Série

Apontamentos

Maria do Carmo Nicoletti

A CARTILHA DA LÓGICA

Universidade Federal de São Carlos

EdUFSCar

Aluno: [illegible]

Universidade Federal de São Carlos
Editora da Universidade Federal de São Carlos
Via Washington Luís, km 235 - 13565-905 - São Carlos, SP, Brasil
Telefax [16] 3351-8137
E-mail edufscar@ufscar.br
http://www.editora.ufscar.br

Maria do Carmo Nicoletti

A cartilha da lógica

São Carlos

EdUFSCar

2008

Coordenação Editorial
Luis Gustavo Sousa Sguissardi

Preparação e Revisão de Texto
Marina Venâncio Grandolpho
Ingrid Pereira de Souza Favoretto

Editoração Eletrônica
Vítor Massola Gonzales Lopes

Impressão e Acabamento
Departamento de Produção Gráfica da Universidade Federal de São Carlos

Ficha catalográfica elaborada pelo DePT da Biblioteca Comunitária da UFSCar

Nicoletti, Maria do Carmo.
N643c A cartilha da lógica / Maria do Carmo Nicoletti -- São Carlos : EdUFSCar, 2008.
205 p.

ISBN – 978-85-7600-117-1

1. Lógica. 2. Lógica proposicional. 3. Lógica de primeira ordem. 4. Álgebra booleana. 5. Interpretação. 6. Modelo de Herbrand. I. Título.

CDD – 160 (20ª)
CDU – 16

A professora Maria do Carmo Nicoletti é docente do Departamento de Computação da UFSCar.

SUMÁRIO

PREFÁCIO

A Lógica desempenha um papel central em muitas áreas de conhecimento, particularmente em Matemática e Computação. A estreita ligação entre Lógica e Computação está principalmente relacionada ao desenvolvimento de linguagens para modelar situações e problemas encontrados, com vistas às suas soluções. Raciocinar sobre essas situações significa construir argumentos sobre elas – o objetivo é fazer isso formalmente, de maneira que a prova da validade de argumentos possa ser automatizada. Para a especificação e o desenvolvimento de argumentos de maneira rigorosa é preciso primeiro o estabelecimento de uma linguagem na qual se possa expressar sentenças evidenciando sua estrutura lógica. A Lógica Proposicional e a Lógica de Predicados, os principais assuntos tratados nestes *Apontamentos*, são duas dessas linguagens formais.

A principal motivação para a elaboração deste *Apontamento* foi a de disponibilizar material impresso para o acompanhamento das aulas da disciplina Introdução à Lógica, que faz parte do elenco de disciplinas introdutórias dos cursos de Bacharelado em Ciências da Computação e de Engenharia da Computação da Universidade Federal de São Carlos. Os assuntos tratados seguem a ementa da disciplina. Acredito que a disponibilidade de material impresso sobre os assuntos discutidos em classe contribui para o melhor entendimento de conceitos apresentados; permite que a atenção seja direcionada ao que está sendo discutido/apresentado em sala de aula; dá segurança ao aprendizado, uma vez que se torna uma referência dos conceitos discutidos; abre perspectivas para novas discussões/questões, por meio dos exemplos fornecidos e dos exercícios propostos.

Os conceitos e resultados descritos podem ser e encontrados em diversas referências sobre os assuntos tratados. A teoria descrita no Capítulo 2, sobre Álgebra de Boole, foi substancialmente fundamentada em notação e definições encontradas em Berztiss (1975), publicação que, apesar da idade avançada (como referência a assuntos computacionais), é precisa e formal em muitos de seus capítulos, o que a permite ser, até os dias de hoje, altamente apreciada e respeitada.

São Carlos, março 2008
Maria do Carmo Nicoletti

1. A LÓGICA PROPOSICIONAL

A Lógica Proposicional (ou Cálculo Proposicional) é um dos mais simples formalismos lógicos existentes e pode ser considerada a formalização de formas simples de raciocínio. Apesar de simples e sem recursos para a representação de muitos problemas/situações do mundo real, ainda assim a Lógica Proposicional, como será visto, é conveniente e poderosa o suficiente para lidar com muitos problemas e formalizar precisamente a busca de suas soluções. Este capítulo aborda os principais elementos da Lógica Proposicional, tendo como objetivo, além de fornecer e discutir os fundamentos da Lógica Proposicional, estabelecer a conceituação necessária para a compreensão dos capítulos que o seguem.

A Lógica Proposicional lida apenas com enunciados declarativos, chamados proposições. As sentenças exclamativas, imperativas e interrogativas são, pois, excluídas.

1.1 INTRODUÇÃO

Definição 1.1 Uma *proposição* é uma sentença declarativa que pode assumir os valores-verdade v (com o significado de que a proposição é verdadeira) ou f (com o significado de que a proposição é falsa), um excluindo o outro.

Exemplo 1.1 Exemplos de proposições:

1. A soma dos números 5 e 6 é igual a 11.
2. O nome do menino é Pedro.
3. O programa tem problemas.
4. As refeições são saborosas e caras.
5. Um triângulo ABC é retângulo se e somente se tem um ângulo reto.
6. Se eu como muito então eu engordo.
7. A mesa é de madeira ou o chapéu é de palha.

Exemplo 1.2 As quatro sentenças a seguir não são tratadas na Lógica Proposicional, uma vez que as duas primeiras são interrogativas, a terceira é imperativa e a quarta é exclamativa.

1. Quando você pretende viajar?
2. Que horas são?
3. Que Deus a acompanhe.
4. O relato de sua viagem é incrível!

Observação 1.1 É importante mencionar que muitas sentenças comumente utilizadas no dia a dia não são completamente verdade e tampouco são complemente falsas. Por exemplo, se um programa computacional teve como saída um resultado inesperado, freqüentemente não está claro se o resultado foi efetivamente causado por um problema no programa ou se foi conseqüência de problemas em programas subjacentes. A Lógica Proposicional, entretanto, lida apenas com proposições verdadeiras ou falsas; qualquer proposição que não se adequar a essa bivalência não é considerada.

Observação 1.2 As proposições 1, 2 e 3 do Exemplo 1.1 são *proposições atômicas* (ou *átomos*), uma vez que nelas não aparecem os conectivos **e**, **ou**, **se...então** e **se e somente se**, tratados a seguir. As proposições 4, 5, 6 e 7 são *proposições compostas* justamente porque nelas tais conectivos aparecem. Toda proposição composta contém pelo menos um conectivo lógico.

Observação 1.3 A Lógica Proposicional lida apenas com proposições que têm valores-verdade v ou f, isto é, segue o *princípio do terceiro excluído* ou, equivalentemente, é considerada uma lógica *bivalente*. Qualquer proposição que não tenha o valor-verdade v necessariamente terá que ter o valor-verdade f.

Observação 1.4 [Grassmann & Tremblay, 1996] Freqüentemente sentenças em língua natural (i.e., português, inglês, francês, espanhol, etc.) são ambíguas uma vez que algumas palavras usadas nas sentenças podem ter mais do que um significado. A especificação da linguagem lógica tem como objetivo evitar essa ambigüidade. Além disso, os conectivos de muitas línguas naturais refletem preocupação com causalidade, intenção, ênfase, crenças e temporalidade. A palavra "se", por exemplo, freqüentemente implica algum tipo de causalidade. A lógica clássica, por outro lado, apenas lida com verdade e falsidade, o que significa que a noção de causalidade não é possível de ser expressa. Por essa razão, são introduzidos símbolos matemáticos para representar os conectivos da lógica. Esses símbolos podem então ser definidos de maneira não ambígua.

Definição 1.2 O *alfabeto* da Lógica Proposicional é constituído por:

- Símbolos de pontuação: ( )
- Símbolos de verdade: verdade, falso
- Símbolos proposicionais atômicos: p, q, r, s, t, u, ...
- Conectivos lógicos: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$

1.2 FÓRMULAS BEM-FORMADAS (WFF) E INTERPRETAÇÃO

Como estabelecido na Observação 1.2, novas proposições podem ser construídas com base em outras, por meio do uso de conectivos lógicos. Essas novas proposições, bem como as proposições atômicas, são chamadas de fórmulas bem-formadas (wff – *well-formed formulas*), desde que satisfaçam a Definição 1.3.

Observação 1.5 Para representar proposições gerais (sejam elas atômicas ou compostas) são usadas metavariáveis proposicionais, representadas por letras do alfabeto grego: α, β, λ, χ, δ, ε, etc.

Definição 1.3 Uma *wff* é definida recursivamente como segue:

1. os símbolos de verdade (i.e., verdade e falso, como especificados em Definição 1.2) são wffs;
2. um átomo é uma wff;
3. se α e β são wffs, então as proposições compostas listadas na Tabela 1.1 também são wffs;
4. as únicas wffs são as definidas nos itens 1, 2 e 3 acima.

Tabela 1.1 Regras para a formação de wffs.

wff	lida como	wff chamada de
$(\neg\alpha)$	não α	negação
$(\alpha \wedge \beta)$	α e β	conjunção
$(\alpha \vee \beta)$	α ou β	disjunção
$(\alpha \rightarrow \beta)$	se α então β	condicional
$(\alpha \leftrightarrow \beta)$	α se e somente se β	bicondicional

Observação 1.6 O operador $\neg$(não) é um *operador unário*, ou seja, é aplicado a um único operando. Quando o operando em questão for um átomo (p, por exemplo), tanto o átomo quanto o átomo negado (ou seja, tanto p quanto $\neg$p), é denominado *literal.* Os conectivos $\wedge$, $\vee$, $\rightarrow$ e $\leftrightarrow$ são *conectivos binários*, pois são usados para conectar duas wffs.

Observação 1.7 Se cuidados não forem tomados, expressões lógicas podem ser ambíguas. Considere, por exemplo, as três proposições atômicas:

p definida como "Maria termina o relatório"

q definida como "Maria está feliz" e

r definida como "Maria vai ao cinema".

Considere a proposição composta p $\rightarrow$ q $\vee$ r. Sem o estabelecimento de uma regra precisa, essa expressão pode ser interpretada de duas maneiras:

(1) pode significar $(p \rightarrow q) \vee r$ que, parafraseada em língua natural, expressa "se Maria termina o relatório, Maria está feliz e irá, em qualquer circunstância, ao cinema";

(2) pode significar $p \rightarrow (q \vee r)$ que, parafraseada em língua natural, expressa "se Maria termina o relatório Maria está feliz e Maria irá ao cinema".

Portanto, a expressão $p \rightarrow q \vee r$ é ambígua. Uma maneira de evitar ambigüidade é por meio do estabelecimento de regras, como as da Definição 1.3, que define a sintaxe da Lógica Proposicional. Uma expressão lógica que segue as regras da Definição 1.3 não é ambígua.

Observação 1.8 É importante enfatizar que a Definição 1.3 é recursiva, ou seja, para que uma fórmula seja uma wff é preciso que suas "partes" também sejam wffs. Na expressão $(\neg\alpha)$, α é considerada o escopo da negação e o conectivo $\neg$ é chamado de *conectivo principal* da expressão $\neg\alpha$. Em uma conjunção, i.e., uma expressão lógica da forma $(\alpha \wedge \beta)$, o conectivo principal é o $\wedge$ e α e β são chamados respectivamente de *escopo à esquerda* e *escopo à direita* da conjunção. Uma nomeação similar se aplica às expressões $(\alpha \vee \beta)$, $(\alpha \rightarrow \beta)$ e $(\alpha \leftrightarrow \beta)$. Os escopos podem, por sua vez, ser expressões compostas e, portanto, os conectivos encontrados nos escopos são subconectivos da expressão em questão. Por exemplo, na expressão $(p \wedge q) \rightarrow r$ o conectivo principal é $\rightarrow$ e o subconectivo do escopo à esquerda é $\wedge$. O escopo à direita do conectivo principal não tem conectivo. Note que o $\wedge$ é o conectivo principal do escopo à esquerda.

Exemplo 1.3 Considere o procedimento para verificar se uma dada fórmula α é bem-formada ou não.

(a) Suponha que a fórmula α seja $((p \leftrightarrow q) \vee (r \wedge (\neg s))) \leftrightarrow (\neg p)$. O conectivo principal de α é $\leftrightarrow$. Seguindo as regras apresentadas na Tabela 1.1, para que α seja bem-formada, é preciso mostrar que $((p \leftrightarrow q) \vee (r \wedge \neg s))$ e que $(\neg p)$ são bem-formadas. Para mostrar que $((p \leftrightarrow q) \vee (r \wedge \neg s))$, cujo conectivo principal é $\vee$, é bem-formada, é preciso evidenciar que tanto $(p \leftrightarrow q)$ quanto $(r \wedge \neg s)$ são bem-formadas.

Note que $(p \leftrightarrow q)$ é bem-formada uma vez que p é bem-formada (é um átomo), q é bem-formada (é um átomo) e a composição de duas fórmulas bem-formadas, usando o conectivo $\leftrightarrow$ é bem-formada (regra da Tabela 1.1).

Note que na fórmula $(r \wedge (\neg s))$ a fórmula r é bem-formada, uma vez que é um átomo e $(\neg s)$ é também bem-formada, uma vez que é a negação de um átomo. Dado que ambas, tanto r quanto $(\neg s)$, são bem-formadas, $(r \wedge (\neg s))$ é bem-formada pois é uma conjunção de duas fórmulas bem-formadas.

Como ambas, $(p \leftrightarrow q)$ e $(r \wedge (\neg s))$, são bem-formadas, também o será a disjunção dessas duas fórmulas, $((p \leftrightarrow q) \vee (r \wedge (\neg s)))$. Por outro lado, a fórmula $(\neg p)$ é bem-formada, pois é a negação de um átomo. Dado que ambas são bem-formadas, a fórmula composta de duas fórmulas bem-formadas por meio do uso do conectivo $\leftrightarrow$ é também bem-formada.

(b) Suponha que a fórmula α seja $((p \rightarrow q) \leftrightarrow (p \vee))$. Note que o conectivo principal de α é $\leftrightarrow$, e para α ser bem-formada, de acordo com a Tabela 1.1, ambas as fórmulas, $(p \rightarrow q)$ e $(p \vee)$, precisam ser bem-formadas.

A fórmula $(p \rightarrow q)$ é bem-formada uma vez que p e q são fórmulas atômicas e a fórmula resultante, que é a implicação $(p \rightarrow q)$, é bem-formada. Dado que $\vee$ é o conectivo principal da fórmula $(p \vee)$, tal fórmula só seria provada ser bem-formada se o conectivo estivesse conectando duas fórmulas bem-formadas. No caso, uma das fórmulas é a fórmula atômica p, que é bem-formada. No entanto, não existe a segunda outra fórmula, o que caracteriza $(p \vee)$ como não bem-formada e, conseqüentemente, a fórmula α não é bem-formada. Outros exemplos de fórmulas não bem-formadas:

$$((p \rightarrow \rightarrow q) \vee s)$$

$$((p \rightarrow \wedge q)\ s)$$

$$(\ (p \vee (\neg(p \rightarrow q)) \wedge \vee q)\ )$$

Definição 1.4 Seja α uma fórmula da Lógica Proposicional. Uma *subfórmula* de α é definida por:

1. α é subfórmula de α;
2. se α: $(\neg\beta)$ então β é uma subfórmula de α;
3. se α é uma fórmula em qualquer dos padrões:

$$(\beta \wedge \gamma), (\beta \vee \gamma), (\beta \rightarrow \gamma) \text{ ou } (\beta \leftrightarrow \gamma)$$

então β e γ são subfórmulas de α;

4. se α é uma subfórmula de β então toda subfórmula α é subfórmula de β.

