliminado com uma substituição θ, então $(C - \{l\})\theta$ é teorema de W_c. Isto pode ser comprovado da seguinte forma: sendo l completamente eliminado por θ todas as cláusulas duais representadas pelas coordenadas de l são explicitamente contraditórias em $W_d\theta$ e podem ser eliminadas gerando $W_{d'}\theta$; como nas coordenadas de $(C - \{l\})$ estão representadas todas as cláusulas duais de $W_{d'}\theta$,$(C - \{l\})\theta$ é uma das cláusulas obtidas pela transformação dual de $W_{d'}\theta$; então $W_c \rightarrow (C - \{l\})\theta$. Por fim, como a eliminação de todos os literais de uma cláusula de W_c não implica todas as cláusulas duais de W_d serem contraditórias, deve-se estender também a condição de refutação da fórmula:

> quando na representação cruzada estendida, a eliminação *completa e simultânea* de todos os literais de uma cláusula da fórmula configura prova de sua insatisfazibilidade.

É natural que, uma vez incluídas as coordenadas subsumidas na representação, elas devem também ser consideradas na definição das coordenadas eliminadas por uma substituição advinda de um par de literais complementares. Dados

- $F_c(Subs_c(l))$ coordenadas conjuntivas que ocorrem em $Subs_c(l)$, tal que $(n \in F_c(Subs_c(l_1))) \rightarrow \exists l_2 \in Subs_c(l)((n \in F_c(l_2)) \vee (n \in F_c(Subs_c(l_2))))$
- $\Omega(l)$ conjunto das coordenadas do literal l candidatas à eliminação, no qual cada membro é um par (α, β), onde α é um conjunto de coordenadas e β é uma substituição, tal que:

– se l é literal de cláusula unitária, então $((F_c(l) \cup F_c(Subs_c(l))), \emptyset)$ é o único membro de $\Omega(l)$,

– senão, os elementos de $\Omega(l)$ são:

  * $(F_c(l), \emptyset)$,
  * $\forall (l_2, \theta) \in Subs_c(l), (F_c(l) \cup F_c(l_2), \theta) \in \Omega(l)$.

tem-se que as coordenadas eliminadas pelo par (l_1, l_2) com a substituição θ é o conjunto $\mho(l_1, l_2)$ de tuplas $\langle \theta, \alpha, \beta \rangle$ onde θ é a substituição da tupla, α são as coordenadas eliminadas e β é sua substituição de restrição. As tuplas são calculadas de forma que $\alpha = \alpha_1 \cup \alpha_2$ e $\beta = \beta_1 \circ \beta_2$ para $(\alpha_1, \beta_1) \in \omega(l_1)$, $(\alpha_2, \beta_2) \in \omega(l_2)$, $\alpha \neq \emptyset$, β_1 e β_2 compatíveis, β compatível com θ e $\circ$ é a operação de combinação de substituições. Duas tuplas t_1 e t_2 podem ser combinadas gerando t_3 – onde $\theta_3 = \theta_1 \circ \theta_2, \alpha_3 = \alpha_1 \cup \alpha 2, \beta_3 = \beta_1 \circ \beta_2$ – se θ_1 e θ_2 são compatíveis, β_1 e β_3 são compatíveis e θ_3 e β_3 são compatíveis.

Para entender os conjuntos Ω e $\mho$ deve-se entender alguns detalhes da forma disjuntiva condensada de uma fórmula. Retornemos à forma conjuntiva do exemplo 42:

$$\begin{array}{rcll} W_c & = & \langle & 1 \quad [\neg P(x_1)^{(\{3\},\{\})}, P(f(x_1))^{(\{1,2\},\{\})}] \\ & & & 2 \quad [\neg P(f(x_2))^{(\{2\},\{\neg P(x_1)^{\{3\}}\})}, Q(x_2)^{(\{1\},\{\})}] \\ & & & 3 \quad [P(f(a))^{(\{3\},\{P(f(x_1))^{\{1,2\}}\})}] \\ & & & 4 \quad [\neg Q(a)^{(\{1,2,3\},\{\})}] \ \rangle \end{array}$$

Nela tem-se:

- $\Omega(\neg P(x_1)) = \{(\{3\}, \{\})\}$
- $\Omega(P(f(x_1))) = \{(\{1,2\}, \{\})\}$
- $\Omega(\neg P(f(x_2))) = \{(\{2\}, \{\}), (\{2,3\}, \{x_1/f(x_2)\})\}$
- $\Omega(Q(x_2)) = \{(\{1\}, \{\})\}$
- $\Omega(P(f(a))) = \{(\{1,2,3\}, \{\})\}$
- $\Omega(\neg Q(a)) = \{(\{1,2,3\}, \{\})\}$

o que indica que as coordenadas de todos os literais, exceto $\neg P(f(x_2))$, podem ser utilizadas para a eliminação do literal sem restrição. Já a coordenada 3 do literal $\neg P(f(x_2))$, por encontrar-se subsumida, pode ser utilizada se a substituição envolvida na eliminação for compatível com a substituição da subsunção $(\{x_1/f(x_2)\})$. Esta restrição deve-se à simplificação efetuada na forma disjuntiva quando de sua condensação.

Deve-se atentar que a forma disjuntiva da fórmula tem a configuração apresentada no exemplo 42 porque a substituição da subsunção pode ser livremente aplicada. Consideremos, no entanto, a forma não condensada da fórmula em questão:

$$
\begin{array}{lcll}
W_c & = & \langle & 1 \quad [\neg P(x_1)^{(\{3,4\},\{\})}, P(f(x_1))^{(\{1,2\},\{\})}] \\
 & & & 2 \quad [\neg P(f(x_2))^{(\{2,3\},\{\})}, Q(x_2)^{(\{1,4\},\{\})}] \\
 & & & 3 \quad [P(f(a))^{(\{1,2,3,4\},\{\})}] \\
 & & & 4 \quad [\neg Q(a)^{(\{1,2,3,4\},\{\})}] \ \rangle \\
W_d & = & [& 1 \quad \langle P(f(x_1))^{(\{1\},\{\})}, Q(x_2)^{(\{2\},\{\})}, P(f(a))^{(\{3\},\{\})}, \neg Q(a)^{(\{4\},\{\})}\rangle \\
 & & & 2 \quad \langle P(f(x_1))^{(\{1\},\{\})}, \neg P(f(x_2))^{(\{2\},\{\})}, P(f(a))^{(\{3\},\{\})}, \neg Q(a)^{(\{4\},\{\})}\rangle \\
 & & & 3 \quad \langle \neg P(x_1)^{(\{1\},\{\})}, \neg P(f(x_2))^{(\{2\},\{\})}, P(f(a))^{(\{3\},\{\})}, \neg Q(a)^{(\{4\},\{\})}\rangle \\
 & & & 4 \quad \langle \neg P(x_1)^{(\{1\},\{\})}, Q(x_2)^{(\{2\},\{\})}, P(f(a))^{(\{3\},\{\})}, \neg Q(a)^{(\{4\},\{\})}\rangle \ ]
\end{array}
$$

Observa-se que esta contém uma cláusula dual, número 4, a mais que a sua forma condensada. Esta cláusula dual foi subsumida pela cláusula dual 3 na condensação. Por isto, para considerar a coordenada 3 subsumida no literal $\neg P(f(x_2))^{(\{2\},\{\neg P(x_1)^{\{3\}}\})}$ para eliminação, deve-se garantir que cláusula dual 4 (inexistente na forma condensada) também seja tornada contraditória, e portanto eliminada com a mesma substituição. Isto é garantido para as instâncias compatíveis com a substituição da subsunção. Por isto esta substituição é chamada de *substituição de restrição*. No exemplo, para considerar a coordenada 3 daquele literal, tem-se como restrição a substituição $\{x_1/f(x_2)\}$, porque neste caso a cláusula dual 4 pode também ser eliminada.

Pode-se também usar este exemplo para explicar a ausência de restrições ao uso de coordenadas subsumidas quando estas ocorrem em um literal de cláusula unitária, como as coordenadas do literal $P(f(a))^{(\{3\},\{P(f(x_1))^{\{1,2\}}\})}$ do exemplo anterior. Sabe-se, pela r.e. 4, que um literal de cláusula unitária ocorre em todas as cláusulas duais da forma disjuntiva não condensada. Assim, a retirada deste literal de uma cláusula dual por subsunção, não causa a remoção de nenhuma outra cláusula dual do conjunto. Isto pode ser explicado como segue: dadas D_1 e D_2 cláusulas duais da forma não condensada de uma fórmula, l um literal de cláusula unitária e $D_1 - \{l\}$ a forma simplificada de D_1 (ou seja, l é subsumido por algum literal de D_1). Como $l \in D_2$, se $D_1 - \{l\} \succ D_2$ então $D_1 - \{l\} \succ D_2 - \{l\}$ uma vez que a ocorrência de l é única em cada cláusula dual. Portanto, se $D_1 - \{l\} \succ D_2 - \{l\}$ é porque $D_1 \succ D_2$, ou seja, a subsunção independe de l.

$1\ [\ \neg P(x_1)^{(\{3\},\{\})},\ P(f(x_1))^{(\{1,2\},\{\})}\ ]$

$2\ [\ \neg P(f(x_2))^{(\{2\},\{\neg P(x_1)^{\{3\}}\})},\ Q(x_2)^{(\{1\},\{\})}\ ]$

$\{x_2/a\}$, eliminando 2,3

$3\ [\ P(f(a))^{(\{3\},\{P(f(x_1))^{\{1,2\}}\})}\ ]$

$\{x_2/a\}$, eliminando 1

$4\ [\ \neg Q(a)^{(\{1,2,3\},\{\})}\ ]$

Figura 8.5: A refutação da fórmula do exemplo 42.

Conhecendo-se o conjunto Ω de cada literal, pode-se calcular o conjunto $\mho$ para definir como as coordenadas (ou caminhos) podem ser eliminados. Para a fórmula do exemplo, tem-se:

- $\mho(\neg P(x_1), P(f(a))) = \{\langle\{x_1/f(a)\}, \{3\}, \{\}\rangle\}$
- $\mho(\neg P(f(x_2)), P(f(x_1))) = \{\langle\{x_1/x_2\}, \{2\}, \{\}\rangle\}$
- $\mho(\neg P(f(x_2)), P(f(a))) = \{\langle\{x_2/a\}, \{2\}, \{\}\rangle, \langle\{x_2/a\}, \{2,3\}, \{x_1/f(x_2)\}\rangle\}$
- $\mho(\neg Q(a), Q(x_2)) = \{\langle\{x_2/a\}, \{1\}, \{\}\rangle\}$

Com isto, da combinação de $\langle\{x_2/a\}, \{1\}, \{\}\rangle$ e $\langle\{x_2/a\}, \{2,3\}, \{x_1/f(x_2)\}\rangle$ tem-se $\langle\{x_2/a\}, \{1,2,3\}, \{x_1/f(x_2)\}\rangle$. Desta tupla conclui-se que todas as coordenadas podem ser simultaneamente eliminadas com a substituição $\{x_2/a\}$, configurando prova da insatisfazibilidade da fórmula do exemplo 42. As eliminações consideradas para a prova podem ser vistas na figura 8.5.

A restrição imposta ao uso de coordenadas subsumidas serve para "preservar a constituição da forma disjuntiva condensada". Por preservar entende-se "manter a forma atual" quanto ao número de cláusulas duais. Deve-se considerar que a forma disjuntiva condensada contém as cláusulas duais que contém devido à subsunções consideradas. Esta forma condensada é a mesma para todas as instâncias da fórmula compatíveis com as substituições consideradas na condensação. Já em instâncias incompatíveis, a forma condensada é diferente (normalmente contendo mais cláusulas duais, o que acarreta mais coordenadas a serem consideradas). No exemplo em questão, esta restrição é a substituição $\{x_1/f(x_2)\}$. Se considerarmos a instância $x_1 = b, x_2 = c$ para a fórmula do exemplo 42, veremos que sua forma condensada é igual à forma não condensada, contendo quatro cláusulas duais (uma vez que as relações de subsunção não mais ocorrem).

Tomemos por exemplo, uma fórmula bem simples:

$$W_c = \langle [P(x), Q(a)], [P(a), Q(b)] \rangle$$

cuja forma disjuntiva não condensada é:

$$W_d = [\langle P(x), P(a) \rangle, \langle P(x), Q(b) \rangle, \langle Q(a), P(a) \rangle, \langle Q(a), Q(b) \rangle]$$

que pode ser condensada para:

$$W_{d'} = [\langle P(x) \rangle, \langle Q(a), P(a) \rangle, \langle Q(a), Q(b) \rangle]$$

na qual foi considerada a substituição $\{x/a\}$ para simplificação. Calculando-se a instância $\theta_1 = \{x/a\}$ das fórmulas, *compatível* com a substituição de simplificação, tem-se:

$$W_c\theta_1 = \langle [P(a), Q(a)], [P(a), Q(b)] \rangle$$

$$W_d\theta_1 = [\langle P(a), P(a) \rangle, \langle P(a), Q(b) \rangle, \langle Q(a), P(a) \rangle, \langle Q(a), Q(b) \rangle]$$

$$W_{d'}\theta_1 = [\langle P(a) \rangle, \langle Q(a), P(a) \rangle, \langle Q(a), Q(b) \rangle]$$

e calculando-se a instância $\theta_2 = \{x/b\}$ das fórmulas, *incompatível* com a substituição de simplificação, tem-se:

$$W_c\theta_2 = \langle [P(b), Q(a)], [P(a), Q(b)] \rangle$$

$$W_d\theta_2 = [\langle P(b), P(a) \rangle, \langle P(b), Q(b) \rangle, \langle Q(a), P(a) \rangle, \langle Q(a), Q(b) \rangle]$$

$$W_{d'}\theta_2 = [\langle P(b) \rangle, \langle Q(a), P(a) \rangle, \langle Q(a), Q(b) \rangle]$$

Considerando-se as fórmulas apresentadas tem-se as seguintes relações de equivalência:

- $W_c \leftrightarrow W_d \leftrightarrow W_{d'}$, como já sabido;
- $W_c\theta_1 \leftrightarrow W_d\theta_1 \leftrightarrow W_{d'}\theta_1$,
- $W_c\theta_2 \leftrightarrow W_d\theta_2$ como era de se esperar, mas
- $W_{d'}\theta_2 \not\leftrightarrow W_c\theta_2$ e $W_{d'}\theta_2 \not\leftrightarrow W_d\theta_2$.

Devido a esta não equivalência das fórmulas nas instâncias não compatíveis às subsunções consideradas, restringe-se a eliminação de coordenadas subsumidas de um par de literais complementares, a instâncias compatíveis àquelas consideradas na condensação.

8.2.3 A não eliminação

Até aqui o uso da forma disjuntiva condensada não representou, ao que parece, nenhum grande problema. Resta-nos ainda apresentar um caso decorrente da simplificação. Considere o seguinte exemplo:

Exemplo 43 *Uma fórmula contraditória:*

$$
\begin{array}{lllll}
W_c & = & \langle & 1 & [\neg P(f(x_1))^{(\{3\},\{\neg P(x_2)^{\{2\}}\})}, P(x_1)^{(\{1\},\{\})}] \\
 & & & 2 & [\neg P(x_2)^{(\{2\},\{\})}, Q(x_2)^{(\{1,3\},\{\})}] \\
 & & & 3 & [P(f(a))^{(\{2,3\},\{P(x_1)^{\{1\}}\})}] \\
 & & & 4 & [\neg Q(a)^{(\{1,2,3\},\{\})}] \ \rangle \\
W_d & = & [& 1 & \langle P(x_1)^{(\{1\},\{P(f(a))^{\{3\}}\}))}, Q(x_2)^{(\{2\},\{\})}, \neg Q(a)^{(\{4\},\{\})}\rangle \\
 & & & 2 & \langle \neg P(x_2)^{(\{2\},\{\neg P(f(x_1))^{\{1\}}\})}, P(f(a))^{(\{3\},\{\})}, \neg Q(a)^{(\{4\},\{\})}\rangle \\
 & & & 3 & \langle \neg P(f(x_1))^{(\{1\},\{\})}, Q(x_2)^{(\{2\},\{\})}, P(f(a))^{(\{3\},\{\})}, \neg Q(a)^{(\{4\},\{\})}\rangle \ ]
\end{array}
$$

cuja prova seria dada pelos pares complementares $\{(\neg Q(a), Q(x_2)), (\neg P(x_2), P(x_1)), (\neg P(f(x_1)), P(f(a)))\}$ com a substituição resultante para eliminação dos caminhos $\{x_2/a, x_2/x_1, x_1/a\}$.

No entanto, devido à condensação, o par de literais $(\neg P(x_2), P(x_1))$ não apresenta nenhuma coordenada em comum. Segundo a r.e. 7, as tuplas da distribuição de W_c nas quais os literais de uma cláusula ocorrem diferem apenas por estes literais. Assim, como na forma disjuntiva o literal $\neg P(x_2)$ subsume o literal $\neg P(f(x_1))$, a cláusula dual resultante subsume todas as cláusulas duais (da forma não condensada) onde ocorre $\neg P(x_2)$ e os demais literais de mesma cláusula que $\neg P(f(x_1))$ (no caso $P(x_1)$ que ocorre na cláusula dual $\langle P(x_1)^{(\{1\},\{\})}, \neg P(x_2)^{(\{2\},\{\})}, P(f(a))^{(\{3\},\{\})}, \neg Q(a)^{(\{4\},\{\})}\rangle$ subsumida na forma condensada). Este caso de não intersecção das coordenadas pode ser definido como segue: dados l_1 e l_2 literais de mesma cláusula C_1 e l_3 literal de outra cláusula, C_2, se l_3 subsume l_1, então as coordenadas de l_2 e l_3 são disjuntas. Ou $\forall l_1 \in C_1, l_2 \in C_1, l_3 \in C_2, C_1 \neq C_2$ $(l_3 \succ l_1 \rightarrow (F_c(l_3) \cup F_c(Subs_c(l_3))) \cap (F_c(l_2) \cup F_c(Subs_c(l_2))) = \emptyset)$.

Como já dito anteriormente, uma coordenada subsumida de um literal pode ser utilizada para eliminação de caminhos se a constituição da forma disjuntiva condensada for preservada. Isto é o mesmo que dizer: se as cláusulas duais subsumidas forem também eliminadas (tornadas contraditórias). Uma maneira é garantir que a substituição de eliminação seja compatível com a substituição da subsunção, como apresentado anteriormente.

Porém, quando o literal subsunçor contradiz o literal com o qual não apresenta intersecção nas coordenadas, sabe-se que as cláusulas duais subsumidas são contraditórias, uma vez que contêm este par de literais complementares. Considerando-se os mesmos l_1, l_2 e l_3 tem-se: como $l_3 \succ_{\theta_1} l_1$, então $l_3 \in Subs_c(l_1)$. Para considerar $F_c(l_3)$ coordenadas subsumidas em l_1 na eliminação de caminhos com um literal l_4 complementar à l_1 com eliminação θ_2, pode-se proceder como anteriormente (considerando θ_1 como restrição) ou, se l_3 e l_2 são literais complementares com a substituição θ_3, pode-se combinar θ_3 à $\theta 2$. (Como l_2 e l_3 são complementares, as cláusulas duais subsumidas contém o par (l_3, l_2) e são tornada contraditória na instância θ_3. Como θ_3 provém da contradição de literal de mesma cláusula de l_1, então para considerar as coordenadas de $F_c(l_3)$, subsumidas em l_1, a substituição θ_3 deve ser não apenas compatível com a substituição de eliminação θ_2, mas deve ser combinada a esta e fazer parte da prova.) Para melhor entendimento, retornemos à fórmula do exemplo 43. Nela tem-se:

- $l_1 = \neg P(f(x_1))^{(\{3\},\{\neg P(x_2)^{\{2\}}\})}$, com subsunção $\theta_1 = \{x_2/f(x_1)\}$,
- $l_2 = P(x_1)^{(\{1\},\{\})}$,
- $l_3 = \neg P(x_2)^{(\{2\},\{\})}$, que contradiz l_2 com $\theta_3 = \{x_1/x_2\}$, e
- $l_4 = P(f(a))^{(\{2,3\},\{P(x_1)^{\{1\}}\})}$, que contradiz l_1 com $\theta_2 = \{x_1/a\}$.

Para definir as coordenadas eliminadas pelo par (l_1, l_4) tem-se de definir, para cada literal quais as coordenadas candidatas à eliminação:

- $\Omega(l_4) = \{(\{1,2,3\},\{\})\}$, por ser literal de cláusula unitária, todas as coordenadas de l_4 podem ser consideradas sem restrição;
- por apresentar coordenadas subsumidas, os membros de $\Omega(l_1)$ são:
  - $(\{3\},\{\})$, a coordenada 3, sem restrição, pois pertence à $F_c(l_1)$;
  - $(\{2,3\},\{x_2/f(x_1)\})$, as coordenadas 2 e 3, com restrição θ_1;
  - $(\{2,3\},(\circ\{x_1/x_2\},(\neg P(x_2),P(x_1)))$, as coordenadas 2 e 3, com a substituição θ_3, do par $(\neg P(x_2),P(x_1))$ a ser combinada na eliminação.

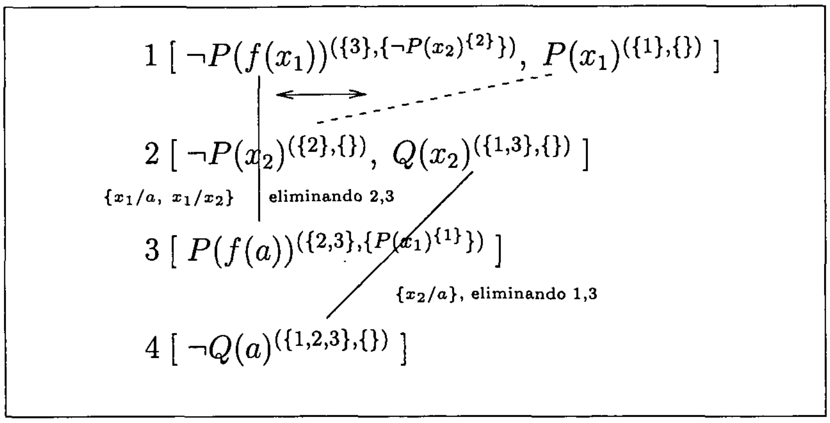

Figura 8.6: A refutação da fórmula do exemplo 43.

Para calcular os membros de $\mho(l_1, l_4)$, o conjunto de coordenadas subsumidas pelo par complementar, procede-se como anteriormente, porém considerando as substituições que devem ser combinadas:

- $\langle\{x_1/a\}, \{3\}, \{\}\rangle$, que elimina a coordenada 3 com θ_2 sem restrição,
- $\langle\{x_1/a\}, \{2,3\}, \{x_2/f(x_1)\}\rangle$, que elimina as coordenadas 2 e 3 com θ_2 com restrição θ_1, e
- $\langle\{x_1/a, x_1/x_2\}, \{2,3\}, (\neg P(x_2), P(x_1))\rangle$, que elimina as coordenadas 2 e 3 com $\theta_2 \circ \theta_3$, sem restrição mas associando o par (l_2, l_3) à eliminação (isto é, toda vez que esta tupla for considerada, a eliminação destas coordenadas deve-se aos pares (l_1, l_4) e (l_2, l_3)).

Resta-nos, agora, calcular $\mho$ para os demais pares complementares da fórmula:

- $\mho(P(x_1)^{(\{1\},\{\})}, \neg P(x_2)^{(\{2\},\{\})}) = \{\langle\{x_1/a, x_1/x_2\}, \{2,3\}, (\neg P(f(x_1)), P(f(a)))\rangle\}$, que associa o par (l_1, l_4) à eliminação,
- $\mho(\neg P(x_2)^{(\{2\},\{\})}, P(f(a))^{(\{2,3\},\{P(x_1)^{\{1\}}\})}) = \{\langle\{x_2/f(a)\}, \{2\}, \{\}\rangle\}$, e
- $\mho(\neg Q(a)^{(\{1,2,3\},\{\})}, Q(x_2)^{(\{1,3\},\{\})}) = \{\langle\{x_2/a\}, \{1,3\}, \{\}\rangle\}$

Dos conjuntos $\mho$ calculados, a tupla $\langle\{x_1/a, x_1/x_2\}, \{2,3\}, \{\}\rangle$ pode ser combinada com a tupla $\langle\{x_2/a\}, \{1,3\}, \{\}\rangle$, gerando a tupla $\langle\{x_1/a, x_1/x_2, x_2/a\}, \{1,2,3\}, \{\}\rangle$. Desta conclui-se que todos os caminhos podem ser eliminados simultaneamente, constituindo a refutação da fórmula. Esta refutação pode ser vista na figura 8.6.

Com a eliminação das coordenadas eliminadas por um par de literais complementares redefinida no cálculo de $\mho$, para ser utilizada para fórmulas condensadas a rotina *Goal* deve somente ser alterada de modo a respeitar, a cada par complementar, as restrições de substituições aplicadas e os pares associados à eliminação (se o elemento de $\mho$ considerado assim exigir).

8.3 Conclusão

Neste capítulo foi abordado o uso da forma disjuntiva condensada na inferência baseada em caminhos. Vamos resumir e discutir este uso e, para tal, salientamos alguns pontos:

- primeiramente foi considerada apenas a igualdade de literais para a condensação da fórmula. Esta condensação resulta em fórmulas que podem ser representadas na representação cruzada (não estendida) pois não necessita da representação das coordenadas subsumidas. Com isto a rotina *Goal* pode ser utilizada exatamente como apresentada. Esta condensação reduz a busca pela refutação da fórmula. Resta ainda confirmar se a rotina *Goal* mantém-se correta nesta condensação;

- no caso da condensação considerando a relação de subsunção entre literais, a representação da fórmula é mais complexa e deve ser estendida de forma a representar as coordenadas subsumidas. Mais que isto, a determinação dos caminhos eliminados por um par de literais é também bastante mais complexa. Porém, com a definição dos conjuntos Ω e $\mho$, esta determinação é correta e "parece" manter a completude da rotina *Goal* (esta completude deve ser mais profundamente estudada e comprovada);

- nesta condensação pode ocorrer o caso que denominamos a *não eliminação* de coordenadas por um par de literais complementares. Neste caso, este par de literais complementar torna-se *ligado, conectado* a outro par o que implica o uso simultâneo destes pares para a eliminação. Isto poderia, a primeira vista, parecer uma vantagem apresentada pela condensação, pois pode-se pensar: "muito bom, quando considero um par de literais estou, de fato considerando dois pares, e isto deve simplificar a busca pela refutação". "Jein" :) Na realidade esta conexão de pares já ocorre na fórmula. A diferença é que, com o uso da forma condensada, esta conexão torna-se explícita, é apenas um artifício que possibilita a consideração de pares complementares que não eliminam caminho algum. Este artifício é necessário pois o par em questão pode ser necessário à refutação e deve, portanto, ser considerado (como no caso do exemplo 43). Para entender o que esta conexão representa, consideremos algumas cláusulas da fórmula do exemplo 43.