Informalmente, uma subfórmula de uma fórmula α é uma parte de α que é uma fórmula (de acordo com a Definição 1.3).

Exemplo 1.4 Considere a fórmula α: $(p \rightarrow q) \leftrightarrow r$. As fórmulas $(p \rightarrow q)$, p, q e r são todas subfórmulas de α.

Observação 1.9 Como pode ser evidenciado nas várias referências bibliográficas ao final dessa publicação, poucos autores usam expressões lógicas completamente parentizadas, dado que tais expressões podem ser bastante longas e, freqüentemente, são difíceis de serem lidas. Os parênteses podem ser omitidos quando sua eliminação não introduzir ambigüidade.

Particularmente, os parênteses externos de uma expressão lógica são, quase sempre, omitidos. Por exemplo, ao invés de escrever $(p \vee q)$ escreve-se $p \vee q$; ao invés de escrever $((p \vee q) \leftrightarrow$ verdade) escreve-se $(p \vee q) \leftrightarrow$ verdade. Ao eliminar tais parênteses, no entanto, não deve ser esquecido que eles devem voltar a ser introduzidos se a expressão em questão for composta com alguma outra expressão.

Parênteses dentro de uma expressão também podem ser omitidos. A leitura da expressão resultante é feita usando as chamadas regras de prioridade, que estabelecem uma hierarquia entre os conectivos. A convenção que a maioria dos autores estabelece para os conectivos lógicos segue a ordem decrescente de precedência especificada na Tabela 1.2.

Tabela 1.2 Precedência de conectivos lógicos.

$\neg$	maior
$\wedge$	
$\vee$	$\Downarrow$
$\rightarrow$	
$\leftrightarrow$	menor

O conectivo $\neg$ tem sempre a maior precedência. Por essa razão, a expressão lógica $\neg p \vee q$ deve ser entendida como $(\neg p) \vee q$ e não como $\neg(p \vee q)$.

De acordo com a Tabela 1.2, na expressão $p \wedge q \vee r$, o $\wedge$ tem precedência sobre o $\vee$ quando da especificação de subexpressões. Portanto, $p \wedge q \vee r$ deve ser entendida como $(p \wedge q) \vee r$. De maneira semelhante, $p \rightarrow q \vee r$ deve ser entendida como $p \rightarrow (q \vee r)$, uma vez que o $\vee$ tem precedência sobre o $\rightarrow$. O conectivo $\leftrightarrow$ tem a precedência mais baixa de todas, o que significa que $p \leftrightarrow p \rightarrow r$ deve ser entendido como $p \leftrightarrow (p \rightarrow r)$.

Em algumas expressões, entretanto, as regras de prioridade não são suficientes para remover todas as ambigüidades. Por exemplo, a expressão $p \rightarrow q \rightarrow r$ pode ser entendida ou como $(p \rightarrow q) \rightarrow r$ ou como $p \rightarrow (q \rightarrow r)$. A escolha da interpretação usada dependerá da associatividade do conectivo $\rightarrow$. Todos os conectivos lógicos binários são associativos à esquerda, e, conseqüentemente, $p \rightarrow q \rightarrow r$ deve ser entendido como $(p \rightarrow q) \rightarrow r$.

Definição 1.5 Cada uma das expressões envolvendo α e β na Definição 1.3 é chamada de *forma sentencial*. Uma forma sentencial é uma especificação abstrata da sintaxe de um número infinito de wffs compostas de símbolos que representam proposições atômicas. Uma wff que sintaticamente se ajusta a uma forma sentencial é chamada de uma *instância de substituição da forma sentencial*.

Exemplo 1.5 A wff $(p \wedge (q \rightarrow r))$ é uma instância de substituição de qualquer uma das seguintes formas sentenciais:

1. α, tal que α é $(p \wedge (q \rightarrow r))$.
2. $(\alpha \wedge \beta)$, tal que α é p e β é $(q \rightarrow r)$.
3. $(\alpha \wedge (\beta \rightarrow \gamma))$, tal que α é p, β é q e γ é r.

Definição 1.6 A *semântica* – ou significado – de uma wff é o valor-verdade a ela associado por meio de uma interpretação. Uma *interpretação* I na Lógica Proposicional é uma função cujo contradomínio possui apenas dois elementos, como mostra a Figura 1.1, tal que:

- o domínio de I é o conjunto das wffs da Lógica Proposicional;
- o contradomínio de I é o conjunto {v,f};
- os símbolos de verdade do alfabeto, verdade e falso, são interpretados por I como

$$I[\text{verdade}] = v \text{ e } I[\text{falso}] = f;$$

- Dado um símbolo proposicional p, $I[p] \in \{v,f\}$.

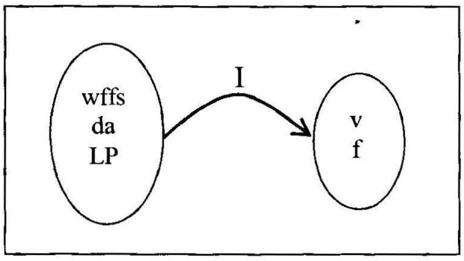


Figura 1.1 Interpretação de fórmulas como uma função em {v,f}.

Observação 1.10 A interpretação dos símbolos verdade é fixa, ou seja, para qualquer interpretação I, I[verdade] = v e I[falso] = f.

Definição 1.7 Seja γ uma wff e considere uma interpretação I. O *significado* de γ dado por I, notado por $I[\gamma]$, é determinado pelas regras:

1. Se γ: p, tal que p é um símbolo proposicional, então $I[\gamma] = I[p]$ e $I[p] \in \{v,f\}$.
2. Se γ: verdade então $I[\gamma] = v$. Se γ: falso, então $I[\gamma] = f$.
3. Seja β uma wff. Se γ: $\neg\beta$, então

$$I[\gamma] = I[\neg\beta] = v \text{ se } I[\beta] = f$$

$$I[\gamma] = I[\neg\beta] = f \text{ se } I[\beta] = v$$

4. Sejam α e β duas wffs. Se γ: $(\alpha \wedge \beta)$, então

$$I[\gamma] = I[(\alpha \wedge \beta)] = v \quad \text{se } I[\alpha] = v \text{ e } I[\beta] = v$$

$$I[\gamma] = I[(\alpha \wedge \beta)] = f \text{ se } \begin{cases} I[\alpha] = v \text{ e } I[\beta] = f \\ I[\alpha] = f \text{ e } I[\beta] = v \\ I[\alpha] = f \text{ e } I[\beta] = f \end{cases}$$

5. Sejam α e β duas wffs. Se γ: $(\alpha \vee \beta)$, então

$$I[\gamma] = I[(\alpha \vee \beta)] = v \text{ se } \begin{cases} I[\alpha] = v \text{ e } I[\beta] = v \\ I[\alpha] = v \text{ e } I[\beta] = f \\ I[\alpha] = f \text{ e } I[\beta] = v \end{cases}$$

$$I[\gamma] = I[(\alpha \vee \beta)] = f \text{ se } I[\alpha] = f \text{ e } I[\beta] = f$$

6. Sejam α e β duas wffs. Se γ: $(\alpha \rightarrow \beta)$, então

$$I[\gamma] = I[(\alpha \rightarrow \beta)] = v \text{ se } \begin{cases} I[\alpha] = v \text{ e } I[\beta] = v \\ I[\alpha] = f \text{ e } I[\beta] = v \\ I[\alpha] = f \text{ e } I[\beta] = f \end{cases}$$

$$I[\gamma] = I[(\alpha \rightarrow \beta)] = f \text{ se } I[\alpha] = v \text{ e } I[\beta] = f$$

7. Sejam α e β duas wffs. Se γ: $(\alpha \leftrightarrow \beta)$, então

$$I[\gamma] = I[(\alpha \leftrightarrow \beta)] = v \text{ se } \begin{cases} I[\alpha] = v \text{ e } I[\beta] = v \\ I[\alpha] = f \text{ e } I[\beta] = f \end{cases}$$

$$I[\gamma] = I[(\alpha \leftrightarrow \beta)] = f \text{ se } \begin{cases} I[\alpha] = v \text{ e } I[\beta] = f \\ I[\alpha] = f \text{ e } I[\beta] = v \end{cases}$$

Exemplo 1.6 Considere a fórmula α: $((\neg p) \wedge q) \rightarrow \neg r$ e uma interpretação I dada por:

	p	q	r
I	v	f	f

Para determinar o significado semântico de α de acordo com I (i.e., $I[\alpha]$), considere:

	p	q	r	$\neg p$	$\neg r$	$(\neg p) \wedge q$	α
I	v	f	f	f	v	f	v

ou seja, $I[\alpha] = v$.

Exemplo 1.7 Considere a fórmula α: $((\neg p) \vee q) \rightarrow ((\neg r) \leftrightarrow (\neg s))$ e uma interpretação I dada por:

	p	q	r	s
I	v	v	f	v

Para determinar o significado semântico de α de acordo com I (i.e., I[α]), considere:

	p	q	r	s	$\neg p$	$\neg r$	$\neg s$	$(\neg p) \vee q$	$(\neg r) \leftrightarrow (\neg s)$	α
I	v	v	f	v	f	v	f	v	f	f

ou seja, I[α] = f.

Definição 1.8 Dada uma wff α, a *tabela-verdade* de α mostra a semântica de α sob todas as possíveis interpretações; cada linha da tabela é associada a uma possível interpretação. O número de possíveis interpretações é função do número de átomos presentes em α. A tabela-verdade de uma fórmula α com n átomos tem 2^n possíveis interpretações, ou seja, 2^n linhas.

Exemplo 1.8 Considere a fórmula α: $((\neg p) \vee q) \rightarrow ((\neg r) \leftrightarrow (\neg s))$. A Tabela 1.3 é a tabela-verdade de α. Note que α tem quatro átomos (p, q, r e s) e, portanto, sua tabela-verdade tem $2^4 = 16$ possíveis interpretações.

Tabela 1.3 As 16 possíveis interpretações da expressão α: $((\neg p) \vee q) \rightarrow ((\neg r) \leftrightarrow (\neg s))$.

	p	q	r	s	$\neg p$	$\neg r$	$\neg s$	$(\neg p) \vee q$	$(\neg r) \leftrightarrow (\neg s)$	α
I_1	v	v	v	v	f	f	f	v	v	v
I_2	v	v	v	f	f	f	v	v	f	f
I_3	v	v	f	v	f	v	f	v	f	f
I_4	v	v	f	f	f	v	v	v	v	v
I_5	v	f	v	v	f	f	f	f	v	v
I_6	v	f	v	f	f	f	v	f	f	v
I_7	v	f	f	v	f	v	f	f	f	v
I_8	v	f	f	f	f	v	v	f	v	v
I_9	f	v	v	v	v	f	f	v	v	v
I_{10}	f	v	v	f	v	f	v	v	f	f
I_{11}	f	v	f	v	v	v	f	v	f	f
I_{12}	f	v	f	f	v	v	v	v	v	v
I_{13}	f	f	v	v	v	f	f	v	v	v
I_{14}	f	f	v	f	v	f	v	v	f	f
I_{15}	f	f	f	v	v	v	f	v	f	f
I_{16}	f	f	f	f	v	v	v	v	v	v

Note, na Tabela 1.3, que as 16 possíveis interpretações foram geradas sistematicamente, seguindo o padrão adotado na literatura. A geração sistemática permite o controle e a fácil identificação de uma particular interpretação. A interpretação I considerada no Exemplo 1.6 é a interpretação nomeada I_3 na Tabela 1.3.

Observação 1.11 Os conectivos $\wedge$, $\vee$ e $\leftrightarrow$ são simétricos no sentido de que a ordem das duas expressões conectadas por eles não afeta o valor-verdade da expressão resultante. No entanto, o conectivo $\rightarrow$ não é simétrico, pois as expressões $p \rightarrow q$ e $q \rightarrow p$ têm diferentes valores-verdade em uma mesma interpretação. Os valores-verdade de todos os conectivos estão sumarizados na Tabela 1.4, usando duas wffs atômicas.

Tabela 1.4 Resumo dos conectivos.

	p	q	$\neg p$	$\neg q$	$p \wedge q$	$p \vee q$	$p \rightarrow q$	$p \leftrightarrow q$
I_1	v	v	f	f	v	v	v	v
I_2	v	f	f	v	f	v	f	f
I_3	f	v	v	f	f	v	v	f
I_4	f	f	v	v	f	f	v	v

1.3 VALIDADE E INCONSISTÊNCIA

O valor-verdade – ou simplesmente valor – de uma fórmula sempre diz respeito a uma determinada interpretação. A Definição 1.9 estabelece uma terminologia que se baseia no valor associado às fórmulas.

Definição 1.9

1. Se uma fórmula α tem valor v em uma certa interpretação I, diz-se que α é *verdadeira* na interpretação I.
2. Uma fórmula α é *satisfatível* (ou *consistente*) se existe pelo menos uma interpretação I tal que $I[\alpha] = v$.
3. Uma fórmula é denominada *válida* quando for verdadeira em todas as interpretações possíveis. Essas fórmulas são conhecidas também como *tautologias*.
4. Se uma fórmula α tem valor f em uma certa interpretação I, diz-se que α é *falsa* na interpretação I.
5. Uma fórmula α é *inválida* se existe pelo menos uma interpretação I tal que $I[\alpha] = f$.
6. Uma fórmula α é *insatisfatível* (ou inconsistente) quando for falsa em todas as interpretações possíveis. Essas fórmulas são conhecidas também como *contradições*.

7. Na Lógica Proposicional, as fórmulas que não são nem tautologias nem contradições são comumente chamadas de *contingentes*.
8. Dada uma fórmula α e uma interpretação I, então I *satisfaz* α se $I[\alpha] = v$.
9. Um conjunto de fórmulas $C=\{\alpha_1, \alpha_2, \alpha_3, ...\alpha_n\}$ é *satisfatível* se e somente se existe uma interpretação I tal que $I[\alpha_1] = I[\alpha_2] = I[\alpha_3] = ... = I[\alpha_n] = v$. Neste caso, I satisfaz o conjunto de fórmulas C.

Exemplo 1.9 Considere a fórmula α: $(p \vee q) \rightarrow (p \wedge q)$. Como essa fórmula possui dois átomos, ela admite $2^2 = 4$ interpretações, como mostra a Tabela 1.5.