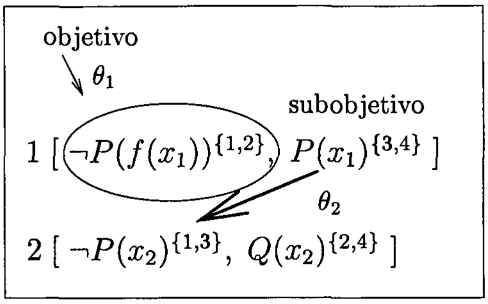

Figura 8.7: A conexão de pares do exemplo 43 via $\neg P(f(x_1))$.

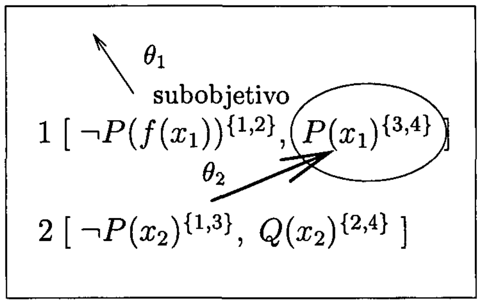

Figura 8.8: A conexão de pares do exemplo 43 via $P(x_1)$.

Na figura 8.7 é mostrado o caso da refutação da fórmula, onde o literal $\neg P(f(x_1))$ é utilizado para solucionar um objetivo l (no caso o literal $P(f(a))$) com uma substituição θ_1. Sendo assim, todos os demais literal da cláusula de $\neg P(f(x_1))$ são subobjetivos dependentes. No caso como o literal $P(x_1)$ é o único subobjetivo, e como o único literal complementar a ele é $P(x_2)$ com substituição θ_2, então toda vez que $\neg P(f(x_1))$ solucionar um objetivo com substituição θ_1, θ_2 deverá ser considerada (e o par $(P(x_1), \neg P(x_2)))$ como única solução do subobjetivo. Assim, de certa forma, os pares $(l, \neg P(f(x_1)))$ e $(P(x_1), \neg P(x_2))$ estão conectados, ligados um ao outro.

Na figura 8.8 é mostrado o caso inverso, onde o literal $P(x_1)$ é usado para solucionar com θ_2 o objetivo l (no caso $\neg P(x_2)$). Neste caso, o único subobjetivo dependente é $\neg P(f(x_1))$, cuja único literal complementar é $P(f(a))$ com substituição θ_1. Então, toda vez que $P(x_1)$ solucionar um objetivo com substituição θ_2, θ_1 deverá ser considerada (e o par $(P(f(a)), \neg P(f(x_1))))$ como única solução do subobjetivo. Assim, novamente, os pares $(l, \neg P(f(x_1)))$ e $(P(x_1), \neg P(x_2))$ estão, de certa forma, conectados, ligados um ao outro.

Então, esta conexão de pares não introduz vantagem alguma, uma vez que os mesmos pares seriam considerados pela rotina *Goal* se utilizada para fórmulas sem condensação;

- o fato da fórmula condensada conter um número menor de cláusulas duais também não introduz vantagem alguma pois, exceto o caso da condensação apenas por igualdade, todos os pares complementares considerados pela rotina *Goal* quando utilizada e fórmulas não condensadas, são também considerados para fórmulas condensadas. A diferença é que, normalmente, o número de coordenadas eliminadas por um par de literais complementares é menor em fórmulas condensadas.

A única vantagem que se pode considerar é quanto ao cálculo, propriamente dito, da forma disjuntiva condensada. Como o algoritmo *Dual* apresentado em capítulo anterior foi desenvolvido para calcular, de maneira eficiente, diretamente a forma dual condensada, em muitos casos, o cálculo da forma disjuntiva condensada é mais barata computacionalmente do que o cálculo da forma não condensada.

Com tudo isto, pode-se perguntar: para que usar a forma condensada de uma fórmula? Acreditamos que com a representação das coordenadas subsumidas, e a consideração destas, o método permanece correto. Mas ainda assim, para quê? O tratamento destas coordenadas, apesar de correto, é *confuso*. Ainda tem-se a não equivalência das instâncias não compatíveis com as subsunções consideradas na condensação. Pode ocorrer o caso onde a instância da prova é incompatível com a condensação e isto impossibilitar a refutação? Não sabemos, esta hipótese não foi estudada. (Porém, nunca conseguimos criar um exemplo onde ela ocorresse.)

Volta a pergunta: para quê? Tudo o que conseguimos até o momento foi reproduzir, com a forma condensada, a refutação que ocorreria naturalmente com a forma não condensada (e toda vez que isto não ocorreu, foi devido ao tratamento errôneo das coordenadas subsumidas). Não conseguimos, em nenhum caso, obter uma refutação mais simples e *correta* (ou pelo menos diferente) devido ao uso da forma condensada. O único caso foi com o uso da condensação simplificada (baseada em igualdade entre literais) e mesmo este, não afirmamos ser correto.

"If there's no meaning in it," said the King, "that saves a world of trouble, you know, as we needn't try to find any. And yet I don't know," he went on, spreading out the verses on his knee, and looking at them with one eye: "I seem to see some meaning in them, after all."

Lewis Carroll, Alice's Adventures in Wonderland, 1865

Capítulo 9

Além das coordenadas

Inesperadamente, a finalização desta tese, a princípio, foi bastante frustrante. Após tanto estudo — simplificação da forma disjuntiva, a relação desta com a forma conjuntiva, caminhos, coordenadas — chegar à conclusão que andou-se em círculos: pelo estudo das coordenadas, comprova-se que a estratégia apresentada pode ser aplicada sem o cálculo da forma disjuntiva da fórmula, o que torna sem sentido, para esta estratégia, a consideração desta forma – mesmo simplificada.

Porém, ao invés de "em círculo", poderíamos dizer que andamos "em espiral", pois não retornamos exatamente ao ponto de partida. A estratégia *Goal* pode em muito ser estudada e melhorada, e mesmo o estudo das coordenadas pode ser aplicado a outros problemas.

9.1 A rotina "descoordenada"

Seguindo a mesma linha de raciocínio do final do capítulo anterior, pode-se estender a pergunta "Para quê?" para o uso da forma disjuntiva, propriamente dita. O estudo foi bastante extenso: atribuímos coordenadas às formas normais, estabeleceu-se relações entre estas formas, detectou-se que as coordenadas representam caminhos em uma das formas e que estes equivalem às cláusulas da outra. Mas, para que? Que vantagem o uso simultâneo das duas formas normais de uma fórmula agrega à inferência? Como justificar o custo adicional do cálculo da forma disjuntiva?

Estendendo o que afirmamos no final do capítulo anterior: tudo o que conseguimos até o momento foi reproduzir, com o uso simultâneo das formas normais, a refutação que ocorreria naturalmente com apenas a forma conjuntiva da fórmula (e toda vez que isto não ocorreu, foi devido ao tratamento errôneo das coordenadas). Não conseguimos, em nenhum caso, obter uma refutação mais simples e *correta* (ou pelo menos diferente) devido ao uso simultâneo das formas. (Permanece como único, o caso do uso da condensação simplificada, que não sabemos ser correto.)

Consideremos a rotina *Goal* apresentada. Esta pode ser descrita, de uma forma geral, como: dado um literal do teorema (objetivo), sua eliminação dá-se ou (i) por um literal complementar, caso este par elimine todas as coordenadas (*gap*), ou (ii) por este literal complementar e pela eliminação dos demais literais da cláusula deste literal complementar. A rotina é repetida até que todos os literais do teorema sejam eliminados. Ou seja, a eliminação de um literal somente ocorre quando todas suas coordenadas (*gap*) forem contraditórias e a refutação somente ocorre quando todos os literais de uma cláusula forem eliminados, isto é, quando todas as coordenadas da fórmula forem contraditórias.

Como as coordenadas representam os caminhos da forma conjuntiva, um literal somente é eliminado quando todos os caminhos que passam por ele são constatados contraditórios. Conseqüentemente, uma fórmula somente é refutada quando todos os caminhos de sua forma conjuntiva são constatados contraditórios. De uma forma ou de outra, esta é a verificação efetuada por todos os métodos de prova. Até aqui, então nenhuma novidade introduzida pelo conceito de eliminação de coordenadas. A vantagem introduzida pelo uso das coordenadas poderia encontrar-se, então, no conceito de eliminação do *gap* e na forma que esta eliminação é detectada, uma vez que esta detecção encerra a busca pela eliminação de um literal.

Pela descrição da rotina, sabe-se que coordenadas do *gap* são eliminadas por ocorrerem simultaneamente em dois literais complementares e que, dado um objetivo (literal a ser eliminado), a total eliminação de seu *gap* (coordenadas deste literal) configura a sua solução. Na rotina *Goal* apenas dois casos são considerados solução para um objetivo l: ou (i) quando este contradiz um objetivo previamente considerado (verificado na rotina *Previous*) ou (ii) quando o *gap* é totalmente eliminado pela contradição com um literal *neg*. O primeiro caso já foi explicado: sempre que ocorrer a contradição com um objetivo prévio, todo o *gap* é eliminado (e portanto não precisa ser testado). Já no segundo caso, pela distribuição das coordenadas, o *gap* somente é totalmente eliminado quando *neg* é literal de cláusula unitária. Assim, o *gap* aqui também não é necessário.

O segundo caso deve ser melhor explicado. Em resumo, dado um objetivo l a cumprir, este somente é solucionado ou pela contradição à um objetivo superior na árvore de busca ou pela contradição com um literal $\neg l$ de uma cláusula C e soluções (compatíveis e simultâneas) dos demais literais da cláusula $(C-\{\neg l\})$. Pela distribuição dos literais, representada pelas coordenadas, comprova-se que neste caso, *gap* somente é eliminado quando $\neg l$ é o único literal de C $(C-\{\neg l\} = \emptyset)$: se $\neg l$ não é literal de cláusula unitária, então cada literal $l_n \in C - \{\neg l\}$ define um caminho (único) que passam por l_n e l, não contido em $\neg l$ e contidos em *gap* (uma vez que este caminho é um caminho que passa por l). (Salienta-se, novamente, que a eliminação de *gap* em cláusula não unitária pode acontecer no caso de fórmula condensada com a consideração apenas da igualdade entre literais, conforme mostrado no exemplo 41.)

Portanto toda a consideração de coordenadas poder ser retirada da rotina *Goal* sem prejudicar (ou mesmo modificar) seu comportamento. Para a retirada da consideração das coordenadas pela rotina *Goal*, deve-se proceder as seguintes alterações na rotina:

Goal$(W_c, Goals)$
1. Change variable names
2. $new \leftarrow \emptyset$
3. **for all** $C \in Goals$ **do**
   $\Delta \leftarrow W_c - \{C\}$
   $(solutions, theorems) \leftarrow \sum_{l \in C} Solutions(l, \Delta, \{\})$
   $(solutions, theorems) \leftarrow \bigotimes(solutions, theorems, C)$
   if $solutions \neq \emptyset$
      then Return (W_c is contradictory)
      else $new \leftarrow new \cup theorems$
4. $NewW_c \leftarrow \lfloor W_c \cup new \rfloor$
5. **if** $NewW_c = W_c$
      then Return (W_c is closed)
      else $Goal(NewW_c, new \cap NewW_c)$

no passo 3, retirar o *gap* ($F_c(l)$) da chamada da rotina *Solutions* e, no passo 4, retirar o cálculo da representação cruzada (que inclui o cálculo da forma disjuntiva);

Solutions$(Lit, W_c, Prev)$
1. $solutions \leftarrow Previous(Lit, Prev)$
2. $\Omega \leftarrow Complementaries(Lit, W_c)$
3. **if** $\Omega = \emptyset$
   then $theorems \leftarrow \{(\{Lit\}, \{\})\}$
   else $(solutions, theorems) \leftarrow \bigcup_{(neg,\theta) \in \Omega} Subgoals(Lit, neg, \theta, W_c, Prev)$
4. **if** $solutions \neq \emptyset$
   then $solutions \leftarrow Proj(solutions, Lit \cup Prev)$
5. Return $(solutions, theorems)$

retirar o *gap* dos parâmetros de entrada da rotina e das chamadas das rotinas *Previous*, *Complementaries* e *Subgoals*;

Subgoals$(Lit, neg, \theta, W_c, Prev)$
1. $L \leftarrow C(neg) - \{neg\}$
2. **if** $L = \emptyset$
    then $solutions \leftarrow \{\theta\}$
    else $\Delta \leftarrow \lceil W_c - \{C\} \rceil, P \leftarrow Prev \cup \{Lit\}$
      $(solutions, theorems) \leftarrow \sum_{l \in L} Solutions(l\theta, \Delta, P)$
      $(solutions, theorems) \leftarrow \bigotimes(solutions, theorems, L)$
      if $solutions \neq \emptyset$
        then $solutions \leftarrow Proj(Compatible(solutions, \theta), Lit \cup Prev)$
        else $theorems \leftarrow Compatible(theorems, \theta)$
3. Return $(solutions, theorems)$

retirar o *gap* dos parâmetros de entrada da rotina e da chamada da rotina *Solutions*; trocar o passo 1, que atualizava o *gap*, para atribuir a L os demais literais de mesma cláusula do literal *neg* e, no passo 2, trocar o teste $Gap = \emptyset$ por $L = \emptyset$ (*neg* é literal de cláusula unitária), conforme explicado anteriormente.

Conclui-se, assim, que a consideração da forma disjuntiva pela rotina *Goal*, é redundante (e, por conseguinte, desnecessária), uma vez que esta rotina pode realizar exatamente as mesmas refutações (percorrendo os mesmos caminhos) utilizando apenas a forma conjuntiva da fórmula. A versão sem coordenadas da rotina *Goal* foi implementada e comparada com sua versão primeiramente apresentada. O resultado desta comparação é apresentado na tabela 9.1. Nela comprova-se que o número de ciclos e teoremas gerados são os mesmos para as duas versões, como era de se esperar. A diferença de tempo de processamento (indicado nas respectivas colunas "seg"[1]) deve-se quase que exclusivamente ao cálculo da forma disjuntiva pela versão com coordenadas. Pode-se observar que esta diferença cresce com o tamanho da fórmula e, conseqüentemente, com o o aumento do número de ciclos necessários à sua refutação.

Terminamos aqui a apresentação do estudo sobre o uso simultâneo das formas normais de uma fórmula para inferência lógica.

[1]Tempo de "user run time"em segundos.

Fórmula			*Com coordenadas*			*Sem coordenadas*		
nome	*conj.*	*disj.*	*ciclos*	*teor.*	seg.	*ciclos*	*teor.*	seg.
Pelletier 44	7	64	1	0	0.24	1	0	0.15
Pelletier 48	7	24	3	18	642.75	3	18	119.79
Pelletier 55	12	288	1	0	22.52	1	0	9.66
Pelletier 58	7	24	1	0	32.91	1	0	14.49
TPTP COM001-1	11	36	1	0	11.02	1	0	4.05
TPTP COM002-1	19	36	3	4	104.92	3	4	19.87

Tabela 9.1: Refutações nas duas versões da rotina *Goal*.

9.2 Trabalhos futuros

Durante nosso estudo, diversos "detalhes" interessantes não receberam a devida importância por não serem parte prioritária do trabalho em questão. Muitas idéias surgiram e foram deixadas de lado, algumas sem importância e outras cujo estudo acreditamos promissor, como as que apresentamos agora.

9.2.1 A implementação

Para implementação da rotina *Goal*, nos preocupamos basicamente com os aspectos teóricos da rotina e não em eficiência computacional. Nesta, por exemplo, o paralelismo inerente à estrutura da rotina é apenas simulado. Seria interessante desenvolver uma melhor implementação da rotina e testá-la em toda a biblioteca de problemas *TPTP*. Este teste é uma espécie de "certidão de nascimento" de um método de prova para a comunidade de prova automática de teoremas pois, uma vez incluídos os resultados obtidos neste teste na página (web page) da biblioteca *TPTP*, o método "existe" para a comunidade. Não por atribuir existência a um provador automático de teoremas, este teste é importante por possibilitar a comparação de seu desempenho com o desempenho dos demais provadores pesquisados.

9.2.2 Relembrando caminhos

Durante a apresentação deste trabalho foi abordada diversas vezes a "eliminação de literais". Este conceito foi definido mas não lhe foi dada a devida ênfase. Acreditamos que o conceito de eliminação de literais pode introduzir um importante atributo a provadores automáticos de teoremas: memória, isto é, uma forma de reconhecer inferências já realizadas e reutilizá-las.

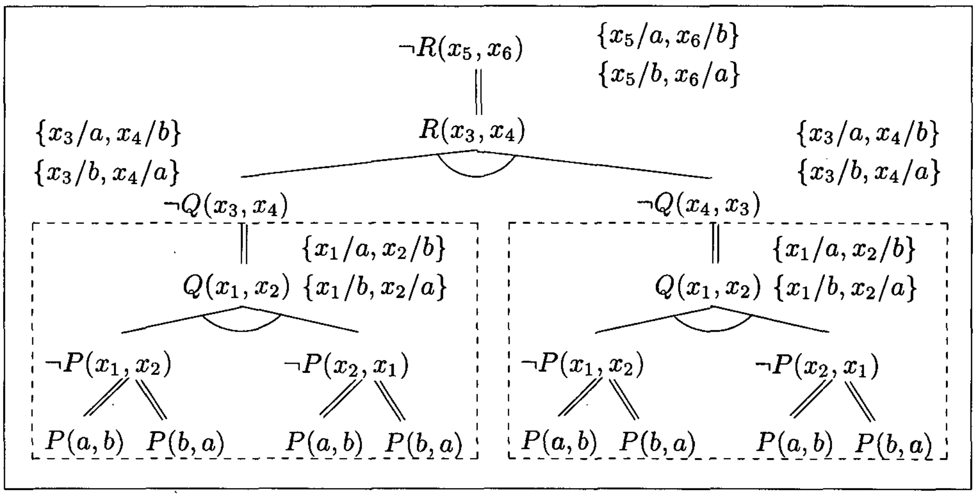

Figura 9.1: Uma refutação com soluções intermediárias repetidas.

Para exemplificar o que queremos dizer, considere a inferência apresentada apresentada na figura 9.1 relativa à fórmula contraditória:

$$
\begin{array}{rcl}
W_c & = & \langle\ [P(a,b)], \\
 & & [P(b,a)], \\
 & & [\neg P(x_1,x_2), \neg P(x_2,x_1), Q(x_1,x_2)], \\
 & & [\neg Q(x_3,x_4), \neg Q(x_4,x_3), R(x_3,x_4)], \\
 & * & [\neg R(x_5,x_6)]\ \rangle
\end{array}
$$

Nesta figura vê-se que o literal $Q(x_1, x_2)$ foi utilizado como solução dos literais $\neg Q(x_3, x_4)$ e $\neg Q(x_4, x_3)$. Para cada um destes literais, foram calculadas as soluções compatíveis para os subobjetivos $\neg P(x_1, x_2)$ e $\neg P(x_2, x_1)$. Na figura verifica-se que a árvore desenhada para solução de $\neg Q(x_3, x_4)$ e $\neg Q(x_4, x_3)$ é a mesma. Portanto, quando o literal $Q(x_1, x_2)$ é utilizado pela segunda vez como solução, a rotina poderia "saber" que seus subobjetivos dependentes são eliminados com a substituição $\{x_1/a, x_2/b\}$ ou com a substituição $\{x_1/b, x_2/a\}$, evitando assim o cálculo repetido daquela parte da árvore.

Porém, para realmente utilizar esta "memória" pelo menos um problema deve ainda ser resolvido: as soluções de um literal são restringidas por substituições já consideradas e pelo espaço de busca. As soluções previamente calculadas não representam um grande problema, uma vez que pode-se calcular as soluções possíveis dentro do espaço de busca, armazená-las e utilizar apenas as compatíveis com as substituições já consideradas.

Esta seria uma solução possível para esta restrição. A restrição do espaço de busca, porém, é um problema um pouco mais complexo: como o espaço de busca para soluções de um literal é restrito às cláusulas ainda não utilizadas em ramos superiores da árvore de busca, as soluções deste literal podem diferir dependendo da árvore de busca já percorrida. Pode também diferir de um ciclo para outro pois, com a inclusão de novos teoremas na fórmula, o espaço de busca para o cálculo de soluções pode ter sido alterado. O problema a ser resolvido é: dado um literal para o qual soluções foram calculadas com um espaço de busca Δ_1, como calcular novas soluções (ou remover algumas) dado um espaço de busca Δ_2.

Consideramos que o conceito de eliminação de um literal é muito importante por conferir, à árvore de busca percorrida durante uma inferência, uma estrutura. A definição desta estrutura que permite o armazenamento das informações processadas. Não temos conhecimento, até o momento, de nenhum método de prova que apresente alguma forma de memória. Mesmo "Otter" sendo um dos melhores, apresentando resultados (velocidade de processamento) impressionantes, não contém nada semelhante à memória. "Otter" é um sistema extremamente bem implementado e, se não apresenta esta facilidade, deve ser devido à *Resolução* propriamente dita, que não apresenta nenhuma estrutura em seu processo de inferência. Já em inferências realizadas pelo método das *Conexões* o conceito de eliminação de um literal seria facilmente adaptável, mas não conhecemos nenhum resultado ou pesquisa nesta direção.

Uma implementação "rudimentar" desta idéia foi realizada (considerando soluções já calculadas apenas no mesmo ciclo de inferência) e apresentou melhorias em tempos de processamento que justificam seu estudo. Por exemplo, o problema nomeado "mcnum" que com a versão sem coordenadas da rotina *Goal* é refutado em 46,64 segundos, com esta implementação rudimentar é refutado em 6,68 segundos.

9.2.3 O avesso das coordenadas

No desenrolar do trabalho foi dada ênfase nas coordenadas conjuntivas de uma fórmula, i.e. a posição de cada literal na forma disjuntiva. Pouco foi dito sobre as coordenadas disjuntivas, que refletem a ocorrência dos literais na forma conjuntiva. Uma possibilidade de estudo que achamos interessante é o uso, para lógica proposicional, destas coordenadas disjuntivas na forma conjuntiva. Pode parecer estranha esta possibilidade já que estaria-se utilizando as coordenadas referentes à posição dos literais na própria forma considerada, o que parece, à primeira vista, redundante. No entanto, ao indicar "as" ocorrências de um literal, as coordenadas indicam a existência ou não na fórmula de outro literal de mesmo símbolo proposicional e, com isto, esta informação não é redundante. Não pretendemos aqui desenvolver completamente este estudo, mas apenas apresentar parte dele já desenvolvida e que justifica nosso interesse.

As novas coordenadas

Considerando as coordenadas disjuntivas[2] de uma fórmula proposicional na representação cruzada estendida, pode-se estabelecer algumas relações estruturais expressas por estas coordenadas. Seja $F_d(l)$ o conjunto de coordenadas disjuntivas de um literal, $L(x)$ a coleção de literais de x, N_c o conjunto dos números das cláusulas de W_c e N_d o conjunto dos números das cláusulas duais de W_d, tem-se:

- como cada coordenada n contida nos literais de W_d refere-se à ocorrência destes literais na cláusula C_n de W_c, o conjunto de literais de W_d cujas coordenadas disjuntivas contêm n é o conjunto de literais da cláusula C_n: $\forall n \in N_d$ $(\{l \mid l \in L(W_d) \wedge n \in F_d(l)\} = L(C_n))$;
- a união das coordenadas disjuntivas dos literais de uma cláusula dual é o conjunto de números de cláusulas de W_c: $\bigcup F_d(l \in D) = N_c$;
- por ser W_d a forma disjuntiva condensada de uma fórmula, todo literal de uma cláusula dual ocorre "exclusivamente" em pelo menos uma cláusula de W_c. Ou seja, todo literal de uma cláusula dual possui pelo menos uma coordenada que não ocorre em nenhum outro literal desta mesma cláusula dual: $\forall l_i \in D, \exists n, \forall l_j \in D((n \in F_d(l_i) \wedge n \in F_d(l_j)) \rightarrow i = j)$;

que não são as únicas mas suficientes como exemplo. Estas relações, apesar de serem referentes à forma conjuntiva, definem a forma disjuntiva condensada. Acredita-se que as coordenadas podem ser "mapeadas" para a forma conjuntiva e estas relações utilizadas. Para este estudo utilizou-se uma nova representação da forma conjuntiva de uma fórmula na qual atribui-se, a cada literal, suas coordenadas disjuntivas (com a idempotência considerada), i.e. o número das cláusulas nas quais o literal ocorre, conforme mostra o exemplo 44. Por serem as únicas da representação, as coordenadas disjuntivas são chamadas simplesmente de *coordenadas*.

Exemplo 44 *Com as novas coordenadas, a forma conjuntiva da fórmula do exemplo 36 é assim representada:*

$$W_c = \langle \; 1 : [p^{\{1\}}, q^{\{1,2\}}], 2 : [r^{\{2,3\}}, q^{\{1,2\}}], 3 : [s^{\{3\}}, r^{\{2,3\}}] \; \rangle$$

[2]Que referem-se às posições dos literais na forma conjuntiva.

Estas coordenadas também definem relações entre os literais da fórmula que podem auxiliar o cálculo de sua forma disjuntiva condensada. Por exemplo, seja $F(l)$ as coordenadas de um literal, temos as seguintes relações:

- como W_c é condensada, todos os literais de todas as cláusulas ocorrem pelo menos uma vez na forma disjuntiva condensada W_d;
- como, pela r.e. 6, dada uma cláusula C de W_c, todas as cláusulas duais contêm pelo menos um literal de C, então a união das tuplas nas quais ocorre os literais de C é o conjunto de tuplas da distribuição de W_c;
- para todo par (l_1, l_2) de literais de mesma cláusula, a intersecção de suas coordenadas é não vazia: $\forall l_1 \in C_n, l_2 \in C_n(F(l_1) \cap F(l_2) \neq \emptyset)$;
- novamente, pela r.e. 7, as tuplas de l_1 e l_2, literais de mesma cláusula, são tuplas que diferem apenas por l_1 e l_2. Sendo l_3 um literal de cláusula diferente, para toda cláusula dual D_1, que contém o par (l_1, l_3), existe em W_d uma cláusula dual $D_2 = (D_1 - \{l_1\}) \cup \{l_2\}$. Assim quando $l_1 = l_3$, D_1 subsume D_2. O mesmo pode ser estendido para l_2 um literal de mesma cláusula de l_3. Então, a distribuição de l_1 com os demais literais das cláusula que contém literais iguais a l_1 produz tuplas subsumidas por aquelas produzidas pela distribuição de l_1 com seus iguais. Como as coordenadas de um literal referem-se a todas ocorrências de literais de mesmo símbolo proposicional, a distribuição de l_1 com as cláusulas contidas em suas coordenadas - $\forall C \in F(l_1)$ - dá origem a tuplas subsumidas;
- decorrente do item anterior, se l_2 ocorre em todas as cláusulas nas quais l_1 ocorre - $F(l_2) \subset F(l_1)$ - então toda tupla da distribuição de W_c contendo l_1 e l_2 são tuplas subsumidas;
- decorrente do item anterior, toda tupla da distribuição de W_c contendo os literais $\{l_1, l_2, \ldots, l_n\}$ e em todas as cláusulas nas quais l_1 ocorre, também ocorre outro literal da mesma tupla - $F(l_1) \subset \bigcup F(\{l_2, \ldots, l_n\})$ - é uma tupla subsumida;
- decorrente do item anterior, todo literal de $\{l_1, l_2, \ldots, l_n\}$ uma tupla da distribuição de W_c não subsumida, ocorre "exclusivamente" em pelo menos uma cláusula de W_c: $\forall l \in (l_1, \ldots, l_n), \exists n(n \in F(l) \rightarrow n \notin \bigcup F(\{l_1, l_2, \ldots, l_n\} - \{l\}))$;