Tabela 1.5 Tabela-verdade da fórmula α: $(p \vee q) \rightarrow (p \wedge q)$.

	p	q	$(p \vee q)$	$(p \wedge q)$	α
I_1	v	v	v	v	v
I_2	v	f	v	f	f
I_3	f	v	v	f	f
I_4	f	f	f	f	v

Com relação à fórmula α, pode-se dizer que:

- é verdadeira nas interpretações I_1 e I_4;
- é satisfatível, dado que existe pelo menos uma (neste caso, existem duas) interpretação cuja fórmula é v;
- é falsa nas interpretações I_2 e I_3;
- é inválida, dado que é falsa nas interpretações I_2 e I_3;
- é uma fórmula contingente, dado que não é uma tautologia e tampouco uma contradição.

Exemplo 1.10 Considere a fórmula β: $(p \wedge q \wedge r) \rightarrow r$. Como essa fórmula possui três átomos, ela admite $2^3 = 8$ interpretações, como mostra a Tabela 1.6.

Tabela 1.6 Tabela-verdade da fórmula β: $(p \wedge q \wedge r) \rightarrow r$.

	p	q	r	$p \wedge q$	$(p \wedge q \wedge r)$	β
I_1	v	v	v	v	v	v
I_2	v	v	f	v	f	v
I_3	v	f	v	f	f	v
I_4	v	f	f	f	f	v
I_5	f	v	v	f	f	v
I_6	f	v	f	f	f	v

Tabela 1.6 *Continuação...*

	p	q	r	$p \wedge q$	$(p \wedge q \wedge r)$	β
I_7	f	f	v	f	f	v
I_8	f	f	f	f	f	v

Como para qualquer interpretação I_i, i=1,...,8, $I_i[\beta]$ é v, a fórmula em questão é uma tautologia.

Exemplo 1.11 Considere a fórmula γ: $(p \vee \neg p) \rightarrow (q \wedge \neg q)$. Como essa fórmula possui dois átomos, ela admite $2^2 = 4$ interpretações, como mostra a Tabela 1.7.

Tabela 1.7 Tabela-verdade da fórmula γ: $(p \vee \neg p) \rightarrow (p \wedge \neg q)$.

	p	q	$\neg p$	$\neg q$	$p \vee \neg p$	$q \wedge \neg q$	γ
I_1	v	v	f	f	v	f	f
I_2	v	f	f	v	v	f	f
I_3	f	v	v	f	v	f	f
I_4	f	f	v	v	v	f	f

Como para qualquer interpretação I_i, i=1,...,4, $I_i[\beta]$ é f, a fórmula em questão é uma contradição.

Exemplo 1.12 Considere o conjunto de fórmulas $C=\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$, tal que:

$$\alpha_1: (p \rightarrow q)$$

$$\alpha_2: ((\neg p) \wedge (\neg q))$$

$$\alpha_3: ((\neg q) \wedge p)$$

$$\alpha_4: p$$

O conjunto C não é *satisfatível,* pois, como pode ser visto na Tabela 1.8, não existe uma interpretação, entre as quatro possíveis, que torna as quatro fórmulas simultaneamente v.

Tabela 1.8 Conjunto $C=\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$, tal que α_1: $(p \rightarrow q)$, α_2: $((\neg p) \wedge (\neg q))$, α_3: $((\neg q) \wedge p)$ e α_4: p não são satisfatíveis.

	p	q	$\neg p$	$\neg q$	α_1	α_2	α_3	α_4
I_1	v	v	f	f	v	f	f	v
I_2	v	f	f	v	f	f	v	v
I_3	f	v	v	f	v	f	f	f
I_4	f	f	v	v	v	v	f	f

Exemplo 1.13 Considere o conjunto de fórmulas $C=\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$, tal que:

α_1: $(p \rightarrow q)$

α_2: $(r \vee (\neg q))$

α_3: $(s \vee (\neg r))$

α_4: p

O conjunto C é *satisfatível*, pois, como pode ser visto na Tabela 1.9, existe uma interpretação (I_1) que torna todas as quatro fórmulas v.

Tabela 1.9 Conjunto C=$\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$, tal que α_1: $(p \rightarrow q)$, α_2: $(r \vee (\neg q))$, α_3: $(s \vee (\neg r))$ e α_4: p são satisfatíveis.

	p	q	r	s	$\neg q$	$\neg r$	α_1	α_2	α_3	α_4
I_1	v	v	v	v	f	f	v	v	v	v
I_2	v	v	v	f	f	f	v	v	f	v
I_3	v	v	f	v	f	v	v	f	v	v
I_4	v	v	f	f	f	v	v	f	v	v
I_5	v	f	v	v	v	f	f	v	v	v
I_6	v	f	v	f	v	f	f	v	f	v
I_7	v	f	f	v	v	v	f	v	v	v
I_8	v	f	f	f	v	v	f	v	v	v
I_9	f	v	v	v	f	f	v	v	v	f
I_{10}	f	v	v	f	f	f	v	v	f	f
I_{11}	f	v	f	v	f	v	v	f	v	f
I_{12}	f	v	f	f	f	v	v	f	v	f
I_{13}	f	f	v	v	v	f	f	v	v	f
I_{14}	f	f	v	f	v	f	f	v	f	f
I_{15}	f	f	f	v	v	v	f	v	v	f
I_{16}	f	f	f	f	v	v	f	v	v	f

Exemplo 1.14 A Tabela 1.10 evidencia que a wff $(\neg(p \wedge q) \vee q)$ é uma tautologia.

Tabela 1.10 Tabela-verdade da wff $(\neg(p \wedge q) \vee q)$.

	p	q	$(p \wedge q)$	$\neg(p \wedge q)$	$(\neg(p \wedge q) \vee q)$
I_1	v	v	v	f	v
I_2	v	f	f	v	v

Tabela 1.10 *Continuação...*

	p	q	$(p \wedge q)$	$\neg(p \wedge q)$	$(\neg(p \wedge q) \vee q)$
I_3	f	v	f	v	v
I_4	f	f	f	v	v

Exemplo 1.15 A fórmula tautológica $(p \vee (\neg p))$ tem por tabela-verdade a Tabela 1.11. Essa fórmula é chamada de *lei do meio excluído*, que estabelece que ou p é verdade ou falso e todo o resto é excluído. A Tabela 1.11 prova essa lei mostrando que é interpretada v sob toda possível interpretação. Como a fórmula tem apenas um átomo, o número de possíveis interpretações para essa fórmula é apenas $2^1 = 2$.

Tabela 1.11 Tabela-verdade da wff $(p \vee (\neg p))$.

	p	$(\neg p)$	$(p \vee (\neg p))$
I_1	v	f	v
I_2	f	v	v

Exemplo 1.16 Duas fórmulas α_1: (falso $\rightarrow$ p) e α_2: (p$\rightarrow$ verdade) são evidenciadas como tautologias na Tabela 1.12.

Tabela 1.12 Tabela-verdade das wffs α_1: (falso $\rightarrow$ p) e α_2 = (p$\rightarrow$ verdade).

	p	(falso $\rightarrow$ p)	(p$\rightarrow$ verdade)
I_1	v	v	v
I_2	f	v	v

Observação 1.12 Se α for uma tautologia que contém uma subfórmula β, uma nova expressão lógica α' pode ser construída por meio da substituição de todas as ocorrências de β por uma expressão arbitrária. A expressão lógica resultante α' continua sendo uma tautologia. Considere a wff tautológica α: $(p \vee (\neg p))$ do Exemplo 1.15 (Tabela 1.11). Considere a substituição de todas as ocorrências de p em α por qualquer wff arbitrária, por exemplo, $(p \wedge q)$. A nova wff, $\alpha' = ((p \wedge q) \vee (\neg(p \wedge q)))$ é também tautológica, como mostra a Tabela 1.13.

Tabela 1.13 Tabela-verdade da wff $\alpha' = ((p \wedge q) \vee (\neg(p \wedge q)))$.

	p	q	$(p \wedge q)$	$\neg(p \wedge q)$	α'
I_1	v	v	v	f	v
I_2	v	f	f	v	v
I_3	f	v	f	v	v
I_4	f	f	f	v	v

Substituições semelhantes podem ser feitas em qualquer outra tautologia. Isso se deve ao fato de que, na avaliação do valor-verdade de qualquer expressão, apenas o valor-verdade de suas subexpressões imediatas tem efeito. O fato de o valor em questão ter sido obtido diretamente pela interpretação ou por algum outro tipo de avaliação anterior é irrelevante.

Definição 1.10 (O princípio de substituição) Se α for uma fórmula tendo β como subfórmula, o valor de α não muda se β for substituída por uma expressão que tenha os mesmos valores-verdade que β. Se α for uma tautologia, α permanece uma tautologia independentemente da interpretação de β ser v ou f.

Exemplo 1.17 Considere a wff α: $((p \lor q) \land (p \lor r))$. Considere a subfórmula β: $(p \lor q)$ e considere a fórmula α': $(((\neg p) \to q) \land (p \lor r))$ obtida pela substituição de β por uma fórmula com os mesmos valores que β em cada interpretação, no caso, a fórmula β'= $((\neg p) \to q)$. A Tabela 1.14 evidencia que as fórmulas α e α' têm o mesmo valor sob cada uma das oito interpretações, evidenciando o princípio de substituição.

Tabela 1.14 Tabela-verdade das wffs α e α', evidenciando o princípio da substituição.

	p	q	r	$\neg p$	$(\neg p)\to q$	$(p \lor q)$	$(p \lor r)$	α	α'
I_1	v	v	v	f	v	v	v	v	v
I_2	v	v	f	f	v	v	v	v	v
I_3	v	f	v	f	v	v	v	v	v
I_4	v	f	f	f	v	v	v	v	v
I_5	f	v	v	v	v	v	v	v	v
I_6	f	v	f	v	v	v	f	f	f
I_7	f	f	v	v	f	f	v	f	f
I_8	f	f	f	v	f	f	f	f	f

Exemplo 1.18 O fato de a wff α: $(\neg(p \land q) \lor q)$ ser tautológica (ver Exemplo 1.14) permite dizer que a wff α': $(\neg((p \lor q) \land r) \lor q)$ é também tautológica, como evidenciado na Tabela 1.15. Observe que a forma sentencial associada a ambas é $(\neg(\alpha_1 \land \alpha_2) \lor \alpha_2)$. Na fórmula α, α_1: p e α_2: q. Na fórmula α', α_1: $(p \lor q)$ e α_2: r.

Tabela 1.15 Tabela-verdade das wffs α e α', com α tautológica e γ: $\neg((p \lor q) \land r)$.

	p	q	r	$(p \lor q)$	$(p \lor q) \land r$	γ	α	α'
I_1	v	v	v	v	v	f	v	v
I_2	v	v	v	v	v	f	v	v
I_3	v	f	f	v	f	v	v	v

Tabela 1.15 *Continuação...*

	p	q	r	$(p \vee q)$	$(p \vee q) \wedge r$	γ	α	α'
I_4	v	f	f	v	f	v	v	v
I_5	f	v	v	v	v	f	v	v
I_6	f	v	v	v	v	f	v	v
I_7	f	f	f	f	f	v	v	v
I_8	f	f	f	f	f	v	v	v

1.4 CONSEQÜÊNCIA LÓGICA E EQUIVALÊNCIA LÓGICA

Definição 1.11 Dadas as fórmulas $\beta_1, \beta_2, \beta_3,..., \beta_n$ e uma fórmula α, diz-se que α é *conseqüência lógica* de $\beta_1, \beta_2, \beta_3,..., \beta_n$ se e somente se em qualquer interpretação em que $\beta_1, \beta_2, \beta_3,..., \beta_n$ forem simultaneamente v, α é também v. Se α é conseqüência lógica de $\beta_1, \beta_2, \beta_3,..., \beta_n$, diz-se que α segue logicamente de $\beta_1, \beta_2, \beta_3,..., \beta_n$. Para indicar esse fato, a seguinte notação é usada:

$$\beta_1, \beta_2, \beta_3,..., \beta_n \models \alpha$$

Exemplo 1.19 Como pode ser comprovado na Tabela 1.16, a fórmula $(p \vee q)$ é conseqüência lógica da fórmula p, ou seja, toda interpretação que torna p v, torna $(p \vee q)$ v também. De maneira análoga, pode-se dizer que $(p \vee q)$ é conseqüência lógica da fórmula q, ou seja, toda interpretação que torna q v, torna $(p \vee q)$ v também.

Tabela 1.16 Tabela-verdade que evidencia $p \models (p \vee q)$ e também $q \models (p \vee q)$.

	p	q	$(p \vee q)$
I_1	v	v	v
I_2	v	f	v
I_3	f	v	v
I_4	f	f	f

Exemplo 1.20 Considere as fórmulas α: $(p \rightarrow q)$, β: $(r \rightarrow s)$ e γ: $(p \vee r) \rightarrow (q \vee s)$. Verificar se $\alpha, \beta \models \gamma$, isto é, se γ é conseqüência lógica de α e β. Para essa verificação, considere a Tabela 1.17, com todas as 16 possíveis interpretações.

Tabela 1.17 Evidenciando a conseqüência lógica $(p \rightarrow q), (r \rightarrow s) \models (p \vee r) \rightarrow (q \vee s)$.

	p	q	r	s	$(p \rightarrow q)$	$(r \rightarrow s)$	$(p \vee r)$	$(q \vee s)$	$(p \vee r) \rightarrow (q \vee s)$	
I_1	v	v	v	v	v	v	v	v	v	←
I_2	v	v	v	f	v	f	v	v	v	

Tabela 1.17 *Continuação...*

	p	q	r	s	(p → q)	(r → s)	(p ∨ r)	(q ∨ s)	(p ∨ r) → (q ∨ s)	
I_3	v	v	f	v	v	v	v	v	v	←
I_4	v	v	f	f	v	v	v	v	v	←
I_5	v	f	v	v	f	v	v	v	v	
I_6	v	f	v	f	f	f	v	f	f	
I_7	v	f	f	v	f	v	v	v	v	
I_8	v	f	f	f	f	v	v	f	f	
I_9	f	v	v	v	v	v	v	v	v	←
I_{10}	f	v	v	f	v	f	v	v	v	
I_{11}	f	v	f	v	v	v	f	v	v	←
I_{12}	f	v	f	f	v	v	f	v	v	←
I_{13}	f	f	v	v	v	v	v	v	v	←
I_{14}	f	f	v	f	v	f	v	f	f	
I_{15}	f	f	f	v	v	v	f	v	v	←
I_{16}	f	f	f	f	v	v	f	f	v	←

Note na Tabela 1.17 que, nas interpretações nas quais ambas α e β são interpretadas v (ou seja, interpretações I_1, I_3, I_4, I_{11}, I_{12}, I_{13}, I_{15} e I_{16}, assinaladas na tabela com uma flecha), a fórmula γ é também interpretada v. Isto permite dizer que γ é uma conseqüência lógica de α e β, ou seja, pode-se escrever $\alpha, \beta \models \gamma$. Por meio da inspeção da Tabela 1.17 pode-se dizer que a fórmula $\alpha \wedge \beta$ não é conseqüência lógica da fórmula γ. Isso se deve ao fato de existirem interpretações que tornam γ v e, nessas mesmas interpretações, $\alpha \wedge \beta$ não é v (interpretações I_2, I_5, I_7 e I_{10}).

Definição 1.12 Diz-se que uma fórmula α é logicamente *equivalente* ($\equiv$) a uma fórmula β se e somente se α for conseqüência lógica de β e β for conseqüência lógica de α, isto é:

$$\alpha \equiv \beta \text{ se e somente se } \alpha \models \beta \text{ e } \beta \models \alpha$$

Exemplo 1.21 Usando a Definição 1.12, mostre que $(p \to q) \equiv (\neg p \vee q)$. Para tal, é preciso mostrar que:

(1) $(p \to q) \models (\neg p \vee q)$, ou seja, que $(\neg p \vee q)$ é conseqüência lógica de $(p \to q)$.

(2) $(\neg p \vee q) \models (p \to q)$, ou seja, que $(p \to q)$ é conseqüência lógica de $(\neg p \vee q)$.

Mostrar (1) é mostrar que, sempre que $(p \to q)$ for interpretada v, $(\neg p \vee q)$ é também interpretada v. Isso está evidente na Tabela 1.18.

Tabela 1.18 Tabela-verdade que evidencia $(p \rightarrow q) \models (\neg p \vee q)$.

	p	q	$(p \rightarrow q)$	$\neg p$	$(\neg p \vee q)$	
I_1	v	v	**v**	f	**v**	←
I_2	v	f	f	f	f	
I_3	f	v	**v**	v	**v**	←
I_4	f	f	**v**	v	**v**	←

Como pode ser visto na Tabela 1.18, a fórmula $(p \rightarrow q)$ é interpretada v nas interpretações I_1, I_3 e I_4. Note que nessas interpretações a fórmula $(\neg p \vee q)$ é também interpretada v, o que permite escrever $(p \rightarrow q) \models (\neg p \vee q)$.

Mostrar (2) é mostrar que, sempre que $(\neg p \vee q)$ for interpretada v, $(p \rightarrow q)$ é também interpretada v. Isso está evidente na Tabela 1.19.

Tabela 1.19 Tabela-verdade que evidencia $(\neg p \vee q) \models (p \rightarrow q)$.

	p	q	$\neg p$	$(\neg p \vee q)$	$(p \rightarrow q)$	
I_1	v	v	f	**v**	**v**	←
I_2	v	f	f	f	f	
I_3	f	v	v	**v**	**v**	←
I_4	f	f	v	**v**	**v**	←

Como pode ser visto na Tabela 1.19, a fórmula $(\neg p \vee q)$ é interpretada v nas interpretações I_1, I_3 e I_4. Note que nessas interpretações a fórmula $(p \rightarrow q)$ é também interpretada v, o que permite escrever que $(\neg p \vee q) \models (p \rightarrow q)$. Considerando que (1) e (2) foram provadas, pode-se escrever que $(p \rightarrow q) \equiv (\neg p \vee q)$.

Teorema 1.1 Dadas as fórmulas $\beta_1, \beta_2, \beta_3, \ldots, \beta_n$ e uma fórmula α, diz-se que α é conseqüência lógica de $\beta_1, \beta_2, \beta_3, \ldots, \beta_n$ se e somente se a fórmula

$$\beta_1 \wedge \beta_2 \wedge \beta_3 \wedge \ldots \wedge \beta_n \rightarrow \alpha \text{ for uma tautologia.}$$

Prova

(1) Sejam as fórmulas $\beta_1, \beta_2, \beta_3, \ldots, \beta_n$ e α e considere que α seja conseqüência lógica de $\beta_1, \beta_2, \beta_3, \ldots, \beta_n$. Seja I uma interpretação qualquer. Duas situações podem ocorrer:

(1.1) se $\beta_1, \beta_2, \beta_3, \ldots, \beta_n$ forem todas interpretadas v em I, então α é também v em I, dado que é conseqüência lógica dos $\beta_1, \beta_2, \beta_3, \ldots, \beta_n$. Portanto, $\beta_1 \wedge \beta_2 \wedge \beta_3 \wedge \ldots \wedge \beta_n \rightarrow \alpha$ é v em I.

(1.2) se um dos β_i (i=1, ..., n) for f em I, $\beta_1 \wedge \beta_2 \wedge \beta_3 \wedge \ldots \wedge \beta_n$ será também f em I. Independentemente de o valor de α em I ser v ou f, a fórmula $\beta_1 \wedge \beta_2 \wedge \beta_3 \wedge \ldots \wedge \beta_n \rightarrow \alpha$ é v em I, dado que uma implicação é sempre interpretada v se seu antecedente for interpretado f.

De **(1.1)** e **(1.2)** tem-se que $\beta_1 \wedge \beta_2 \wedge \beta_3 \wedge ... \wedge \beta_n \rightarrow \alpha$ é v em qualquer interpretação, ou seja, $\beta_1 \wedge \beta_2 \wedge \beta_3 \wedge ... \wedge \beta_n \rightarrow \alpha$ é uma tautologia.

(2) Do fato de $\beta_1 \wedge \beta_2 \wedge \beta_3 \wedge ... \wedge \beta_n \rightarrow \alpha$ ser uma tautologia, tem-se que $\beta_1 \wedge \beta_2 \wedge \beta_3 \wedge ... \wedge \beta_n \rightarrow \alpha$ é v em qualquer interpretação. Para que isso aconteça, se $\beta_1 \wedge \beta_2 \wedge \beta_3 \wedge ... \wedge \beta_n$ for v em I, α também deve ser v em I, ou seja, α é conseqüência lógica de $\beta_1, \beta_2, \beta_3, ..., \beta_n$.

Exemplo 1.22 Considere novamente que $(p \rightarrow q) \equiv (\neg p \vee q)$ (ou seja, que $(p \rightarrow q) \models (\neg p \vee q)$ e que $(\neg p \vee q) \models (p \rightarrow q)$) só que, desta vez, usando o resultado estabelecido pelo Teorema 1.1. O problema novamente é mostrar que $(p \rightarrow q) \equiv (\neg p \vee q)$. Para tal, é preciso mostrar que:

1. $(p \rightarrow q) \models (\neg p \vee q)$ o que, pelo Teorema 1.1, significa mostrar que $(p \rightarrow q) \rightarrow (\neg p \vee q)$ é uma tautologia (ver Tabela 1.20).
2. $(\neg p \vee q) \models (p \rightarrow q)$ o que, pelo Teorema 1.1, significa mostrar que $(\neg p \vee q) \rightarrow (p \rightarrow q)$ é uma tautologia (ver Tabela 1.21).

Tabela 1.20 Tabela-verdade que evidencia a tautologia $(p \rightarrow q) \rightarrow (\neg p \vee q)$.

	p	q	$(p \rightarrow q)$	$\neg p$	$(\neg p \vee q)$	$(p \rightarrow q) \rightarrow (\neg p \vee q)$
I_1	v	v	v	f	v	v
I_2	v	f	f	f	f	v
I_3	f	v	v	v	v	v
I_4	f	f	v	v	v	v

Tabela 1.21 Tabela-verdade que evidencia a tautologia $(\neg p \vee q) \rightarrow (p \rightarrow q)$.

	p	q	$\neg p$	$(\neg p \vee q)$	$(p \rightarrow q)$	$(\neg p \vee q) \rightarrow (p \rightarrow q)$
I_1	v	v	f	v	v	v
I_2	v	f	f	f	f	v
I_3	f	v	v	v	v	v
I_4	f	f	v	v	v	v

Observação 1.13 A definição de fórmulas equivalentes (Definição 1.12) pode ser reescrita considerando o estabelecido pelo Teorema 1.1. A Definição 1.12 estabelece que duas fórmulas **α e β** são **equivalentes** se ambas as conseqüências lógicas $\alpha \models \beta$ e $\beta \models \alpha$ forem satisfeitas.

Considerando o Teorema 1.1, $\alpha \models \beta$ se e somente se $\alpha \rightarrow \beta$ for uma tautologia e $\beta \models \alpha$ se e **somente se** $\beta \rightarrow \alpha$ for uma tautologia. Portanto, duas fórmulas α e β são equivalentes $(\alpha \equiv \beta)$ se **e somente se** $\alpha \rightarrow \beta$ for uma tautologia e $\beta \rightarrow \alpha$ for uma tautologia, ou seja,

$(\alpha \equiv \beta)$ se e somente se $\alpha \leftrightarrow \beta$ for uma tautologia.

Diz-se que duas fórmulas α e β são equivalentes se e somente se os valores-verdade de α e β coincidirem para qualquer interpretação.

Exemplo 1.23 Considerando a Observação 1.13, uma outra abordagem para evidenciar a equivalência lógica $(p \rightarrow q) \equiv (\neg p \vee q)$ é mostrar que a fórmula $(p \rightarrow q) \leftrightarrow (\neg p \vee q)$ é uma tautologia, como feito na Tabela 1.22.

Tabela 1.22 Tabela-verdade que evidencia a tautologia $(p \rightarrow q) \leftrightarrow (\neg p \vee q)$.

	p	q	$(p \rightarrow q)$	$\neg p$	$(\neg p \vee q)$	$(p \rightarrow q) \leftrightarrow (\neg p \vee q)$
I_1	v	v	v	f	v	v
I_2	v	f	f	f	f	v
I_3	f	v	v	v	v	v
I_4	f	f	v	v	v	v

Teorema 1.2 Dadas as fórmulas $\beta_1, \beta_2, \beta_3, \ldots, \beta_n$ e uma fórmula α, diz-se que α é conseqüência lógica de $\beta_1, \beta_2, \beta_3, \ldots, \beta_n$ se e somente se a fórmula

$$\beta_1 \wedge \beta_2 \wedge \beta_3 \wedge \ldots \wedge \beta_n \wedge \neg\alpha \text{ for uma contradição.}$$

Prova Sabe-se pelo Teorema 1.1 que a fórmula α é conseqüência lógica das fórmulas $\beta_1, \beta_2, \beta_3, \ldots, \beta_n$ se e somente se $\beta_1 \wedge \beta_2 \wedge \beta_3 \wedge \ldots \wedge \beta_n \rightarrow \alpha$ for uma tautologia. Equivalentemente, α é conseqüência lógica das fórmulas $\beta_1, \beta_2, \beta_3, \ldots, \beta_n$ se e somente se a negação de $\beta_1 \wedge \beta_2 \wedge \beta_3 \wedge \ldots \wedge \beta_n \rightarrow \alpha$ for uma contradição. Mas,

$$\neg(\beta_1 \wedge \beta_2 \wedge \beta_3 \wedge \ldots \wedge \beta_n \rightarrow \alpha) \equiv$$

$$\neg(\neg(\beta_1 \wedge \beta_2 \wedge \beta_3 \wedge \ldots \wedge \beta_n) \vee \alpha) \equiv$$

$$\beta_1 \wedge \beta_2 \wedge \beta_3 \wedge \ldots \wedge \beta_n \wedge \neg\alpha$$

ou seja, $\beta_1 \wedge \beta_2 \wedge \beta_3 \wedge \ldots \wedge \beta_n \wedge \neg\alpha$ é uma contradição.

Observação 1.14 No Exemplo 1.22, o Teorema 1.1 foi usado para evidenciar conseqüência lógica como parte do processo de verificação de equivalência lógica. A seguir, o Teorema 1.2 é usado com o mesmo propósito, para a mesma equivalência lógica. Tem-se, pois, que:

1. $(p \rightarrow q) \models (\neg p \vee q)$ o que, pelo Teorema 1.2, significa mostrar que $(p \rightarrow q) \wedge (\neg(\neg p \vee q))$ é uma contradição (ver Tabela 1.23);
2. $(\neg p \vee q) \models (p \rightarrow q)$ o que, pelo Teorema 1.2, significa mostrar que $(\neg p \vee q) \wedge (\neg(p \rightarrow q))$ é uma contradição (ver Tabela 1.24).

Tabela 1.23 Tabela-verdade que evidencia a contradição $(p \to q) \land (\neg(\neg p \lor q))$.

	p	q	$(p \to q)$	$\neg p$	$(\neg p \lor q)$	$\neg(\neg p \lor q)$	$(p \to q) \land (\neg(\neg p \lor q))$
I_1	v	v	v	f	v	f	f
I_2	v	f	f	f	f	v	f
I_3	f	v	v	v	v	f	f
I_4	f	f	v	v	v	f	f

Tabela 1.24 Tabela-verdade que evidencia a contradição $(\neg p \lor q) \land (\neg(p \to q))$.

	p	q	$\neg p$	$(\neg p \lor q)$	$(p \to q)$	$\neg(p \to q)$	$(\neg p \lor q) \land (\neg(p \to q))$
I_1	v	v	f	v	v	f	f
I_2	v	f	f	f	f	v	f
I_3	f	v	v	v	v	f	f
I_4	f	f	v	v	v	f	f

Observação 1.15 Os dois metateoremas anteriores, Teorema 1.1 e Teorema 1.2, são muito importantes. Eles garantem que provar que uma fórmula é conseqüência lógica de um conjunto finito de fórmulas é equivalente a provar que uma fórmula relacionada é uma tautologia ou contradição. Para algumas estratégias de provas usa-se o Teorema 1.1, chamado de *Teorema da Dedução* ou de *Admissão de Premissas*. Outra estratégia de prova, conhecida como *Redução ao Absurdo*, é estabelecida pelo Teorema 1.2.

Observação 1.16 É importante deixar claro que os símbolos de conseqüência lógica ($\models$) e de equivalência lógica ($\equiv$) não são partes do conjunto de símbolos da Lógica Proposicional, mas sim parte da metalinguagem (i.e., uma linguagem usada para descrever outra) usada para descrever certas situações que nela ocorrem.

Exemplo 1.24 Considere as fórmulas α: $(p \lor q) \to r$ e β: $(p \to r) \land (q \to r)$. A seguir, são abordadas as várias maneiras de provar que $\alpha \equiv \beta$. A Tabela 1.25 evidencia os resultados necessários.