O cálculo da forma disjuntiva condensada

Estas relações podem ser utilizadas para o cálculo eficiente da forma disjuntiva condensada. Considerando-se a fórmula do exemplo 44 e por estas relações, com o cálculo das tuplas da distribuição, não subsumidas, que contém o literal p da primeira cláusula e das que contém o literal q da primeira cláusula, calcula-se a forma condensada de W_d. Assim, sendo $N_c = \{1, 2, 3\}$ o número das cláusulas de W_c, a forma disjuntiva condensada W_d podes ser calculada como segue:

- tuplas de $p^{\{1\}}$ - as tuplas de p são todas tuplas, não subsumidas, que contêm os literais da segunda cláusula de W_c:
  - e $r^{\{2,3\}}$: a união de suas coordenadas $\{1\} \cup \{2, 3\} = N_c$ indica que (p, r) é uma tupla de W_d, pois toda outra tupla contendo (p, r) – no caso a tupla (p, r, s) – seria subsumida por esta. Então, $W_d = \{\langle p, r\rangle\}$;
  - e $q^{\{1,2\}}$: a inclusão das coordenadas de p nas coordenadas de q, $\{1\} \subset \{1, 2\}$, indica que toda tupla que contém o $(p, q, \ldots)$ é subsumida por outra que contém $(q, \ldots)$ e, portando, (p, q) não é subconjunto de nenhuma tupla de W_d;
  - como todos os literais da segunda cláusula resultaram em tupla de W_d ou tuplas subsumidas, nenhuma outra tupla de W_d contém o literal p e a busca pode ser encerrada;
- tuplas de $q^{\{1,2\}}$ - como q ocorre na primeira e segunda cláusulas, as tuplas de q são todas as tuplas, não subsumidas, que contêm os literais da terceira cláusula de W_c:
  - e $s^{\{3\}}$: a união de suas coordenadas $\{1, 2\} \cup \{3\} = N_c$ indica que (q, s) é uma tupla de W_d, então, $W_d = W_d \cup \{\langle q, s\rangle\}$;
  - e $r^{\{2,3\}}$: a não inclusão de nenhum conjunto de coordenadas indica não tratar-se de tupla subsumida, a intersecção de suas coordenadas $\{1, 2\} \cap \{2, 3\} = \{2\}$ indica tupla que representa cláusulas duais iguais ($\langle q, q, r\rangle$ e $\langle q, r, r\rangle$) e a união de suas coordenadas $\{1, 2\} \cup \{2, 3\} = N_c$ indica que (q, r) é uma tupla de W_d e, portanto, $W_d = W_d \cup \{\langle q, r\rangle\}$;
  - pelo mesmo motivo anterior a busca pode ser encerrada;
- o cálculo resulta em $W_d = [\langle p, r\rangle, \langle q, s\rangle, \langle q, r\rangle]$ que é a forma disjuntiva condensada da fórmula.

O procedimento aqui demonstrado para cálculo da forma disjuntiva condensada de uma fórmula em forma conjuntiva W_c, é sistematizado pela rotina $Transf_d(W_c)$ definida como segue:

$$Transf_d(\Delta) = \begin{cases} \emptyset & se\ \Delta = \emptyset \\ \{Tuples(l_1^{C_m}) \cup \ldots \cup Tuples(l_n^{C_m})\} & \text{caso contrário} \end{cases}$$

$$Tuples(l) = \begin{cases} \emptyset & se\ \beta \subset F(l), \text{senão} \\ \{\langle l \rangle\} & se\ \alpha \subset F(l), \text{senão} \\ \{\langle l \rangle \cup \gamma \mid \gamma \in \Gamma\} & \text{caso contrário} \end{cases}$$

onde $\alpha = \{n \mid C_n \in \Delta\}$, $\beta = N_c - \alpha$ e $\Gamma = Transf_d(\Delta - \{C_n \mid n \in F(l)\})$ e C_m uma cláusula qualquer de Δ.

Esta rotina calcula o mesmo que a rotina *Dual* apresentada anteriormente. Na realidade a rotina *Dual* já utiliza estas coordenadas e algumas destas relações para o cálculo da transformação dual. A diferença entre estas rotinas é que $Transf_d$ não utiliza o hipercubo como estrutura subjacente, mas procede a transformação considerando uma cláusula da fórmula de cada vez. Em alguns casos este procedimento pode ter vantagens sobre o procedimento realizado pela rotina *Dual*: no caso de fórmulas com muitos símbolos proposicionais e poucas cláusulas, o hipercubo associado apresenta um grande número de dimensões, o que pode prejudicar o funcionamento da rotina *Dual* pelo processamento de propagações entre os vértices; e no caso de fórmulas com poucos símbolos proposicionais e muitas cláusulas, o funcionamento da rotina *Dual* pode ser prejudicado pelo excesso de processamento em cada órbita. A rotina $Transf_d$ apresenta a vantagem do processamento da subrotina *Tuples* poder ser independente para cada literal e poder ser processado em paralelo. Com isto, a distribuição do processamento pode ser feita baseada no número de literais de uma cláusula. Para uma melhor comparação, deve-se realizar um estudo mais detalhado.

Os modelos

Pode-se também utilizar estas coordenadas para o cálculo dos modelos de uma fórmula proposicional. Considere a fórmula:

$$\begin{aligned} W_c = \langle\ & 1 : [p^{\{1,3\}}, q^{\{1,2\}}], 2 : [\neg p^{\{2,4\}}, q^{\{1,2\}}], \\ & 3 : [p^{\{1,3\}}, \neg q^{\{3,4\}}], 4 : [\neg p^{\{2,4\}}, \neg q^{\{3,4\}}]\ \rangle \end{aligned}$$

Se utilizarmos a rotina $Transf_d$ nesta fórmula veremos que as únicas cláusulas duais geradas são $\langle p, \neg p \rangle$ e $\langle q, \neg q \rangle$ ambas contraditórias, o que determina a refutação da mesma. A transformação desta fórmula pela rotina $Transf_d$ dá-se quase que imediatamente, pois pelas coordenadas dos pares $(p^{\{1,3\}}, \neg p^{\{2,4\}})$ e $(q^{\{1,2\}}, \neg q^{\{3,4\}})$ é fácil detectar que cada par é, por si, uma cláusula dual (já que a união de suas coordenadas é igual ao conjunto de cláusulas N_c). Sendo assim, para o cálculo dos modelos, a rotina $Transf_d$ poderia ser modificada de forma a procurar primeiramente as cláusulas duais que seriam contraditórias e evitar a sua geração. Com isto, no final do processamento, esta rotina retornaria apenas as cláusulas duais não contraditórias, ou seja, os *modelos* da fórmula. A ausência de modelos configuraria a refutação da fórmula.

O cálculo de modelos de fórmulas proposicionais é um problema cuja solução é intensamente pesquisada por outra área de pesquisa, conhecida como "Sat-n". Nesta área, é estudada a determinação da satisfazibilidade de uma fórmula proposicional. Este estudo encontra um grande número de aplicações, uma vez que vários problemas são especificados em lógica proposicional, como por exemplo, a simplificação de circuitos lógicos, dentre outros.

Talvez mereça ser estudada a possibilidade de extensão destas coordenadas para fórmulas da lógica de primeira ordem. Talvez também, aquele caso do exemplo 41, cuja forma condensada foi calculada considerando apenas a igualdade dos símbolos proposicionais, se correto, possa ser adaptado para o uso destas coordenadas e simplificar, assim, a refutação.

9.3 Conclusão geral do trabalho

Chegamos ao final deste trabalho. A idéia original que lhe deu início foi o desenvolvimento de uma estratégia para o método Dual. Neste método, inferências lógicas são realizadas através de sucessivas transformações da fórmula em suas formas normais conjuntiva e disjuntiva e retiradas de cláusulas duais contraditórias. O método apresenta pelo menos uma característica muito interessante: o não uso de regras de inferência explícitas. Porém, o método "puro" como apresentado no capítulo 5 tem normalmente um desempenho bastante ruim, equivalente à estratégia da Resolução *level saturation*. Devido à isto, iniciamos o estudo de uma estratégia que inferisse eficiência ao método Dual.

Nossa idéia para restringir as inferências realizadas foi considerar, além das contradições explicitadas pela forma disjuntiva, a estrutura da fórmula na forma conjuntiva; idéia esta baseada na representação cruzada já utilizada pelo algoritmo de transformação dual (apresentado na seção 4.4 do capítulo 4). Também devido ao uso das coordenadas pelo algoritmo e no intuito de simplificar a inferência, foi escolhido o utilizar a forma disjuntiva condensada da fórmula.

Seguindo nosso plano inicial, passamos a basear nosso estudo na representação cruzada das formas normais condensadas de uma fórmula. Tínhamos uma idéia de como a inferência poderia ser restringida. Com o passar do tempo deparamo-nos com diversas sutilezas apresentadas pelo cálculo desta condensação. Foram tempos difíceis, poderíamos dizer. Vasto é o material encontrado na literatura referente à condensação de fórmulas em forma conjuntiva e é praticamente inexistente o material referente à forma disjuntiva. Bom, "arregaça-se as mangas" e vai-se ao trabalho! Se nada encontramos na literatura, vamos desenvolver este estudo. Para o estudo da condensação da forma disjuntiva, foi estudada primeiramente a relação entre as formas normais para posterior adaptação do cálculo da condensação da forma conjuntiva para a forma disjuntiva. Este estudo foi bastante aprofundado e deu origem à diversas publicações, como demonstrado no capítulo 4.

Com o estudo da condensação das formas normais concluído, passamos a nos dedicar novamente ao desenvolvimento da estratégia. As sutilezas apresentadas para o correto tratamento das coordenadas subsumidas da representação cruzada estendida de uma fórmula foram tantas, que decidimos estudar o que realmente estas coordenadas representam. Para saber o que as coordenadas subsumidas representam, estudamos primeiramente o que representam as coordenadas não subsumidas. Para tal estudamos as coordenadas da representação cruzada de uma fórmula, isto é, coordenadas que representam a distribuição dos literais de uma fórmula em suas formas conjuntiva e disjuntiva não condensada. Neste estudo definimos diversas relações estruturais existentes entre estes literais e representadas pelas coordenadas.

Conhecidas as relações representadas pelas coordenadas da representação cruzada, desenvolveu-se uma estratégia com as características que tínhamos em mente desde o início: basear-se em eliminação de caminhos e de literais. Este estudo deu origem à rotina *Goal* como apresentada no capítulo 7. O passo seguinte, naturalmente, foi adaptar o estudo daquelas relações para a forma disjuntiva condensada e, finalmente, desenvolver uma estratégia que utilizasse simultaneamente as formas normais condensadas de uma fórmula. O estudo desenvolvido nesta direção, apresentado no capítulo 8, mostrou-se tão complicado que foi inevitável a pergunta "Para que usar as coordenadas subsumidas?". A busca pela resposta à esta pergunta levou-nos a um resultado inesperado: o uso da forma disjuntiva condensada não apresenta nenhuma vantagem sobre o uso da forma não condensada, pelo contrário, apenas introduz complicações extras à correta utilização das coordenadas.

Impossível também foi evitar estender a mesma pergunta "Para quê?" para o uso das coordenadas propriamente ditas e, conseqüentemente, para o uso simultâneo das duas formas normais de uma fórmula. Este resultado foi ainda mais inesperado: não conseguimos encontrar nenhuma justificativa para este uso. Pelo estudo das relações estruturais expressas pelas coordenadas, conclui-se que seu uso não introduz vantagem alguma; a inferência processada pela rotina *Goal* pode ser simulada "exatamente" considerando apenas a forma conjuntiva da fórmula (conforme apresentado no início deste capítulo).

"Se não tem sentido, isto poupa um mundo de trabalho, uma vez que não é preciso procurar um" escreveu Lewis Carroll[3]. Esta foi a primeira sensação ante aos resultados encontrados neste estudo: tanto estudo para nada. "Mas eu acho que vejo algum sentido, depois de tudo", continua a citação e, de fato, não acreditamos ter sido, este estudo, sem sentido. Conforme mostrado na seção 9.2, mesmo que o objetivo inicial não tenha apresentado o resultado esperado, muito do que foi estudo pode ser aproveitado.

O estudo das formas normais e sua condensação é, em si, um resultado que pode ser aproveitado, pois a simplificação de fórmulas lógicas é interessante a todo problema expresso em lógica. O estudo das coordenadas da representação cruzada, que considera ambas formas normais de uma fórmula, pode ser aproveitado e "mapeado" para o uso de apenas uma das formais. Apresentamos alguns resultados com estas novas coordenadas (que chamamos "ao avesso") e acreditamos que muito mais pode ser estudado nesta direção. E, por fim a estratégia desenvolvida que é, por si, um resultado que justifica todo o estudo realizado.