Tabela 1.25 Tabela-verdade que evidencia $\alpha \equiv \beta$ para α: $(p \lor q) \to r$ e β: $(p \to r) \land (q \to r)$.

	p	q	r	$(p \lor q)$	α	$(p \to r)$	$(q \to r)$	β	$\alpha \to \beta$	$\alpha \land (\neg\beta)$	$\beta \to \alpha$	$\beta \land (\neg\alpha)$	$\alpha \leftrightarrow \beta$
I_1	v	v	v	v	v	v	v	v	v	f	v	f	v
I_2	v	v	f	v	f	f	f	f	v	f	v	f	v
I_3	v	f	v	v	v	v	v	v	v	f	v	f	v
I_4	v	f	f	v	f	f	v	f	v	f	v	f	v
I_5	f	v	v	v	v	v	v	v	v	f	v	f	v

Tabela 1.25 *Continuação...*

	p	q	r	$(p \vee q)$	α	$(p \to r)$	$(q \to r)$	β	$\alpha \to \beta$	$\alpha \wedge (\neg\beta)$	$\beta \to \alpha$	$\beta \wedge (\neg\alpha)$	$\alpha \leftrightarrow \beta$
I_6	f	v	f	v	f	v	f	f	v	f	v	f	v
I_7	f	f	v	f	v	v	v	v	v	f	v	f	v
I_8	f	f	f	f	v	v	v	v	v	f	v	f	v

Usando a definição, $\alpha \equiv \beta$ se

(a) $\beta \models \alpha$ (ou seja, α é conseqüência lógica de β) e

(b) $\alpha \models \beta$ (ou seja, β é conseqüência lógica de α).

(a) Três abordagens para evidenciar $\beta \models \alpha$:

(a.1) considerando as oito interpretações (I_1,..., I_8), verifica-se na Tabela 1.25 que todas as interpretações que tornam β v (no caso, as interpretações I_1, I_3, I_5, I_7 e I_8) tornam α v também. Pode-se dizer, pois, que α é conseqüência lógica de β;

(a.2) o Teorema 1.1 pode ser usado para estabelecer (ou não) a conseqüência lógica, em que basta verificar se $\beta \to \alpha$ é uma tautologia (12ª coluna da Tabela 1.25);

(a.3) o Teorema 1.2 também pode ser usado, mostrando que $\beta \wedge (\neg\alpha)$ é uma contradição (13ª coluna da Tabela 1.25).

(b) Três abordagens para evidenciar $\alpha \models \beta$:

(b.1) considerando as oito interpretações (I_1,..., I_8), verifica-se na Tabela 1.25 que todas as interpretações que tornam α v (no caso, as interpretações I_1, I_3, I_5, I_7 e I_8) tornam β v também. Pode-se dizer, pois, que β é conseqüência lógica de α;

(b.2) o Teorema 1.1 pode ser usado para estabelecer (ou não) a conseqüência lógica, em que basta verificar se $\alpha \to \beta$ é uma tautologia (10ª coluna da Tabela 1.25);

(b.3) o Teorema 1.2 também pode ser usado, mostrando que $\alpha \wedge (\neg\beta)$ é uma contradição (11ª coluna da Tabela 1.25).

Note que a equivalência $\alpha \equiv \beta$ também pode ser provada (ver Observação 1.13), mostrando que $\alpha \leftrightarrow \beta$ é uma tautologia (última coluna da Tabela 1.25).

Exemplo 1.25 Considere as fórmulas α: $p \to (q \wedge r)$ e β: $(p \to q) \wedge (p \to r)$. A seguir, são abordadas as várias maneiras de mostrar que $\alpha \equiv \beta$. A Tabela 1.26 evidencia os resultados necessários.

Tabela 1.26 Tabela-verdade que evidencia $\alpha \equiv \beta$ para α: $p \to (q \wedge r)$ e β: $(p \to q) \wedge (p \to r)$.

	p	q	r	$(q \wedge r)$	α	$(p \to q)$	$(p \to r)$	β	$\alpha \to \beta$	$\alpha \wedge (\neg\beta)$	$\beta \to \alpha$	$\beta \wedge (\neg\alpha)$	$\alpha \leftrightarrow \beta$
I_1	v	v	v	v	v	v	v	v	v	f	v	f	v
I_2	v	v	f	f	f	v	f	f	v	f	v	f	v
I_3	v	f	v	f	f	f	v	f	v	f	v	f	v
I_4	v	f	f	f	f	f	f	f	v	f	v	f	v
I_5	f	v	v	v	v	v	v	v	v	f	v	f	v
I_6	f	v	f	f	v	v	v	v	v	f	v	f	v
I_7	f	f	v	f	v	v	v	v	v	f	v	f	v
I_8	f	f	f	f	v	v	v	v	v	f	v	f	v

Usando a definição, $\alpha \equiv \beta$ se

(a) $\beta \models \alpha$ (ou seja, α é conseqüência lógica de β) e

(b) $\alpha \models \beta$ (ou seja, β é conseqüência lógica de α).

(a) Três abordagens para evidenciar $\beta \models \alpha$:

(a.1) considerando as oito interpretações (I_1,..., I_8), verifica-se na Tabela 1.26 que todas as interpretações que tornam β v (no caso, as interpretações I_1, I_5, I_6, I_7 e I_8) tornam α v também. Pode-se dizer, pois, que α é conseqüência lógica de β;

(a.2) o Teorema 1.1 pode ser usado para estabelecer (ou não) a conseqüência lógica, em que basta verificar se $\beta \to \alpha$ é uma tautologia (12ª coluna da Tabela 1.26);

(a.3) o Teorema 1.2 também pode ser usado, mostrando que $\beta \wedge (\neg\alpha)$ é uma contradição (13ª coluna da Tabela 1.26).

(b) Três abordagens para evidenciar $\alpha \models \beta$:

(b.1) considerando as oito interpretações (I_1, ..., I_8), verifica-se na Tabela 1.26 que todas as interpretações que tornam α v (no caso, as interpretações I_1, I_5, I_6, I_7 e I_8) tornam β v também. Pode-se dizer, pois, que β é conseqüência lógica de α;

(b.2) o Teorema 1.1 pode ser usado para estabelecer (ou não) a conseqüência lógica, em que basta verificar se $\alpha \to \beta$ é uma tautologia (10ª coluna da Tabela 1.26);

(b.3) o Teorema 1.2 também pode ser usado, mostrando que $\alpha \wedge (\neg\beta)$ é uma contradição (11ª coluna da Tabela 1.26).

Note que a equivalência $\alpha \equiv \beta$ também pode ser provada (ver Observação 1.13), mostrando que $\alpha \leftrightarrow \beta$ é uma tautologia (última coluna da Tabela 1.26).

1.5 ÁLGEBRA DA LÓGICA PROPOSICIONAL

A seguir, são discutidas equivalências importantes usadas principalmente para a simplificação e manipulação de expressões lógicas com vistas à prova da validade de argumentos. Algumas dessas equivalências (às vezes referenciadas como leis) foram já provadas anteriormente usando o método de tabela-verdade.

Note que, com exceção da lei da dupla negação, todas as leis listadas na Tabela 1.27 são abordadas em pares, denominados *pares duais*. Para cada expressão, o dual é encontrado substituindo todas as ocorrências dos símbolos verdade e falso por falso e verdade, respectivamente, bem como substituindo todo conectivo $\wedge$ por $\vee$ e todo conectivo $\vee$ por $\wedge$. As substituições devem ocorrer simultaneamente. A lei da dupla negação é seu próprio dual; todas as leis listadas na Tabela 1.27 têm um dual, e o mesmo vale para todos os resultados derivados dessas leis.

Tabela 1.27 Leis da negação, conjunção e disjunção.

Leis	Nome
$\alpha \wedge \neg\alpha \equiv$ falso	Lei da contradição
$\alpha \vee \neg\alpha \equiv$ verdade	Lei do meio excluído
$\alpha \wedge$ verdade $\equiv \alpha$ $\alpha \vee$ falso $\equiv \alpha$	Leis da identidade
$\alpha \wedge$ falso $\equiv$ falso $\alpha \vee$ verdade $\equiv$ verdade	Leis da dominação
$\alpha \wedge \alpha \equiv \alpha$ $\alpha \vee \alpha \equiv \alpha$	Leis idempotentes
$\neg(\neg\alpha) \equiv \alpha$	Lei da dupla negação
$\alpha \wedge \beta \equiv \beta \wedge \alpha$ $\alpha \vee \beta \equiv \beta \vee \alpha$	Leis comutativas
$(\alpha \wedge \beta) \wedge \gamma \equiv \alpha \wedge (\beta \wedge \gamma)$ $(\alpha \vee \beta) \vee \gamma \equiv \alpha \vee (\beta \vee \gamma)$	Leis associativas
$\alpha \wedge (\beta \vee \gamma) \equiv (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$ $\alpha \vee (\beta \wedge \gamma) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$	Leis distributivas
$\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$ $\neg(\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta$	Leis De Morgan

Observação 1.17 Um procedimento usualmente adotado na manipulação de expressões é a substituição (com base no princípio da substituição) de expressões envolvendo o conectivo condicional ($\rightarrow$) e o bicondicional ($\leftrightarrow$) por suas equivalentes lógicas, como mostra a Tabela 1.28.

Tabela 1.28 Equivalências da condicional e da bicondicional.

$(\alpha \to \beta)$	$\equiv$	$\neg\alpha \vee \beta$	(1)
$(\alpha \leftrightarrow \beta)$	$\equiv$	$(\alpha \to \beta) \wedge (\beta \to \alpha)$	(2)
$(\alpha \leftrightarrow \beta)$	$\equiv$	$(\alpha \to \beta) \wedge (\beta \to \alpha)$ $(\neg\alpha \vee \beta) \wedge (\neg\beta \vee \alpha)$	(3)

As equivalências (1), (2) e (3) da Tabela 1.28 são provadas nas Tabelas-verdade 1.29, 1.30 e 1.31 respectivamente.

Tabela 1.29 Tabela-verdade que evidencia a equivalência (1) da Tabela 1.28.

	α	β	$\neg\beta$	$(\alpha \to \beta)$	$(\neg\alpha \vee \beta)$	$(\alpha \to \beta) \leftrightarrow (\neg\beta \vee \alpha)$
I_1	v	v	f	v	f	v
I_2	v	f	f	f	v	v
I_3	f	v	v	v	f	v
I_4	f	f	v	v	f	v

Tabela 1.30 Tabela-verdade que evidencia a equivalência (2) da Tabela 1.28.

	α	β	$\neg\alpha$	$\neg\beta$	$(\alpha \leftrightarrow \beta)$	$(\alpha \to \beta)$	$(\beta \to \alpha)$	$(\alpha \to \beta) \wedge (\beta \to \alpha)$	$(\alpha \leftrightarrow \beta) \leftrightarrow$ $((\alpha \to \beta) \wedge (\beta \to \alpha))$
I_1	v	v	f	f	v	v	v	v	v
I_2	v	f	f	f	f	f	v	f	v
I_3	f	v	v	v	f	v	f	f	v
I_4	f	f	v	v	v	v	v	v	v

Tabela 1.31 Tabela-verdade que evidencia a equivalência (3) da Tabela 1.28.

	α	β	$\neg\alpha$	$\neg\beta$	$(\alpha \leftrightarrow \beta)$	$(\neg\alpha \vee \beta)$	$(\neg\beta \vee \alpha)$	$(\neg\alpha \vee \beta) \wedge (\neg\beta \vee \alpha)$	$(\alpha \leftrightarrow \beta) \leftrightarrow$ $(\neg\alpha \vee \beta) \wedge (\neg\beta \vee \alpha)$
I_1	v	v	f	f	v	v	v	v	v
I_2	v	f	f	f	f	f	v	f	v
I_3	f	v	v	v	f	v	f	f	v
I_4	f	f	v	v	v	v	v	v	v

A Tabela 1.32 mostra duas importantes leis (e respectivos duais) que também podem ser derivadas a partir das leis mostradas na Tabela 1.27. As Tabelas 1.33 e 1.34 provam as duas leis em questão.

Tabela 1.32 Equivalências importantes.

$\alpha \vee (\alpha \wedge \beta)$	$\equiv$	α	absorção
$\alpha \wedge (\alpha \vee \beta)$	$\equiv$	α	absorção
$(\alpha \wedge \beta) \vee (\neg\alpha \wedge \beta)$	$\equiv$	β	
$(\alpha \vee \beta) \wedge (\neg\alpha \vee \beta)$	$\equiv$	β	

Tabela 1.33 Prova das leis da absorção.

	$\equiv (\alpha \wedge \text{verdade}) \vee (\alpha \wedge \beta)$	identidade
$\alpha \vee (\alpha \wedge \beta)$	$\equiv \alpha \wedge (\text{verdade} \vee \beta)$	distributiva
	$\equiv \alpha \wedge \text{verdade}$	dominação
	$\equiv \alpha$	identidade
	$\equiv (\alpha \vee \text{falso}) \wedge (\alpha \vee \beta)$	identidade
$\alpha \wedge (\alpha \vee \beta)$	$\equiv \alpha \vee (\text{falso} \wedge \beta)$	distributiva
	$\equiv \alpha \vee \text{falso}$	dominação
	$\equiv \alpha$	identidade

Tabela 1.34 Prova da lei $(\alpha \wedge \beta) \vee (\neg\alpha \wedge \beta) \equiv \beta$ e sua dual.

	$\equiv ((\alpha \wedge \beta) \vee (\neg\alpha)) \wedge ((\alpha \wedge \beta) \vee \beta)$	distributiva
	$\equiv ((\alpha \vee (\neg\alpha)) \wedge (\beta \vee (\neg\alpha))) \wedge (\beta \vee (\alpha \wedge \beta))$	distributiva e comutativa
$(\alpha \wedge \beta) \vee (\neg\alpha \wedge \beta)$	$\equiv (\text{verdade} \wedge (\beta \vee (\neg\alpha))) \wedge (\beta \vee (\beta \wedge \alpha))$	meio excluído e comutativa
	$\equiv (\beta \vee (\neg\alpha)) \wedge \beta$	identidade e absorção
	$\equiv \beta \wedge (\beta \vee (\neg\alpha))$	comutativa
	$\equiv \beta$	absorção
	$((\alpha \vee \beta) \wedge (\neg\alpha)) \vee ((\alpha \vee \beta) \wedge \beta)$	distributiva
	$\equiv ((\alpha \vee \beta) \wedge (\neg\alpha)) \vee ((\alpha \vee \beta) \wedge \beta)$	distributiva e comutativa
	$\equiv ((\alpha \wedge (\neg\alpha)) \vee (\beta \wedge (\neg\alpha))) \vee (\beta \wedge (\alpha \vee \beta))$	contradição e comutativa
$(\alpha \vee \beta) \wedge (\neg\alpha \vee \beta)$	$\equiv (\text{falso} \vee (\beta \wedge (\neg\alpha))) \vee (\beta \wedge (\beta \vee \alpha))$	identidade e absorção
	$\equiv (\beta \wedge (\neg\alpha)) \vee \beta$	comutativa
	$\equiv \beta \vee (\beta \wedge (\neg\alpha))$	absorção
	$\equiv \beta$	

1.6 FORMAS NORMAIS

É possível observar que há várias maneiras de escrever uma mesma fórmula; as fórmulas equivalentes, a seguir, são duas representações lógicas do mesmo conceito: $(\alpha \rightarrow \beta) \wedge \gamma \equiv (\neg\alpha \vee \beta) \wedge \gamma$.

Muitas vezes, é conveniente adotar uma certa padronização na notação, a fim de poder expressar as fórmulas de uma maneira única; a padronização facilita tanto a identificação de uma fórmula quanto a comparação entre duas ou mais fórmulas.

Duas formas, denominadas Formas Normais – FN – , são particularmente utilizadas e conhecidas como Forma Normal Conjuntiva (FNC) e Forma Normal Disjuntiva (FND). Dada uma expressão da Lógica Proposicional, é sempre possível determinar uma expressão equivalente que esteja representada tanto na Forma Normal Conjuntiva quanto na Forma Normal Disjuntiva. Como ficará mais claro nas seções seguintes, freqüentemente é necessário transformar a expressão de uma fórmula para a forma normal.

1.6.1 Forma Normal Conjuntiva

Definição 1.13 Diz-se que uma fórmula proposicional α está na Forma Normal Conjuntiva (FNC) quando α for uma conjunção $\beta_1 \wedge \beta_2 \wedge \beta_3 \wedge \ldots \wedge \beta_n$, em que cada β_i $(1 \leq i \leq n)$ é uma *cláusula*, ou seja, é uma disjunção de literais ou um literal. Pode-se então dizer que uma fórmula α está na FNC se e somente se:

1. contém como conectivos apenas $\wedge$, $\vee$ e $\neg$;
2. $\neg$ só opera sobre proposições atômicas, isto é, não tem alcance sobre $\wedge$ e $\vee$;
3. não apresenta operadores de negação sucessivos como $\neg\neg$;
4. $\vee$ não tem alcance sobre $\wedge$, ou seja, não existem expressões tais como $p \vee (q \wedge r)$.

Se β é uma fórmula na Forma Normal Conjuntiva equivalente a α, então β é referenciada como FNC(α).

Exemplo 1.26
(a) Para a fórmula α: $(\neg p \vee q) \rightarrow r$, tem-se que:

$$\text{FNC}(\alpha)\text{: } (\neg p \vee \neg q \vee r) \wedge (p \vee \neg q \vee r) \wedge (p \vee q \vee r).$$

É fácil mostrar que uma FNC é tautológica se e somente se cada elemento da conjunção for tautológico.

(b) As seguintes fórmulas da Lógica Proposicional estão na FNC:

(b.1) $p \wedge (q \vee r)$

(b.2) $p \wedge$ verdade

(b.3) $\neg p \wedge (\neg q \vee \neg r) \wedge s$

Já a expressão $p \wedge (r \vee (p \wedge s))$ não está na FNC porque a disjunção $(r \vee (p \wedge s))$ contém uma conjunção como subfórmula. Para que uma expressão possa ser qualificada como uma expressão na FNC, nenhuma disjunção deve ter uma conjunção como subfórmula.

Procedimento 1.1 Obtenção da FNC.

Para a obtenção da FNC de uma fórmula não tautológica α com n átomos, procura-se na tabela-verdade de α as interpretações que avaliam α como f. Para cada uma dessas interpretações I_i ($1 \leq i \leq 2^n$) constrói-se uma disjunção da seguinte maneira: se na interpretação I_i o átomo p da fórmula α é avaliado como v, toma-se $\neg p$ e, se for avaliado como f, toma-se p. Em seguida, determina-se a conjunção das disjunções obtidas em cada uma das interpretações I_i. Se a fórmula α for uma tautologia, determina-se que FNC(α): $p \vee (\neg p)$, na qual p é uma fórmula atômica.

Exemplo 1.27

(a) Considere a fórmula α: $(\neg p \vee q) \rightarrow r$ e sua Tabela-verdade 1.35.

Tabela 1.35 Tabela-verdade da fórmula $(\neg p \vee q) \rightarrow r$.

	p	q	r	$(\neg p \vee q) \rightarrow r$	
I_1	v	v	v	v	
I_2	v	v	f	**f**	←
I_3	v	f	v	v	
I_4	v	f	f	v	
I_5	f	v	v	v	
I_6	f	v	f	**f**	←
I_7	f	f	v	v	
I_8	f	f	f	**f**	←

Focalizando as interpretações cujos valores-verdade de α são f, de acordo com o Procedimento 1.1, FNC(α): $(\neg p \vee \neg q \vee r) \wedge (p \vee \neg q \vee r) \wedge (p \vee q \vee r)$.

1.6.2 Forma Normal Disjuntiva

Definição 1.14 Diz-se que uma fórmula proposicional α está na Forma Normal Disjuntiva (FND) quando α for uma disjunção $\beta_1 \vee \beta_2 \vee \beta_3 \vee \ldots \vee \beta_n$, em que cada β_i ($1 \leq i \leq n$) é uma conjunção de literais ou um literal. Pode-se então dizer que uma fórmula α está na FND se e somente se:

1. contém como conectivos apenas $\wedge$, $\vee$ e $\neg$;
2. $\neg$ só opera sobre proposições atômicas, isto é, não tem alcance sobre $\wedge$ e $\vee$;
3. não apresenta operadores de negação sucessivos como $\neg\neg$;

4. $\wedge$ não tem alcance sobre $\vee$, ou seja, não existem expressões tais como $p \wedge (q \vee r)$.

Se β é uma fórmula na Forma Normal Disjuntiva equivalente a α, então β é referenciada como FND(α).

Exemplo 1.28 As seguintes fórmulas do Cálculo Proposicional estão na FND:

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

$$p \vee \text{verdade}$$

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

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

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

Já a expressão $\neg(p \wedge q) \vee r$ não está na FND, uma vez que ela contém uma subfórmula (não atômica) negada. Apenas subfórmulas atômicas podem estar negadas na representação em forma normal.

Procedimento 1.2 Obtenção da FND.

Para a obtenção da FND de uma fórmula não contraditória α com n átomos, procura-se na tabela-verdade de α as interpretações que avaliam α como v. Para cada uma dessas interpretações I_i ($1 \leq i \leq 2^n$) constrói-se uma conjunção da seguinte maneira: se na interpretação I_i o átomo p da fórmula α é avaliado como v, toma-se p e, se for avaliado como f, toma-se $\neg p$. Em seguida, determina-se a conjunção das disjunções obtidas em cada uma das interpretações I_i. Se a fórmula α for uma contradição, determina-se que FND(α): $p \wedge (\neg p)$, na qual p é uma fórmula atômica.

Exemplo 1.29 Considere uma fórmula α cuja tabela-verdade seja Tabela 1.36.

Tabela 1.36 Tabela-verdade da fórmula α.

	p	q	r	α	
I_1	v	v	v	**v**	$\leftarrow$
I_2	v	v	f	**v**	$\leftarrow$
I_3	v	f	v	f	
I_4	v	f	f	f	
I_5	f	v	v	f	
I_6	f	v	f	f	
I_7	f	f	v	f	
I_8	f	f	f	**v**	$\leftarrow$

Focalizando as interpretações cujos valores-verdade de α são v, de acordo com o Procedimento 1.2, FND(α): $(p \wedge q \wedge r) \vee (p \wedge q \wedge \neg r) \vee (\neg p \wedge \neg q \wedge \neg r)$.

1.6.3 Obtenção da FNC sem o uso de tabela-verdade

A obtenção da FNC de uma dada fórmula α pode ser conseguida por meio da substituição de subfórmulas de α por suas fórmulas equivalentes. Este processo é repetido até que a fórmula normal desejada seja obtida.

Procedimento 1.3

Para a obtenção da forma normal conjuntiva de uma fórmula α os seguintes passos devem ser seguidos, quando passíveis de serem aplicados.

(1) repetidamente usar as equivalências a seguir para a eliminação dos conectivos lógicos $\leftrightarrow$ e $\rightarrow$.

$$\alpha \leftrightarrow \beta \equiv ((\neg\alpha) \vee \beta) \wedge ((\neg\beta) \vee \alpha)$$
$$\alpha \rightarrow \beta \equiv ((\neg\alpha) \vee \beta)$$

(2)

(a) repetidamente utilizar a lei da dupla negação para a eliminação de negações múltiplas:

$$\neg(\neg\alpha) \equiv \alpha$$

(b) repetidamente utilizar as leis De Morgan para a redução do escopo da negação:

$$\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$$
$$\neg(\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta$$

(3) quando a expressão obtida não tiver subexpressões compostas negadas, as duas leis a seguir são utilizadas para reduzir o escopo do $\vee$.

$$\alpha \vee (\beta \wedge \gamma) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma) \quad (1.1)$$
$$(\alpha \wedge \beta) \vee \gamma \equiv (\alpha \vee \gamma) \wedge (\beta \vee \gamma) \quad (1.2)$$

As duas leis anteriores seguem das leis comutativa e distributiva.

Exemplo 1.30 A obtenção da FNC da expressão lógica $\neg((p \vee \neg q) \wedge \neg r)$ está mostrada na Tabela 1.37.

Tabela 1.37 Determinação da FNC de $\neg((p \vee \neg q) \wedge \neg r)$.

	$\equiv$	$\neg(p \vee \neg q) \vee \neg\neg r$	De Morgan
	$\equiv$	$\neg(p \vee \neg q) \vee r$	dupla negação
$\neg((p \vee \neg q) \wedge \neg r)$	$\equiv$	$(\neg p \wedge \neg\neg q) \vee r$	De Morgan
	$\equiv$	$(\neg p \wedge q) \vee r$	dupla negação
	$\equiv$	$(\neg p \vee r) \wedge (p \vee r)$	(1.2)

Exemplo 1.31 A obtenção da FNC da expressão lógica $(p \wedge q) \vee (r \wedge (s \vee t))$ está mostrada na Tabela 1.38.

Tabela 1.38 Determinação da FNC de $(p \wedge q) \vee (r \wedge (s \vee t))$.

$(p \wedge q) \vee (r \wedge (s \vee t))$	$\equiv$	$(p \vee (r \wedge (s \vee t))) \wedge (q \vee (r \wedge (s \vee t)))$	(1.2)
	$\equiv$	$(p \vee r) \wedge (p \vee s \vee t) \wedge (q \vee r) \wedge (q \vee s \vee t)$	(1.1)

Exemplo 1.32 A FNC da expressão lógica α: $(\neg p \vee q) \rightarrow r$ é obtida usando a regra da tabela (Tabela 1.39) e, na seqüência, obtida usando equivalências lógicas por meio do Procedimento 1.3.

Tabela 1.39 Tabela-verdade da fórmula $(\neg p \vee q) \rightarrow r$.

	p	q	r	$\neg p$	$(\neg p \vee q)$	$(\neg p \vee q) \rightarrow r$	
I_1	v	v	v	f	v	v	
I_2	v	v	f	f	v	f	←
I_3	v	f	v	f	f	v	
I_4	v	f	f	f	f	v	
I_5	f	v	v	v	v	v	
I_6	f	v	f	v	v	f	←
I_7	f	f	v	v	v	v	
I_8	f	f	f	v	v	f	←

De acordo com o Procedimento 1.1, FNC($(\neg p \vee q) \rightarrow r$): $(\neg p \vee \neg q \vee r) \wedge (p \vee \neg q \vee r) \wedge (p \vee q \vee r)$.

A Tabela 1.40 evidencia a equivalência das duas representações, ou seja, $((\neg p \vee q) \rightarrow r) \equiv (\neg p \vee \neg q \vee r) \wedge (p \vee \neg q \vee r) \wedge (p \vee q \vee r)$. Para facilitar a visualização, a seguinte nomeação foi adotada, α: $((\neg p \vee q) \rightarrow r)$, β: $(\neg p \vee \neg q \vee r)$, δ: $(p \vee \neg q \vee r)$ e λ: $(p \vee q \vee r)$, respectivamente.

Tabela 1.40 Tabela-verdade da equivalência $\alpha \equiv \beta \wedge \delta \wedge \lambda$.

	p	q	r	$\neg p$	$\neg q$	α	β	δ	λ	$\beta \wedge \delta \wedge \lambda$	$\alpha \leftrightarrow (\beta \wedge \delta \wedge \lambda)$
I_1	v	v	v	f	f	v		v	v	v	v
I_2	v	v	f	f	f	f	f	v	v	f	v

Tabela 1.40 *Continuação...*

	p	q	r	$\neg p$	$\neg q$	α	β	δ	λ	$\beta \wedge \delta \wedge \lambda$	$\alpha \leftrightarrow (\beta \wedge \delta \wedge \lambda)$
I_3	v	f	v	f	v	v	v	v	v	v	v
I_4	v	f	f	f	v	v	v	v	v	v	v
I_5	f	v	v	v	f	v	v	v	v	v	v
I_6	f	v	f	v	f	f	v	f	v	f	v
I_7	f	f	v	v	v	v	v	v	v	v	v
I_8	f	f	f	v	v	f	v	v	f	f	v

Por outro lado, usando o Procedimento 1.3, a FNC pode ser obtida por meio do uso de equivalências lógicas, como mostra a Tabela 1.41.

Tabela 1.41 Determinação da FNC de $((\neg p \vee q) \rightarrow r)$ usando equivalências lógicas.

	$\equiv \neg(\neg p \vee q) \vee r$	equivalência da implicação
$((\neg p \vee q) \rightarrow r)$	$\equiv (\neg(\neg p) \wedge \neg q) \vee r$	De Morgan
	$\equiv (p \wedge \neg q) \vee r$	dupla negação
	$\equiv (p \vee r) \wedge (\neg q \vee r)$	(1.2)

A FNC obtida pelo Procedimento 1.3, mostrado na Tabela 1.40, é equivalente à FNC obtida por meio de equivalências lógicas, mostrada na Tabela 1.41 – a segunda expressão, entretanto, é uma versão simplificada da primeira. A Tabela 1.42 exibe a equivalência entre as duas expressões, ou seja:

$$\beta \wedge \delta \wedge \lambda \equiv (p \vee r) \wedge (\neg q \vee r)$$

tal que β: $(\neg p \vee \neg q \vee r)$, δ: $(p \vee \neg q \vee r)$ e λ: $(p \vee q \vee r)$.