Tende-se, na área, a chamar de estratégia o que na realidade é um método, ou um cálculo, de inferência lógica. Tende-se a falar, por exemplo, no método da Resolução e suas estratégias (*set-of-support*, *level-saturation*, etc.) quando, na realidade, cada uma destas "estratégias" define um método baseado na Resolução. Lembramos que um método é composto pela linguagem lógica, um conjunto de regras de inferência e uma estratégia de aplicação destas regras. Assim, a rotina *Goal*, como apresentada no início deste capítulo, define um método de inferência lógica (cuja completude ainda precisa ser comprovada). Este método apresenta características bastante interessantes:

- é um método híbrido, que combina características que definem a família da Resolução com características que definem a família da Conexão. Acreditamos que este método define uma nova família de provadores automáticos de teoremas por não poder ser incluído em nenhuma das famílias existentes;

- a regra de inferência utilizada é baseada no conceito de eliminação de literais,

[3]Em "As Aventuras de Alice no País das Maravilhas", no trecho transcrito na citação na página inicial deste capítulo

completamente novo na área. Este conceito confere ao processo de inferência uma estrutura que permite a reutilização de inferências já realizadas. Esta característica merece ser estudada pois acreditamos ser extremamente importante por poder (potencialmente) resolver um grande problema enfrentado por todos os provadores automáticos de teoremas. Por mais que se esforce em otimizar a realização de inferências durante um processo de prova, muitas vezes não é possível torná-lo simples ou imediato. Em muitos casos, a prova requer um grande número de inferências, é uma característica inerente à formula. Porém, principalmente nestes casos, grande é o número de inferências repetidas e, é exatamente neste caso, que acreditamos ser importante à um provador automático de teoremas, apresentar a habilidade de "relembrar" inferências já utilizadas. É interessante que não temos conhecimento de nenhum trabalho nesta direção. Mesmo o provador de teoremas "Otter" pode ser considerado completamente "desmemoriado". (Mas é rápido, executa quinhentas vezes a mesma coisa, e nem se percebe!);

- o método parece apresentar outra importante característica: *confluência*. Um método é confluente quando, independente do caminho percorrido, as mesmas provas são encontradas. Mais tecnicamente, quando os mesmos valores são encontrados para as variáveis do teorema. Por exemplo, dada a cláusula $[P(a), P(b)]$ e o teorema negado $[\neg P(x)]$, um método confluente encontra, na refutação, os valores "a" e "b" para a variável x do teorema. Normalmente, os métodos não são confluentes, apenas preocupam-se em encontrar uma prova. A confluência pode, no entanto, ser desejada para algumas aplicações. Nenhum estudo foi desenvolvido nesta direção e, por isto, este assunto não foi abordado.

E o método dual continua como originalmente proposto, sem uma estratégia que lhe atribua eficiência ... mas isto é assunto para outra tese!

Apêndice A

Créditos e Agradecimentos Formais

Quer-se, neste capítulo, agradecer às pessoas e entidades colaboradoras, atribuir autoria e indicar publicações prévias de partes deste trabalho. Grande parte do trabalho aqui apresentado foi escrito pela autora exclusivamente, mas algumas partes do texto foram adaptadas ou de artigos previamente publicados pela autora e pelo Prof. Guilherme Bittencourt ou do outras fontes, o que torna necessário um agradecimento explícito. A autora também gostaria de agradecer toda a contribuição recebida para tornar possível a execução deste trabalho.

Seguindo um conselho dado, certa vez, por um professor — "Primeiro a barriga, depois a filosofia!" (adaptado do original de Bertold Bretch "Primeiro a barriga depois a moral") — inicia-se agradecendo a ajuda financeira recebida:

- ao CNPQ pela bolsa de estudo recebida por quatro anos (1997 - 2001), processo número 143359/97-5, sem a qual seria financeiramente impossível a realização deste trabalho;

- à Capes pela ajuda financeira recebida para a participação no projeto *Métodos Formais para Recuperação de Conhecimento em Rede e Bancos de Dados Distribuídos* (CAPES/DAAD/PROBAL número 060/98), sob coordenação dos professores Guilherme Bittencourt (Brasil) e Jaques Calmet (Alemanha). A participação neste projeto possibilitou, durante três anos consecutivos (1998 - 2000), a visita da autora ao Instituto para Algoritmos e Sistemas Cognitivo da Universidade de Karlsruhe, Alemanha (e, principalmente, à maravilhosa biblioteca desta universidade).

O trabalho foi desenvolvido nas instalações do *LCMI/DAS - Laboratório de Controle e Micro-Informática do Departamento de Automação e Sistemas* da *UFSC - Uni-*

versidade Federal de Santa Catarina e nas instalações do Instituto para Algoritmos e Sistemas Cognitivo da Universidade de Karlsruhe. Agradeço aos colegas e professores de ambas universidades.

Este trabalho é totalmente baseado em idéias e trabalhos do Prof. Guilherme. De fato é bastante difícil separar qual parte deste trabalho é seu ou da autora. Grande parte do trabalho foi desenvolvido em conjunto, tanto idéias como implementação. Poderia-se dizer que este trabalho é um estudo, realizado pela autora, sobre as idéias do Prof. Guilherme referentes ao método dual e sua estratégia. Vejamos, capítulo a capítulo:

3 Lógica e IA: grande parte deste capítulo foi extraída da dissertação de mestrado da autora, uma vez que o assunto deste trabalho é uma continuação desta. Agradecimento especial ao Prof. Guilherme pela autorização dada à autora de praticamente copiar partes de seu livro *Inteligência Artificial Ferramentas e Teorias* para compor as subseções intituladas *Um pouco de teoria* e *Método de Tableaux*;

4 Duas formas de dizer: este capítulo foi praticamente todo adaptado dos artigos apresentados nos LAPTEC e JAR, de autoria do Prof. Guilherme e da autora. A subseção intitulada *A simplificação propriamente dita* (referente à simplificação da forma disjuntiva) é resultado de estudos realizados pela autora, posteriores àquelas publicações. A seção relativa ao algoritmo dual é um resumo do que encontra-se na dissertação de mestrado da autora;

5 Eliminando o impossível: o método dual é de autoria exclusiva do Prof. Guilherme. O texto, porém, e a forma de explicar o método, é da autora. O texto foi baseado em sua dissertação de mestrado, mas quase que totalmente reescrito e melhorado para este trabalho;

6 Caminhos e coordenadas: aqui começa, de fato, o estudo realizado exclusivamente pela autora. A representação cruzada, a eliminação de caminhos e de literais são idéias do Prof. Guilherme. O estudo feito, as relações estruturais estabelecidas e utilizadas em todo o restante do trabalho, no entanto, são de responsabilidade da autora;

7 Acelerando o passo: a idéia básica da estratégia, isto é, a consideração da eliminação de literais, como já dito, é do Prof. Guilherme. Contudo, a estratégia *Goal* propriamente dita é uma modificação, feita pela autora, na estratégia apresentada no artigo *A Proof Strategy Based on a Dual Representation.* A definição da estratégia *Goal* bem como sua implementação é de responsabilidade da autora;

8 Simplificando(?): novamente, o uso da forma disjuntiva condensada e a conseqüente extensão da representação cruzada (com a representação das coordenadas subsumidas) são idéias do Prof. Guilherme. O estudo apresentado, no entanto, e a conclusão obtida é de responsabilidade da autora; e, por fim

9 Além das coordenadas: este capítulo é de total responsabilidade da autora.

Por fim, uma breve apresentação de *Lewis Carrol* que, com sua obra, também contribuiu para este trabalho:

> *Lewis Carrol* (pseudônimo do Rev. Charles Lutwidge Dodgson) foi professor de matemática na Universidade de Oxford, mas ele é mais conhecido como o autor de *Alice's Adventures in Wonderland* (1865) e *Through the Looking-Glass* (1871), bem como de *The Hunting of the Snark* (1876). Escreveu também *Symbolic Logic* e *Game of Logic*, não tão conhecidos mas não menos interessantes.

Por toda contribuição recebida, muito obrigada.

Apêndice B

Pequena história da lógica contemporânea

Uma matéria bastante interessante, escrita pelo Prof. Newton C.A. da Costa, foi publicada no caderno *FOLHA Mais!* do jornal *Folha de São Paulo*, na edição de domingo, 04 de agosto de 2002. Achamos interessante incluí-la pela relevância dos tópicos abordados, relacionados com o contexto deste trabalho. Agradecemos ao Prof. Newton a autorização fornecida para aqui reproduzir esta matéria.

A lógica é disciplina que tem história curiosa. Sendo, como sustenta a maioria dos especialistas, criação de Aristóteles (384-322 a.C.), permaneceu praticamente imutável durante dois milênios. Pensadores como Kant (1724-1804) achavam que nada se havia feito de essencial, em lógica, depois do grande filósofo grego.

Seu objetivo era o estudo da inferência válida (algumas vezes incluindo também a inferência dita indutiva, não válida, porém possuindo certo caráter verossímil), de um prisma formal. Na inferência válida, de premissas verdadeiras chega-se, sempre, a conclusões verdadeiras. As regras da lógica, devidamente utilizadas, assegurariam isso. Portanto, também se sustentava que ela era uma disciplina devotada ao raciocínio formalmente correto. Tal concepção ainda perdura em variados centros de ensino, sobretudo na América Latina.

Todavia a história duplamente milenar da lógica começou a mudar, a partir da segunda metade do século 19 e durante todo o século 20. Com efeito, ela sofreu uma assombrosa transformação que merece ser melhor conhecida, mormente por filósofos e cientistas. Para nós, aqui, a lógica consiste naquilo que os lógicos profissionais realizam. Trata-se da matéria que é cultivada nos núcleos importantes de investigação e que figura, em boa parte, nas revistas técnicas conceituadas, em nível internacional, tais como "The Journal of Symbolic Logic", "The Journal of Applied Non Classical Logic" e "Logique et Analyse".

Ela deixou de ser tão somente a ciência das formas válidas de raciocínio, embora a teoria da argumentação ainda pertença ao campo de suas aplicações. No momento, ela versa sobre determinadas estruturas abstratas, que podemos denominar de sistemas lógicos, análogos, em espírito, às estruturas da álgebra ou de outros ramos da matemática.

Na lógica estritamente pura, tais estruturas são consideradas de modo abstrato e amplo, juntamente com outras estruturas de índole mais matemática, embora fundamentais para a compreensão das primeiras. O lógico trata de sistemas lógicos diversificados, de conformidade com sua relevância intrínseca ou pelo seu significado no tocante às aplicações (em filosofia, nos fundamentos das teorias matemáticas, em física etc.).

Além disso, há a lógica aplicada, que se volta para as aplicações dos sistemas e métodos lógicos em todas as áreas do conhecimento. Existe determinada semelhança entre lógica e geometria: na geometria pura, estudam-se diversas estruturas geométricas (geometrias afim, euclidiana, riemanniana, finita etc.), enquanto que na geometria aplicada voltamo-nos para as estruturas geométricas com ênfase em aplicações mais ou menos precisas, como nos casos da geometria do espaço de Minkowski em relatividade restrita ou da geometria de Riemann em relatividade geral.

Facilmente se fará uma idéia, não obstante vaga, do abismo existente entre a visão tradicional da ciência em apreço e aquilo que os lógicos profissionais atualmente fazem, citando-se alguns dos assuntos dos quais se ocupam esses últimos: teoria geral da recursão, teoria dos modelos, álgebras cilíndricas, teorias de conjuntos não-clássicas, ultraprodutos, teoria intuicionista das sequências de livre escolha, teoria de Galois generalizada, "forcing" de Cohen e "forking" de Shelah. Para um historiador afeito ao tema, três são os traços principais da nova lógica que, como já deve ter ficado claro, transfigurou-se nos últimos 150 anos: 1) extraordinário desenvolvimento técnico; 2) aparecimento das chamadas lógicas não-clássicas, que complementam ou se afastam daquela batizada de clássica, a qual se inspira em pressupostos da tradição aristotélica, embora ampliando-a de maneira quase incrível; 3) a eclosão de variadas e numerosas aplicações, em quase todos os domínios do saber, acima de tudo em tecnologia (hoje se ministra a disciplina em escolas de engenharia e em todos os ramos do ensino da computação).

Ciência nobre

Há muita coisa a discutir no tocante à evolução dessa ciência. Salientemos, apenas, que ela, deixando de ser absolutamente trivial sob os ângulos técnico e matemático, converteu-se em ciência na qual se edificaram as mais profundas teorias, comparáveis às da matemática. Com efeito, tudo isso se pode confirmar considerando alguns exemplos interessantes, como os teoremas de Gödel, a teoria geral da recursão e a teoria de Galois generalizada. As indagações de Kurt Gõdel (1906-78) enquadram-se entre as mais notáveis realizações da história da cultura, tendo repercussões em todas as manifestações do saber e mudando os próprios paradigmas da lógica e da matemática. Por seu turno, a teoria da recursão pode ser encarada como teoria abstrata das máquinas computacionais, sendo os computadores com os quais estamos acostumados produtos físicos que exemplificam as máquinas teóricas da teoria em apreço (máquinas de Turing, algoritmos de Markov, autômatos finitos etc.). Da teoria geral e abstrata podemos derivar vários resultados sobre os computadores comuns. Por curiosidade, mencionaremos uma consequência referente a quaisquer vírus: cada categoria de vírus requer métodos "antivirais" específicos. Uma das teorias de enorme riqueza da álgebra é a teoria de Évariste Galois (1811-32). Ela se refere a certas estruturas algébricas, mas hoje foi estendida para abranger os sistemas lógicos. Ela nos conduz a uma imagem unificada da lógica, inclusive das não-clássicas, e estabelece novos vínculos entre essa ciência e a matemática. Do exposto, percebe-se o caráter altamente técnico da lógica contemporânea. Um aspecto pouco entendido e apreciado relaciona-se ao surgimento, acima de tudo a partir do início do século precedente, de lógicas diferentes da chamada clássica. Presentemente, quando se fala em lógica, torna-se necessário que se explicite de qual delas estamos tratando.