Tabela 1.42 Tabela-verdade da equivalência que evidencia $(\beta \wedge \delta \wedge \lambda) \equiv ((p \vee r) \wedge (\neg q \vee r))$.

	p	q	r	$\neg p$	$\neg q$	$\beta \wedge \delta \wedge \lambda$	$p \vee r$	$\neg q \vee r$	$(p \vee r) \wedge (\neg q \vee r)$	$(\beta \wedge \delta \wedge \lambda) \leftrightarrow ((p \vee r) \wedge (\neg q \vee r))$
I_1	v	v	v	f	f	v	v	v	v	v
I_2	v	v	f	f	f	f	v	f	f	v
I_3	v	f	v	f	v	v	v	v	v	v
I_4	v	f	f	f	v	v	v	v	v	v
I_5	f	v	v	v	f	v	v	v	v	v
I_6	f	v	f	v	f	f	f	f	f	v
I_7	f	f	v	v	v	v	v	v	v	v
I_8	f	f	f	v	v	f	f	v	v	v

Existe um procedimento similar ao Procedimento 1.3 para a obtenção da FNC, usando equivalências lógicas. O passo 3 do Procedimento 1.3, entretanto, precisa ser alterado para:

(3)quando a expressão obtida não tiver subexpressões compostas negadas, as duas leis a seguir são utilizadas para reduzir o escopo do $\vee$.

$$\alpha \wedge (\beta \vee \gamma) \equiv (\alpha \wedge \beta) \vee (\alpha \wedge \gamma) \tag{1.3}$$

$$(\alpha \vee \beta) \wedge \gamma \equiv (\alpha \wedge \gamma) \vee (\beta \wedge \gamma) \tag{1.4}$$

1.6.4 A notação clausal

A FNC é de particular interesse no entendimento e uso da linguagem de programação Prolog. Como visto na Seção 1.6.1, uma fórmula α representada na FNC é uma conjunção de cláusulas:

$$\text{FNC}(\alpha)\text{: } C_1 \wedge C_2 \wedge \ldots \wedge C_n$$

Uma cláusula, por sua vez, é uma disjunção de literais (ou seja, átomo ou átomo negado), isto é:

$$C_i = L_1 \vee L_2 \vee \ldots \vee L_s$$

Uma das vantagens de se ter a FNC de uma dada fórmula α é poder garantir que se o valor de α em uma determinada interpretação for v, então cada cláusula é também, separadamente, interpretada v, uma vez que a FNC é uma conjunção de cláusulas. Este fato torna a fórmula mais facilmente manipulável.

Como a FNC de uma fórmula α da Lógica Proposicional é sempre uma conjunção de cláusulas, a ordem em que estas cláusulas são escritas é irrelevante – pela propriedade associativa da conjunção ($\wedge$). Assim, pode-se dizer que a FNC é uma coleção de cláusulas. Escreve-se, então, a FNC de uma fórmula α como:

$$\{C_1, C_2, \ldots, C_n\}$$

sendo que a conjunção entre as cláusulas fica implícita. Nomeia-se de coleção apenas para indicar que a ordem não é relevante. Neste sentido, pode-se dizer que qualquer fórmula da Lógica Proposicional é uma coleção de cláusulas. Analogamente, uma cláusula C_i terá a forma:

$$C_i = L_1 \vee L_2 \vee \ldots \vee L_k$$

em que cada L_i ($1 \leq i \leq k$) é um literal. Aplicando o mesmo raciocínio anterior, pode-se dizer que uma cláusula é uma coleção de literais na qual a disjunção está implícita, ou seja,

$$\{L_1, L_2, \ldots, L_k\}$$

Exemplo 1.33 Seja

$$\alpha: ((\neg p \vee q) \wedge (\neg p \vee r)) \rightarrow s$$

e seja β a FNC(α),

$$\beta: (p \vee \neg q \vee s) \wedge (\neg p \vee \neg r \vee s) \wedge (\neg q \vee \neg r \vee s)$$

Pode-se escrever que:

$$\beta: C_1 \wedge C_2 \wedge C_3 \text{ tal que}$$

$$C_1: (p \vee \neg q \vee s)$$
$$C_2: (\neg p \vee \neg r \vee s) \text{ e}$$
$$C_3: (\neg q \vee \neg r \vee s)$$

A FNC pode ser representada pela coleção das três cláusulas, na qual a conjunção está implícita, isto é:

$$\beta: \{C_1, C_2, C_3\}$$

Uma possível convenção é escrever a fórmula como uma cláusula após a outra, lembrando que elas estão conectadas por $\wedge$:

$$C_1: p \vee \neg q \vee s$$

$$C_2: \neg p \vee \neg r \vee s$$

$$C_3: \neg q \vee \neg r \vee s$$

Como cada cláusula é uma coleção de literais que estão conectadas por $\vee$, para cada cláusula pode-se escrever primeiro os literais positivos e, logo após, os negativos, obtendo-se:

$$C_1: s \vee p \vee \neg q$$

$$C_2: s \vee \neg p \vee \neg r$$

$$C_3: s \vee \neg q \vee \neg r$$

Essa separação entre literais positivos e negativos prepara a cláusula para a introdução da notação definida por Kowalski (1974). Na notação, as cláusulas são representadas por:

C_1:	$s, p \leftarrow q$		$s \vee p \leftarrow q$
C_2:	$s \leftarrow r, p$	significando	$s \leftarrow r \wedge p$
C_3:	$s \leftarrow q, r$		$s \leftarrow q \wedge r$

ou seja, existe uma disjunção implícita nos literais à esquerda do ←, chamados de conclusões, e uma conjunção implícita nos literais à direta do ←, chamados de condições (ou premissas). Deve ser observado que a notação anterior é equivalente a:

$$C_1: q \rightarrow s \vee p$$

$$C_2: r \wedge p \rightarrow s$$

$$C_3: q \wedge r \rightarrow s$$

Observação 1.18 Uma cláusula genérica na notação de Kowalski (1974):

$$A_1, A_2, ..., A_m \leftarrow B_1, B_2, ..., B_n$$

é equivalente a:

$$A_1 \vee A_2 \vee ... \vee A_m \leftarrow B_1 \wedge B_2 \wedge ... \wedge B_n$$

que é equivalente a:

$$A_1 \vee A_2 \vee ... \vee A_m \vee \neg(B_1 \wedge B_2 \wedge ... \wedge B_n$$

que é equivalente a:

$$A_1 \vee A_2 \vee ... \vee A_m \vee \neg B_1 \vee \neg B_2 \vee ... \vee \neg B_n$$

Dependendo do número de literais envolvidos na cláusula, os casos especiais mostrados na Tabela 1.43 podem ocorrer.

Tabela 1.43 Tipos de cláusulas.

$m > 1$	as conclusões são indefinidas, isto é, há várias conclusões.
$m \leq 1$	são as chamadas *cláusulas de Horn*, que têm como casos particulares (a), (b), (c) e (d).
(a) $m = 1, n > 0$	$A_1 \leftarrow B_1, B_2, ..., B_n$ chamada de *cláusula definida*, isto é, existe somente uma conclusão.
(b) $m = 1, n = 0$	$A_1 \leftarrow$ chamada de *cláusula definida incondicional* ou *fato*. Neste caso, o símbolo ← é abandonado.

Tabela 1.43 *Continuação...*

(c) $m = 0, n > 0$	$m = 0, n > 0$ $\leftarrow B_1, B_2, ..., B_n$ conhecida como *negação pura* de $B_1, B_2, ..., B_n$.
(d) $m = 0, n = 0$	$\leftarrow$ chamada de *cláusula vazia* e denotada por nil.

As únicas cláusulas que podem ser representadas usando a linguagem de programação Prolog são as cláusulas de Horn (o símbolo $\leftarrow$ é representado por :- na sintaxe do Prolog de Edinburgh). Em uma situação em que um dado conhecimento pode ser representado utilizando Lógica Proposicional, apenas as expressões que forem cláusulas de Horn serão passíveis de serem representadas em Prolog.

1.7 INFERÊNCIA LÓGICA E SISTEMAS DE DERIVAÇÃO

Padrões de raciocínio podem ser expressos de várias maneiras. Em português, por exemplo, a conclusão é tipicamente colocada após as premissas e é "anunciada" por palavras indicativas, tais como: "então", "logo", "portanto", "como conseqüência", "conclui-se", etc. Um argumento é correto se a conclusão segue logicamente das premissas, como formalmente estabelecido na Definição 1.15.

Definição 1.15 Um *argumento* é uma seqüência $\alpha_1, \alpha_2, \alpha_3, ... \alpha_n$ $(n \geq 1)$ de proposições, na qual as proposições α_i $(1 \leq i \leq n-1)$ são chamadas de *premissas* e a proposição α_n é chamada de *conclusão*. Indica-se um argumento por

$$\alpha_1, \alpha_2, \alpha_3, ..., \alpha_{n-1} \vdash \alpha_n$$

Um argumento $\alpha_1, \alpha_2, \alpha_3, ..., \alpha_{n-1} \vdash \alpha_n$ é um *argumento válido* se e somente se a fórmula

$$\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge ... \wedge \alpha_{n-1} \rightarrow \alpha_n \text{ for uma tautologia,}$$

ou seja, $\alpha_1, \alpha_2, \alpha_3, ..., \alpha_{n-1} \models \alpha_n$. Essa afirmação é justificada pelo Teorema 1.1, da Seção 1.4.

Um argumento válido pode ser lido como:

$\alpha_1, \alpha_2, \alpha_3, ..., \alpha_{n-1}$ acarretam α_n ou

α_n decorre de $\alpha_1, \alpha_2, \alpha_3, ..., \alpha_{n-1}$ ou

α_n é conseqüência lógica de $\alpha_1, \alpha_2, \alpha_3, ..., \alpha_{n-1}$.

Para $n = 1$, considera-se por extensão o argumento válido se e somente se α_1 for tautológica.

Eventualmente, a verificação da validade de um argumento por meio de tabelas-verdade pode ser um trabalho longo, dado que depende do número de átomos nele existentes. A verificação da validade de um argumento que envolve sete átomos (ou seja, proposições atômicas), por exemplo, envolve a construção de uma tabela-verdade com $2^7 = 128$ linhas. Outra maneira de evidenciar a validade de argumentos é por meio de um procedimento descrito por uma seqüência de passos, que faz uso de argumentos válidos já conhecidos e de equivalências, processo que leva à noção de derivação ou prova formal.

Observação 1.19 Existem diferentes sistemas para a realização de derivações. Todos os sistemas têm as seguintes características em comum:

1. consideram uma lista de argumentos lógicos admissíveis, chamada de *regras de inferência.* Essa lista é referenciada como L;
2. a derivação é uma lista de expressões lógicas. Originalmente, essa lista é vazia. Expressões podem ser adicionadas à lista se forem premissas ou se puderem ser obtidas a partir das expressões anteriores, por meio da aplicação de regras de inferência. Esse processo continua até que a conclusão seja obtida.

Definição 1.16 Considere as fórmulas $\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n$ e β da Lógica Proposicional. Diz-se que uma seqüência finita de fórmulas $C_1, C_2, \ldots, C_k$ é uma *prova* (ou *dedução* ou *derivação*) de β a partir de $\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n$ (consideradas *premissas*) se e somente se:

1. cada C_i for uma premissa α_j $(1 \leq j \leq n)$; ou
2. C_i provém das fórmulas precedentes, pelo uso de um argumento válido de L; ou
3. C_i provém do uso do princípio de substituição usado em uma fórmula anterior; ou
4. C_k é β.

Diz-se, então, que β é dedutível a partir de $\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n$ ou que β é um *teorema.* Se a seqüência puder ser construída, isto é, se existir uma derivação para a conclusão β, dado que $\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n$ são as premissas e dado que L é um conjunto de regras de inferência admissíveis, diz-se que o argumento é válido, ou seja, $\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n \vdash \beta$ é válido.

Observação 1.20 Na maioria dos sistemas para derivações formais o conjunto de regras de inferência admissíveis é fixo – nenhuma outra regra de inferência pode ser usada a menos que esteja incluída em L. Para os exemplos e discussões a seguir, o conjunto L de regras de inferência considerado é apresentado na Tabela 1.44.

Tabela 1.44 Regras de inferência.

Regra	Nome da regra
$\alpha, \alpha \to \beta \models \beta$	*modus ponens*
$\alpha \to \beta, \neg\beta \models \neg\alpha$	*modus tollens*
$\alpha \to \beta, \beta \to \gamma \models \alpha \to \gamma$	silogismo hipotético (regra da cadeia)
$\alpha \vee \beta, \neg\alpha \models \beta$ $\alpha \vee \beta, \neg\beta \models \alpha$	silogismo disjuntivo silogismo disjuntivo (variante)
$\alpha \wedge \beta \models \alpha$ $\alpha \wedge \beta \models \beta$	simplificação simplificação (variante)
$\alpha, \beta \models \alpha \wedge \beta$	conjunção (ou combinação)
$\alpha \to \beta, \neg\alpha \to \beta \models \beta$	de casos
$\alpha \models \alpha \vee \beta$ $\beta \models \alpha \vee \beta$	adição adição (variante)
$\alpha \to \beta, \gamma \to \delta, \alpha \vee \gamma \models \beta \vee \delta$	dilema construtivo
$\alpha \to \beta, \gamma \to \delta, \neg\beta \vee \neg\delta \models \neg\alpha \vee \neg\gamma$	dilema destrutivo
$\alpha \to \beta \models \neg\beta \to \neg\alpha$	contraposição
$\alpha, \neg\alpha \models \beta$	da inconsistência
$\alpha \to \beta, \beta \to \alpha \models \alpha \leftrightarrow \beta$	introdução da equivalência
$\alpha \leftrightarrow \beta \models \alpha \to \beta$ $\alpha \leftrightarrow \beta \models \beta \to \alpha$	eliminação da equivalência variante da eliminação da equivalência

A regra da inconsistência segue do fato de que $\alpha \wedge \neg\alpha$ é sempre avaliada f e, portanto, $\alpha, \neg\alpha \models \beta$ é trivialmente verdade. A lei da inconsistência tem um grande impacto em sistemas lógicos, pois se uma simples contradição pode ser derivada a partir das premissas (no caso, a derivação de α e também a derivação de $\neg\alpha$), então toda expressão lógica possível e imaginável β também pode ser derivada. É fundamental, pois, que as premissas não permitam a derivação de qualquer contradição, caso contrário, em um tal sistema, tudo pode ser provado.

Regras de inferência devem ser escolhidas de tal maneira que possam derivar apenas resultados que estejam corretos. Isso significa que L não deve conter qualquer falácia. Uma falácia permite encontrar uma conclusão que não possa ser derivada das premissas e, conseqüentemente, não correta.

Observação 1.21 Um sistema para derivações não deve ser apenas correto, mas também completo. Por completo entende-se um sistema que viabiliza a derivação de toda conclusão que logicamente segue das premissas. O sistema de regras descrito na Tabela 1.44 não é completo. Considere a lei do meio excluído $\alpha \vee \neg\alpha$. Essa lei é válida sem qualquer premissa, ou seja, $\models \alpha \vee \neg\alpha$. É fácil ver que, sem qualquer premissa, nenhuma das leis da Tabela 1.44 pode ser usada. Conseqüentemente, $\alpha \vee \neg\alpha$ não pode ser derivada se L for o conjunto de regras da Tabela 1.44. A seguir, são mostrados alguns exemplos de verificação da validade de argumentos, usando argumentos válidos já conhecidos e equivalências.

Observação 1.22 Em Lógica, o que é chamado de teoria é dado por um conjunto de premissas e por um conjunto de todas as conclusões que podem ser derivadas a partir das premissas. Freqüentemente, as premissas de uma teoria são chamadas de *axiomas* e as conclusões que podem ser derivadas dos axiomas são chamadas de *teoremas*.

Observação 1.23 Equivalências representam uma regra especial nas derivações. Toda equivalência pode ser tratada como uma regra de inferência que permite substituir cada uma das wffs por sua equivalente ou que permite substituir uma subfórmula de uma wff por sua equivalente.