Transformação de paradigma

Assim, exemplificando, em certas questões de matemática qualificada de construtiva, deve-se recorrer a uma lógica divergente da clássica, ou seja, à chamada lógica intuicionista; em outros contextos, como os da mecânica quântica, parece conveniente lançarmos mão de uma das lógicas denominadas quânticas. Em tópicos de química e de genética, tem-se apelado para o cálculo lamba e a lógica combinatória, que não pertencem, propriamente, à classe das modalidades clássicas em acepção estrita.

Em síntese, há lógicas variadas como há geometrias distintas. É habitual conceber a criação das geometrias não-euclidianas como uma das máximas transformações de paradigma da ciência.

Pois bem, dado que a lógica se mostra mais básica do que a geometria na hierarquia do conhecimento, decorre que o câmbio correspondente de paradigma na ciência de Aristóteles possui significado singular. Este fato nos leva, sem dúvida, a uma renovação da própria idéia de ciência e provoca debates epistemológicos delicados.

A lógica de hoje encontrou aplicações em praticamente todos os domínios do saber: em filosofia, em filosofia da ciência, em linguística, em matemática (por exemplo, utilização de teoria de modelos em álgebra), em química, em biologia, em direito, em psicologia etc.

Todavia o que mais pode surpreender o não-especialista são as aplicações tecnológicas da nova lógica. Aplicações, exemplificando, da lógica "fuzzy" em engenharia, da paraconsistente em engenharia de produção e no controle de tráfego, da anotada em robótica e no reconhecimento automático de assinaturas em bancos, e da clássica em computação e programação. Outros setores nos quais essa ciência recentemente encontrou aplicações são os seguintes: sistemas especialistas para diagnóstico médico, fabricação de máquinas — por exemplo, geladeiras —, administração de empresas e teoria da computação flexível.

A história evidencia e a exposição anterior, pelo menos em linhas esquemáticas, comprova que a lógica origina muito problemas filosóficos (natureza das noções lógicas, conexões entre ela e a matemática, caráter eterno ou provisório das leis lógicas, realismo nas ciências lógico-matemáticas etc.).

Em virtude disso, além da lógica "tout court", existe a filosófica (filosofia da lógica), devotada à sua análise filosófica, tema de grande atualidade. O notável lógico e filósofo inglês A.N. Whitehead (1861-1947) afirmava que a lógica moderna estava para a tradicional assim como a matemática do século passado para a aritmética das tribos primitivas. Essa afirmação de Whitehead se mostra, em nossa época, como inteiramente justificável.

Newton C.A. da Costa é professor no departamento de filosofia da Faculdade de Filosofia, Letras e Ciências Humanas da USP e professor de fundamentos da computação e lógica da Unip (Universidade Paulista). É autor de, entre outros, "O Conhecimento Científico" (Discurso Editorial).

Apêndice C

Fórmulas utilizadas

sua maioria, como já dito, as fórmulas utilizadas nas tabelas 7.3 do capítulo 7 e 9.1 do capítulo 9, foram retiradas do artigo *Seventy-five Problems for Testing Automatic Theorem Provers* [29] de *Francis Jeffry Pelletier.* Estas fórmulas também fazem parte da biblioteca de problemas *TPTP.* Apenas algumas fórmulas utilizadas fazem parte apenas desta biblioteca.

Para cada fórmula apresentada são dados: o número do problema (fórmula) e o número de pontos atribuídos ao mesmo no referido artigo de *Pelletier*, o nome da fórmula na biblioteca *TPTP* e a fórmula propriamente dita em forma normal conjuntiva. O teorema a ser provado, se indicado, é seguido por um "*". Todas as fórmulas aqui apresentadas são contraditórias.

Pelletier: 21	$[\neg P, F(a)]$,
Pontos: 5	$[\neg F(b), P]$,
TPTP: SYN051-1	$[P, F(x_0)]$, *
	$[\neg F(x_1), \neg P]$ *

Pelletier: 22	$[P, \neg F(x_0)]$,
Pontos: 3	$[F(x_1), \neg P]$,
TPTP: SYN052-1	$[F(x_2), P]$,
	$[\neg P, \neg F(a)]$,
	$[F(x_3), \neg F(a)]$

Pelletier: 24
Pontos: 6
TPTP: SYN054-1

$[\neg S(x_0), \neg Q(x_0)],$
$[\neg P(x_1), Q(x_1), R(x_1)],$
$[P(a), Q(b)],$
$[\neg Q(x_2), S(x_2)],$
$[\neg R(x_3), S(x_3)],$ *
$[\neg P(x_4), \neg R(x_4)]$ *

Pelletier: 25
Pontos: 7
TPTP: SYN055-1

$[P(a)],$
$[\neg F(x_0), \neg G(x_0), \neg R(x_0)],$
$[\neg P(x_1), F(x_1)],$
$[\neg P(x_2), G(x_2)],$
$[\neg P(x_3), Q(x_3), P(b)],$
$[\neg P(x_4), Q(x_4), R(b)],$*
$[\neg Q(x_5), \neg P(x_5)]$*

Pelletier: 27
Pontos: 6
TPTP: SYN057-1

$[F(a)],$
$[\neg G(a)],$
$[\neg F(x_0), H(x_0)],$
$[\neg J(x_1), \neg I(x_1), F(x_1)],$
$[\neg H(x_2), G(x_2), \neg I(x_3), \neg H(x_3)],$
$[J(b)],$ *
$[I(b)]$ *

Pelletier: 28
Pontos: 8
TPTP: SYN058-1

$[\neg P(x_0), Q(x_1)],$
$[\neg Q(b), Q(c)],$
$[\neg Q(b), S(c)],$
$[\neg R(b), Q(c)],$
$[\neg R(b), S(c)],$
$[\neg S(x_2), \neg F(x_3), G(x_3)],$
$[P(d)],$ *
$[F(d)],$ *
$[\neg G(d)]$ *

Pelletier: 30
Pontos: 6
TPTP: SYN060-1

$[\neg F(x_0), \neg H(x_0)]$,
$[G(x_1), F(x_1)]$,
$[\neg G(x_2), \neg H(x_2)]$,
$[G(x_3), H(x_3)]$,
$[I(x_4), F(x_4)]$,
$[I(x_5), H(x_5)]$,
$[\neg I(a)]$ *

Pelletier: 31
Pontos: 5
TPTP: SYN061-1

$[\neg F(x_0), \neg G(x_0)]$,
$[\neg F(x_1), \neg H(x_1)]$,
$[I(a)], h0 : [F(a)]$,
$[H(x_2), J(x_2)]$,
$[\neg I(x_3), \neg J(x_3)]$ *

Pelletier: 32
Pontos: 6
TPTP: SYN062-1

$[\neg F(x_0), \neg G(x_0), I(x_0)]$,
$[\neg F(x_1), \neg H(x_1), I(x_1)]$,
$[\neg I(x_2), \neg H(x_2), J(x_2)]$,
$[\neg K(x_3), H(x_3)]$,
$[F(a)]$, *
$[K(a)]$, *
$[\neg J(a)]$ *

Pelletier: 44
Pontos: 3
TPTP: SYN068-1

$[\neg F(x_0), G(f(x_0))]$,
$[\neg F(x_1), H(x_1, f(x_1))]$,
$[\neg F(x_2), G(g(x_2))]$,
$[\neg F(x_3), \neg H(x_3, g(x_3))]$,
$[J(a)]$,
$[\neg G(x_4), H(a, x_4)]$,
$[\neg J(x_5), F(x_5)]$ *

Pelletier: 46
Pontos: 6
TPTP: SYN070-1

$[\neg F(x_0), F(f(x_0)), G(x_0)]$,
$[\neg F(x_1), H(f(x_1), x_1), G(x_1)]$,
$[\neg F(x_2), \neg G(f(x_2)), G(x_2)]$,
$[\neg F(x_3), G(x_3), F(a)]$,
$[\neg F(x_4), G(x_4), \neg G(a)]$,
$[\neg F(x_5), G(x_5), \neg F(x_6), G(x_6), J(a, x_6)]$,
$[\neg F(x_8), \neg F(x_7), \neg H(x_8, x_7), \neg J(x_7, x_8)]$,
$[F(b)]$, *
$[\neg G(b)]$ *

Pelletier: 47
Pontos: 10
TPTP: PUZ031-1

$[\neg Wolf(x_0), Animal(x_0)]$, $[\neg Fox(x_1), Animal(x_1)]$,
$[\neg Bird(x_2), Animal(x_2)]$, $[\neg Caterpillar(x_3), Animal(x_3)]$,
$[\neg Snail(x_4), Animal(x_4)]$,
$[Wolf(awolf)]$, $[Fox(afox)]$,
$[Bird(abird)]$, $[Caterpillar(acaterpillar)]$,
$[Snail(asnail)]$, $[Grain(agrain)]$,
$[\neg Grain(x_5), Plant(x_5)]$,
$[\neg Animal(x_8), \neg Plant(x_7), \neg Animal(x_9),$
$\neg Plant(x_6), \neg Smaller(x_9, x_8), \neg Eats(x_9, x_6),$
$Eats(x_8, x_7), Eats(x_8, x_9)]$,
$[\neg Bird(x_{11}), \neg Caterpillar(x_{10}), Smaller(x_{10}, x_{11})]$,
$[\neg Snail(x_{12}), \neg Bird(x_{13}), Smaller(x_{12}, x_{13})]$,
$[\neg Fox(x_{15}), \neg Bird(x_{14}), Smaller(x_{14}, x_{15})]$,
$[\neg Fox(x_{16}), \neg Wolf(x_{17}), Smaller(x_{16}, x_{17})]$,
$[\neg Wolf(x_{18}), \neg Fox(x_{19}), \neg Eats(x_{18}, x_{19})]$,
$[\neg Wolf(x_{20}), \neg Grain(x_{21}), \neg Eats(x_{20}, x_{21})]$,
$[\neg Bird(x_{22}), \neg Caterpillar(x_{23}), Eats(x_{22}, x_{23})]$,
$[\neg Bird(x_{24}), \neg Snail(x_{25}), \neg Eats(x_{24}, x_{25})]$,
$[\neg Caterpillar(x_{26}), Plant(caterpillarfood(x_{26}))]$,
$[\neg Caterpillar(x_{27}), Eats(x_{27}, caterpillarfood(x_{27}))]$,
$[\neg Snail(x_{28}), Plant(snailfood(x_{28}))]$,
$[\neg Snail(x_{29}), Eats(x_{29}, snailfood(x_{29}))]$,
$[\neg Animal(x_{30}), \neg Animal(x_{31}), \neg Grain(x_{32}),$
$\neg Eats(x_{30}, x_{31}), \neg Eats(x_{31}, x_{32})]$ *

Pelletier: 48
Pontos: 3
TPTP: SYN071-1

$[Equal(x_0, x_0)]$,
$[\neg Equal(x_2, x_1), Equal(x_1, x_2)]$,
$[\neg Equal(x_4, x_3), \neg Equal(x_3, x_5), Equal(x_4, x_5)]$,
$[Equal(a, b), Equal(c, d)]$,
$[Equal(a, c), Equal(b, d)]$,
$[\neg Equal(a, d)]$, *
$[\neg Equal(b, c)]$ *

Pelletier: 49
Pontos: 5
TPTP: SYN072-1

$[Equal(x_0, x_0)]$,
$[\neg Equal(x_2, x_1), Equal(x_1, x_2)]$,
$[\neg Equal(x_4, x_3), \neg Equal(x_3, x_5), Equal(x_4, x_5)]$,
$[\neg Equal(x_6, x_7), \neg P(x_6), P(x_7)]$,
$[Equal(x_8, c), Equal(x_8, d)]$,
$[P(a)]$,
$[P(b)]$,
$[\neg Equal(a, b)]$,
$[\neg P(e)]$ *

Pelletier: 55
Pontos: 8
TPTP: PUZ001-1

$[Lives(agatha)]$,
$[Lives(butler)]$,
$[Lives(charles)]$,
$[\neg Killed(x_0, x_1), \neg Richer(x_0, x_1)]$,
$[\neg Hates(agatha, x_2), \neg Hates(charles, x_2)]$,
$[\neg Hates(x_3, agatha), \neg Hates(x_3, butler)$,
$\neg Hates(x_3, charles)]$,
$[Hates(agatha, agatha)]$,
$[Hates(agatha, charles)]$,
$[\neg Killed(x_4, x_5), Hates(x_4, x_5)]$,
$[\neg Hates(agatha, x_6), Hates(butler, x_6)]$,
$[\neg Lives(x_7), Richer(x_7, agatha), Hates(butler, x_7)]$,
$[Killed(butler, agatha), Killed(charles, agatha)]$ *

Pelletier: 58	$[Equal(x_0, x_0)]$,
Pontos: 3	$[\neg Equal(x_2, x_1), Equal(x_1, x_2)]$,
TPTP: SYN080-1	$[\neg Equal(x_4, x_3), \neg Equal(x_3, x_5), Equal(x_4, x_5)]$,
	$[\neg Equal(x_6, x_7), Equal(f(x_6), f(x_7))]$,
	$[\neg Equal(x_8, x_9), Equal(g(x_8), g(x_9))]$,
	$[Equal(f(x_{10}), g(x_{11}))]$,
	$[\neg Equal(f(f(a)), f(g(b)))]$ *

Pelletier: 69	$[\neg T(impl(x_0, x_1)), \neg T(x_0), T(x_1)]$,
Pontos: 9	$[T(impl(x_3, impl(x_2, x_3)))]$,
TPTP: LCL039-1	$[T(impl(impl(x_5, impl(x_4, x_6)), impl(impl(x_5, x_4),$
	$impl(x_5, x_6))))]$,
	$[T(impl(impl(not(x_8), not(x_7)), impl(x_7, x_8)))]$,
	$[T(impl(neces(impl(x_9, x_{10})), impl(neces(x_9), neces(x_{10}))))]$,
	$[T(impl(neces(x_{11}), x_{11}))]$,
	$[\neg T(x_{12}), T(neces(x_{12}))]$,
	$[\neg T(impl(neces(a), not(neces(not(a)))))]$ *

TPTP: COM001-1	$[Succ(x_0, x_1), \neg Follows(x_0, x_1)]$,
	$[Succ(x_2, x_4), \neg Succ(x_2, x_3), \neg Succ(x_3, x_4)]$,
	$[Succ(x_7, x_5), \neg Has(x_5, goto(x_6)), \neg Label(x_6, x_7)]$,
	$[Succ(x_1 0, x_8), \neg Has(x_8, ifthen(x_9, x_1 0))]$,
	$[Label(loop, p3)]$,
	$[Has(p3, ifthen(equal_function(register_j, n), p4))]$,
	$[Has(p4, goto(out))]$,
	$[Follows(p5, p4)]$,
	$[Follows(p8, p3)]$,
	$[Has(p8, goto(loop))]$,
	$[\neg Succ(p3, p3)]$ *

TPTP: COM002-1

$[Succ(x_0, x_1), \neg Follows(x_0, x_1)],$
$[Succ(x_2, x_4), \neg Succ(x_2, x_3), \neg Succ(x_3, x_4)],$
$[Succ(x_7, x_5), \neg Has(x_5, goto(x_6)), \neg Label(x_6, x_7)],$
$[Succ(x_10, x_8), \neg Has(x_8, ifthen(x_9, x_10))],$
$[Label(loop, p3)],$
$[Has(p1, assign(register_j, n0))],$
$[Has(p2, assign(register_k, n1))],$
$[Has(p3, ifthen(equal_function(register_j, n), p4))],$
$[Has(p4, goto(out))],$
$[Has(p6, assign(register_k, times(n2, register_k)))],$
$[Has(p7, assign(register_j, plus(register_j, n1)))],$
$[Has(p8, goto(loop))],$
$[Follows(p2, p1)], \quad [Follows(p3, p2)],$
$[Follows(p5, p4)], \quad [Follows(p6, p3)],$
$[Follows(p7, p6)], \quad [Follows(p8, p7)],$
$[\neg Succ(p3, p3)]$ *

TPTP: NUM001-1

$[Equal(x_0, x_0)],$
$[\neg Equal(x_2, x_1), \neg Equal(x_1, x_3), Equal(x_2, x_3)],$
$[Equal(add(x_5, x_4), add(x_4, x_5))],$
$[Equal(add(x_6, add(x_7, x_8)), add(add(x_6, x_7), x_8))],$
$[Equal(sub(add(x_10, x_9), x_9), x_{10})],$
$[Equal(x_{11}, sub(add(x_{11}, x_{12}), x_{12}))],$
$[Equal(add(sub(x_{13}, x_{15}), x_{14}), sub(add(x_{13}, x_{14}), x_{15}))],$
$[Equal(sub(add(x_{16}, x_{18}), x_{17}), add(sub(x_{16}, x_{17}), x_{18}))],$
$[\neg Equal(x_{19}, x_{21}), \neg Equal(x_{20}, add(x_{19}, x_{22})),$
$Equal(x_{20}, add(x_{21}, x_{22}))],$
$[\neg Equal(x_{23}, x_{26}), \neg Equal(x_{24}, add(x_{25}, x_{23})),$
$Equal(x_{24}, add(x_{25}, x_{26}))],$
$[\neg Equal(x_{27}, x_{29}), \neg Equal(x_{28}, sub(x_{27}, x_{30})),$
$Equal(x_{28}, sub(x_{29}, x_{30}))],$
$[\neg Equal(x_{31}, x_{34}), \neg Equal(x_{32}, sub(x_{33}, x_{31})),$
$Equal(x_{32}, sub(x_{33}, x_{34}))],$
$[\neg Equal(add(add(a, b), c), add(a, add(b, c)))]$ *

Referências Bibliográficas

[1] P.B. Andrews. Theorem proving via general matings. *Journal of the ACM*, 28(2):193–214, 1981.

[2] Leo Bachmair, Ta Chen, C.R. Ramakrishnan, and I.V. Ramakrishnan. Subsumption algorithms based on search trees. In Helene Kirchner, editor, *Trees in Algebra and Programming - CAAP'96, Proceedings of the 21st International Colloquium*, volume 1059. Springer Lecture Notes in Computer Science (ISBN 3-540-61064-2), 1996. held in Linkoping, Sweden, 22-24 April, 1996.

[3] W. Bibel and E. Eder. Methods and calculi for deduction. In Dov M. Gabbay, C.J. Hogger, and J.A. Robinson, editors, *Handbook of Logic in Artificil Intelligence and Logic Programing - Logical Foundations*, volume 1, pages 67–182. Oxford University Press, 1993.

[4] Wolfgang Bibel and Peter H. Schmitt, editors. *Automated Deduction - A Basis for Applications*, volume volume I, Foundations - Calculi and Methods of *Applied Logic Series - volume 8*. Kluwer - Netherlands, 1998. ISBN 0-7923-5129-0.

[5] G. Bittencourt. In the quest of the missing link. In *Proceedings of IJCAI 15, Nagoya, Japan, August 23-29*, pages 310–315. Morgan Kaufmann (ISBN 1-55860-480-4), 1997.

[6] G. Bittencourt. Concurrent inference through dual transformation. *Logic Journal of the Interest Group in Pure and Applied Logics (IGPL)*, 6(6):795–834, 1998.

[7] G. Bittencourt. *Inteligência Artificial Ferramentas e Teorias.* Editora da UFSC, ISBN 85-328-0138-2, 362 p., Florianópolis, SC, 2^a edição, 2001.

[8] G. Bittencourt and I. Tonin. A multi-agent approach to first-order logic. In *Proceedings of 8^{th} Portuguese Conference on Artificial Intelligence (EPIA'97), Coimbra, Portugal, October 6-9.* Springer Verlag, 1997.

[9] G. Bittencourt and I. Tonin. A proof strategy based on a dual representation. In E. Roanes-Lozano and J. Calmet, editors, *Proceedings of the Fifth International Conference Artificial Intelligence and Symbolic Computation Theory, Implementations and Applications (AISC'2000)*, pages 78–91. Springer Verlag Lecture Notes in Artificial Intelligence (LNAI 1930), 2000. (held in Madrid, Spain, July, 17-19, 2000).

[10] G. Bittencourt and I. Tonin. An algorithm for dual transformation in first-order logic. *Journal of Automated Reasoning*, 27(4):353–389, 2001.

[11] C.-L. Chang and R.C.-T. Lee. *Symbolic Logic and Mechanical Theorem Proving.* Academic Press, Computer Science Classics, 1973.

[12] Newton C.A. da Costa. Pequena história da lógica contemporânea. Folha de São Paulo, caderno Folha Mais. 04 de agosto de 2002.

[13] E. Eder. Consolution and its relation to resolution. In *Proceedings of IJCAI 12*, 1991.

[14] N. Eisinger and H.J. Ohlbach. Deduction systems based on resolution. In Dov M. Gabbay, C.J. Hogger, and J.A. Robinson, editors, *Handbook of Logic in Artificil Intelligence and Logic Programing - Logical Foundations*, volume 1, pages 183–271. Oxford University Press, 1993.

[15] M. Fitting. *First-Order Logic and Automated Theorem Proving.* Springer Verlag, New York, 1990.

[16] F.J.Pelletier and P.Rudnicki. Non-obviousness. AAR Newsletter, september 1986. number 6.

[17] Dov M. Gabbay, C.J. Hogger, and J.A. Robinson, editors. *Handbook of Logic in Artificil Intelligence and Logic Programing - Deduction Methodologies*, volume 2. Oxford University Press, 1994.

[18] M.R. Genesereth and N.J. Nilsson. *Logical Foundations of Artificial Intelligence.* Morgan Kaufmann Publishers Inc, Los Altos, California, 1987.

[19] Georg Gottlob. Subsumtion and implication. *Information Processing Letters*, 24:109–111, 1987. 30 January.

[20] Georg Gottlob and Christian G. Fermüller. Removing redundancy from a clause. *Artificial Intelligence*, 61:263–289, 1993.

[21] Georg Gottlob and Alexander Leitsch. On the efficiency of subsumption algorithms. *Journal of the ACM*, 32(2):280–295, 1985.

[22] R.A. Kowalski and P. Hayes. Semantic trees in automatic theorem proving. In D. Michie and B. Meltzer, editors, *Machine Intelligence 4*, pages 87–101. Edinburgh University Press, Edinburgh, GB, 1969.

[23] Robert A. Kowalsky. *Studies in the Completeness and Efficiency of Theorem-Proving by Resolution.* PhD thesis, University of Edinburgh, April 1970.

[24] J. McCarthy and P.J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In D. Michie and B. Meltzer, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, Edinburgh, GB, 1969.

[25] A. Newell. Physical symbol systems. *Cognitive Science*, 4:135–183, 1980.

[26] A. Newell and H.A. Simon. The logic theory machine. *IRE Transactions on Information Theory*, 3:61–79, September 1956.

[27] N.J. Nilsson. *Problem Solving Methods in Artificial Intelligence.* McGraw-Hill, New York, 1971.

[28] N.J Nilsson. *Principles of Artificial Intelligence.* Springer Verlag, Berlin, Heidelberg, New York, 1982.

[29] Francis Jeffry Pelletier. Seventy-five problems for testing automatic theorem provers. *Journal of Automted Reasoning*, (2):191–216, 1986.

[30] R. Penrose. *The Emperor's New Mind - Concerning Computers, Minds and Laws of Physics.* Oxford University Press, 1989.

[31] M.J. Quinn. *Desingning Efficient Algorithms for Parallel Computers.* McGraw-Hill International Editions, 1987.

[32] J.A. Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12(1):23–41, January 1965.

[33] S.J. Russell and P. Norvig. *Artificial Intelligence A Modern Approach.* Prentice Hall, Upper Saddle River, NJ, 1995.

[34] J.R. Slagle, C.L. Chang, and R.C.T. Lee. A new algorithm for generating prime implicants. *IEEE Transactions on Computing*, 19(4):304–310, 1970.

[35] R.M. Smullyan. *First Order Logic.* Springer Verlag, 1968.

[36] R. Socher. Optimizing the clausal normal form transformation. *Journal of Automated Reasoning*, 7(3):325–336, 1991.

[37] L. Sterling and E. Shapiro. *The Art of Prolog.* M.I.T. Press Series in Logic Programming, Cambridge, MA, London, England, 1986.

[38] M.E. Stickel. An introduction to automated deduction. In *Fundamentals of Artificial Intelligence, LNCS No. 232*, pages 75–132. Springer Verlag, Berlin Heidlberg New York Tokyo, 1986.

[39] Tanel Tammet. Towards efficient subsumption. In Claude Kirchner and Hèléne Kirchner, editors, *15th International Conference on Automated Deduction, CADE-15*, pages 427–441, 1998. volume 1421 of Springer verlag *LNAI.*

[40] A. Tarski. The concept of truth in formalized languages. In A. Tarski, editor, *Logic Semantics and Mathematics.* Oxford, Clarendon Press, 1956.

[41] I. Tonin and G. Bittencourt. Logik: Um ambiente para o ensino de logica. In *V Congresso Iberoamericano de Educacion Superior en Computacion - EDUC*, 1996. Cidade do México - México.

[42] I. Tonin and G. Bittencourt. Uma implementação orientada a objeto para a transformação dual em lógica de primeira ordem. In *XXIV Brazilian Software and Hardware Seminars (SEMISH'97)*, pages 411–422, 1997. Brasília, DF.

[43] I. Tonin and G. Bittencourt. Forma normal disjuntiva em lógica de primeira ordem. In *Anais do I Congresso de Lógica Aplicada à Tecnologia (LAPTEC'2000)*, pages 417–429. Faculdade SENAC de Ciências Exatas e Tecnologia (ISBN 85-85795-29-8), 2000. (São Paulo, SP, 11 a 15 de setembro de 2000).

[44] I. Tonin and G. Bittencourt. A concurrent algorithm for logical subsumption. In *Anais do II Congresso de Lógica Aplicada à Tecnologia (LAPTEC'2001).* Faculdade SENAC de Ciências Exatas e Tecnologia, 2001. (São Paulo, SP, 12 a 14 de novembro de 2001).

[45] I. Tonin and G. Bittencourt. A concurrent algorithm for logical subsumption. In Jair M. Abe and João I. da Silva Filho, editors, *Frontiers in Artificial Intelligence and its Applications*, pages 263–269. IOS Press - Neatherlands, 2001. ISSN 0922-6389.

[46] Isabel Tonin. Um método para inferência lógica baseado na transformação dual. Master's thesis, UFSC - Universidade Federal de Santa Catarina, EEL - Departamento de Engenharia Elétrica, 1997. Florianópolis-SC, Brasil.

[47] M.H. Van Emden and R.A. Kowalski. The semantics of predicate logic as a programming language. *Journal of the ACM*, 23(4):733–742, October 1976.

[48] G. A. Wilson and J. Minker. Resolution refinements and search strategies – A comparative study. *IEEE Transactions on Computers*, C-25(8):782–800, August 1976.

[49] P.H. Winston. *Artificial Intelligence (2nd Edition)*. Addison-Wesley Publishing Company, Reading, MA, 1984.

[50] Larry Wos. The problem of choosing the type of subsumption to use. *Journal of Automated Reasoning*, 7:435–438, 1991.

[51] Larry Wos and Robert Veroff. Logical basis for automating reasoning. In Dov M. Gabbay, C.J. Hogger, and J.A. Robinson, editors, *Handbook of Logic in Artificil Intelligence and Logic Programing - Deduction Methodologies*, volume 2, pages 1–40. Oxford University Press, 1994.