Exemplo 1.34 Considere:

Se as uvas caem, então a raposa as come.
Se a raposa as come, então estão maduras.
As uvas estão verdes ou caem.
Logo,
A raposa come as uvas se e somente se as uvas caem.

Identificando as proposições atômicas nas sentenças em língua natural escritas neste Exemplo 1.34 e nomeando-as com os símbolos convencionados para átomos na Lógica Proposicional, tem-se:

p: as uvas caem
q: a raposa come as uvas
r: as uvas estão maduras

Reescrevendo o enunciado anterior usando a linguagem da Lógica Proposicional, tem-se:

$$p \rightarrow q$$
$$q \rightarrow r$$
$$\neg r \vee p$$
logo
$$p \leftrightarrow q$$

A Tabela 1.45 exibe a prova da conclusão como estabelecida na Definição 1.16.

Tabela 1.45 Construção da prova de $p \leftrightarrow q$.

Tem-se	C_1	$p \rightarrow q$	premissa
	C_2	$q \rightarrow r$	premissa
	C_3	$\neg r \vee p$	premissa
Deduz-se	C_4	$r \rightarrow p$	(C_3: equivalência)
	C_5	$q \rightarrow p$	(C_2 + C_4 + silogismo hipotético)
	C_6	$(p \rightarrow q) \wedge (q \rightarrow p)$	(C_1 + C_5 + conjunção)
	C_7	$(p \leftrightarrow q)$	(C_6: equivalência)

A seqüência C_1, C_2, C_3, C_4, C_5, C_6, C_7 é uma prova da conclusão $p \leftrightarrow q$ e o argumento $(p \rightarrow q)$, $(q \rightarrow r)$, $(\neg r \vee p) \vdash (p \leftrightarrow q)$ é válido.

Exemplo 1.35 Considere:

Gabriel estuda ou não está cansado.
Se Gabriel estuda, então dorme tarde.
Gabriel não dorme tarde ou está cansado.
Logo,
Gabriel está cansado se e somente se estuda.

Identificando as proposições atômicas nas sentenças em língua natural escritas neste Exemplo 1.35 e nomeando-as com os símbolos convencionados para átomos na Lógica Proposicional, tem-se:

p: Gabriel estuda
q: Gabriel está cansado
r: Gabriel dorme tarde

Note que é conveniente identificar, como átomos, sentenças que não envolvam a negação. No caso particular deste exemplo, em vez de identificar como átomo a sentença *Gabriel não dorme tarde*, identifica-se como átomo (r) a sentença *Gabriel dorme tarde*, e a sentença *Gabriel não dorme tarde* é reescrita em Lógica Proposicional como $\neg r$. Reescrevendo o enunciado anterior usando a linguagem da Lógica Proposicional, tem-se:

$$p \vee \neg q$$
$$p \rightarrow r$$
$$\neg r \vee q$$
logo
$$p \leftrightarrow q$$

A Tabela 1.46 exibe a prova da conclusão como estabelecida na Definição 1.16.

Tabela 1.46 Construção da prova de $p \leftrightarrow q$.

Tem-se	C_1	$p \vee \neg q$	premissa
	C_2	$p \rightarrow r$	premissa
	C_3	$\neg r \vee q$	premissa
Deduz-se	C_4	$q \rightarrow p$	(C_1: equivalência)
	C_5	$r \rightarrow q$	(C_3: equivalência)
	C_6	$p \rightarrow q$	($C_2 + C_5$ + silogismo hipotético)
	C_7	$(p \rightarrow q) \wedge (q \rightarrow p)$	($C_6 + C_4$ + conjunção)
	C_8	$(p \leftrightarrow q)$	(C_7: equivalência)

A seqüência C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8 é uma prova da conclusão $p \leftrightarrow q$ e o argumento $(p \vee \neg q)$, $(p \rightarrow r)$, $(\neg r \vee q) \vdash (p \leftrightarrow q)$ é válido.

Exemplo 1.36 Verificar a validade do argumento descrito em língua natural como:

Se a Terra é redonda então a Lua é oval.
Se a Lua é oval, então Saturno não tem anéis.
Se a Terra não é redonda então Saturno não tem anéis.
Logo
Saturno não tem anéis.

Identificando as proposições atômicas nas sentenças em língua natural escritas neste Exemplo 1.36 e nomeando-as com os símbolos convencionados para átomos na Lógica Proposicional, tem-se:

p: a Terra é redonda
q: a Lua é oval
r: Saturno tem anéis

Reescrevendo o enunciado anterior usando a Lógica Proposicional, tem-se:

$p \rightarrow q$
$q \rightarrow \neg r$
$\neg p \rightarrow \neg r$
logo
$\neg r$

A Tabela 1.47 exibe a prova da conclusão como estabelecida na Definição 1.16.

Tabela 1.47 Construção da prova de $\neg r$.

Tem-se	C_1	$p \to q$	premissa
	C_2	$q \to \neg r$	premissa
	C_3	$\neg p \to \neg r$	premissa
Deduz-se	C_4	$p \to \neg r$	(C_1 + C_2 + silogismo hipotético)
	C_5	$\neg r$	(C_3 + C_4 + de casos)

A seqüência C_1, C_2, C_3, C_4, C_5 é uma prova da conclusão $\neg r$ e o argumento $(p \to q)$, $(q \to \neg r)$, $(\neg p \to \neg r) \mid\!- \neg r$ é válido.

Exemplo 1.37 Identificar a prova da conclusão do argumento:

$$\neg p \to q,\ q \to r,\ \neg r \vee s,\ \neg s \mid\!- p$$

A Tabela 1.48 exibe a prova da conclusão como estabelecida na Definição 1.16.

Tabela 1.48 Construção da prova de p.

Tem-se	C_1	$\neg p \to q$	premissa
	C_2	$q \to r$	premissa
	C_3	$\neg r \vee s$	premissa
	C_4	$\neg s$	premissa
Deduz-se	C_5	$\neg r$	(C_3 + C_4 + silogismo disjuntivo)
	C_6	$\neg q$	(C_2 + C_5 + *modus tollens*)
	C_7	$\neg(\neg p)$	(C_2 + C_7 + *modus tollens*)
	C_8	p	(C_7: equivalência dupla negação)

A seqüência C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8 é uma prova da conclusão p e o argumento $\neg p \to q$, $q \to r$, $\neg r \vee s$, $\neg s \mid\!- p$ é válido.

Exemplo 1.38 Identificar a prova da conclusão do argumento:

$$p \to q,\ \neg q,\ \neg p \to (r \vee s),\ r \to t,\ u \to \neg t,\ u \mid\!- s$$

A Tabela 1.49 exibe a prova da conclusão como estabelecida na Definição 1.16.

Tabela 1.49 Construção da prova de s.

Tem-se	C_1	$p \to q$	premissa
	C_2	$\neg q$	premissa
	C_3	$\neg p \to (r \vee s)$	premissa

Tabela 1.49 *Continuação...*

	C_4	$r \to t$	premissa
	C_5	$u \to \neg t$	premissa
	C_6	u	premissa
Deduz-se	C_7	$\neg p$	(C_1 + C_2 + *modus tollens*)
	C_8	$r \vee s$	(C_3 + C_7 + *modus ponens*)
	C_9	$\neg t$	(C_5 + C_6 + *modus ponens*)
	C_{10}	$\neg r$	(C_4 + C_9 + *modus tollens*)
	C_{11}	s	(C_8 + C_{10} + silogismo disjuntivo)

A seqüência C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9, C_{10}, C_{11} é uma prova da conclusão s e o argumento $p \to q, \neg q, \neg p \to (r \vee s), r \to t, u \to \neg t, u \vdash s$ é válido.

Exemplo 1.39 Identificar a prova da conclusão do argumento:

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

A Tabela 1.50 exibe a prova da conclusão como estabelecida na Definição 1.16.

Tabela 1.50 Construção da prova de $\neg p \wedge \neg r$.

Tem-se	C_1	$p \to q$	premissa
	C_2	$r \to s$	premissa
	C_3	$(q \vee s) \to t$	premissa
	C_4	$\neg t$	premissa
Deduz-se	C_5	$\neg(q \vee s)$	(C_3 + C_4 + *modus tollens*)
	C_6	$\neg q \wedge \neg s$	(C_5 + equivalência De Morgan)
	C_7	$\neg q$	(C_6 + simplificação)
	C_8	$\neg s$	(C_6 + simplificação)
	C_9	$\neg r$	(C_2 + C_8 + *modus tollens*)
	C_{10}	$\neg p$	(C_1 + C_7 + *modus tollens*)
	C_{11}	$\neg p \wedge \neg r$	(C_{10} + C_9 + conjunção)

A seqüência C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9, C_{10}, C_{11} é uma prova da conclusão $\neg p \wedge \neg r$ e o argumento $p \to q, r \to s, (q \vee s) \to t, \neg t \vdash \neg p \wedge \neg r$ é válido.

Observação 1.24 Em Matemática, para provar que $\alpha \to \beta$, o seguinte argumento informal é usado:

1. assuma α, que é adicionado ao conjunto de premissas;

2. prove β, usando α se necessário;
3. descarte α, significando que não é mais necessariamente verdade e conclua $\alpha \rightarrow \beta$.

Exemplo 1.40 [Grassmann & Tremblay, 1996]. Um casal tem um menino e está esperando uma segunda criança. Prove que se a segunda criança for uma menina então o casal terá um menino e uma menina.

Seja:

p: a primeira criança do casal é um menino

q: a segunda criança do casal é uma menina

O que se quer provar é $q \rightarrow (p \wedge q)$, dado que a premissa é p. De acordo com o Teorema da Dedução (discutido a seguir), a prova pode ser conduzida como:

1. p é verdade: a primeira criança do casal é um menino;
2. assuma q: ou seja, assuma que a segunda criança do casal é uma menina;
3. com base em p e em q, conclua $p \wedge q$, pela regra de inferência da conjunção;
4. nesse ponto, o Teorema da Dedução permite concluir que $q \rightarrow (p \wedge q)$. A suposição de q pode ser descartada, ou seja, $q \rightarrow (p \wedge q)$ é verdade mesmo que q seja falso: neste caso, $q \rightarrow (p \wedge q)$ é trivialmente verdade.

É clara a razão pela qual esse padrão de prova é correto. Na prova de $\alpha \rightarrow \beta$ é preciso considerar apenas o caso em que α é avaliada v; se α for f, $\alpha \rightarrow \beta$ é trivialmente v. Se α for v, então pode ser adicionada às premissas.

Essencialmente, a regra estabelece que uma suposição pode ser convertida no antecedente de uma condicional; em algumas referências o processo descrito é tratado como uma regra de inferência chamada de *regra de introdução da condicional*. Ela difere das outras porque emprega um raciocínio hipotético, isto é, uma estratégia de prova fundamentada em hipótese, que pode ser considerada uma suposição feita no contexto do argumento, com o objetivo de mostrar que determinada conclusão segue da suposição. A maneira mais geral para provar uma condicional é colocar seu antecedente como hipótese (ou seja, admiti-lo como possibilidade no contexto do argumento) e então mostrar que seu conseqüente deve se seguir.

Exemplo 1.41 Este exemplo e a discussão sobre ele foram extraídos de Nolt & Rohatyn (1991) e adaptados. Suponha que um corredor machucou seu tornozelo uma semana antes de uma grande corrida, e a intenção seja persuadi-lo a parar de correr por alguns dias, a fim de que seu tornozelo sare. Alguém pode alertá-lo fazendo a seguinte afirmação condicional: "se você continuar a correr, não estará apto a disputar a corrida". A resposta do corredor eventualmente pode ser: "prove isso".

A maneira mais geral de provar uma condicional é considerar seu antecedente como hipótese (isto é, admiti-lo no contexto do argumento) e então mostrar que o conseqüente do condicional deve se seguir. Para fazer isso, o argumento pode ser elaborado da seguinte maneira:

> Seu tornozelo está muito inchado. Suponhamos que você continue a correr. Se ele está muito inchado e você continuar a correr, seu tornozelo não vai sarar em uma semana. Se ele não sarar em uma semana, então você não estará apto a disputar a corrida. Desse modo, não estará apto a disputar a corrida.

Este argumento é um argumento hipotético. A palavra "suponhamos" é um indicador de hipótese, assinalando que o enunciado "você continue a correr" é uma hipótese. Dessa hipótese, a conclusão "você não estará apto a disputar a corrida" é descrita a seguir. O argumento está construído com base em três suposições:

1. seu tornozelo está muito inchado;
2. se seu tornozelo está muito inchado e você continuar a correr, então seu tornozelo não irá sarar em uma semana;
3. se seu tornozelo não sarar em uma semana, então você não estará apto a disputar a corrida.

As três suposições anteriores são consideradas verdadeiras – diferentemente da hipótese, que foi assumida em consideração ao argumento. Uma vez admitida as suposições, o argumento hipotético mostra que se a hipótese "você continuar a correr" for verdadeira, então a conclusão do argumento hipotético, "você não estará apto a disputar a corrida", também deve ser verdadeira. Assim, mostra-se que a expressão condicional:

Se você continuar a correr, então você não estará apto a disputar a corrida.

é verdadeira, o que é exatamente o que se queria provar. A expressão condicional foi deduzida colocando como hipótese seu antecedente e mostrando que seu conseqüente segue-se do conjunto formado pela hipótese com as premissas.

É fato que a correção do argumento depende da veracidade das suposições – na vida real, a veracidade delas pode ser duvidosa. A correção do argumento, entretanto, não depende da veracidade da hipótese. Considerando as suposições e independentemente de o corredor continuar ou não a correr, deve ainda ser verdade que, se ele continuar correndo, ele não estará apto a disputar a corrida. A hipótese é adicionada somente para mostrar que, dadas as suposições, ela implica a conclusão "você não estará apto a disputar a corrida". Uma vez provado isso, a hipótese é descartada e a expressão condicional que representa a conclusão é estabelecida somente com base nas suposições. A formalização do argumento está na Tabela 1.51, com a seguinte simbologia:

p: seu tornozelo está muito inchado.

q: você continua a correr.

r: seu tornozelo não irá sarar em uma semana.

s: você está apto a disputar a corrida.

O argumento pode ser reescrito em Lógica Proposicional como:

$$p, (p \wedge q) \rightarrow \neg r, \neg r \rightarrow \neg s \vdash q \rightarrow \neg s$$

Tabela 1.51 Exemplo de uso da regra de introdução da condicional.

C_1	p	premissa
C_2	$(p \wedge q) \rightarrow \neg r$	premissa
C_3	$\neg r \rightarrow \neg s$	premissa
C_4	q	hipótese
C_5	$p \wedge q$	(C_1 + C_4 + conjunção)
C_6	$\neg r$	(C_2 + C_5 + *modus ponens*)
C_7	$\neg s$	(C_3 + C_6 + *modus ponens*)
C_8	$q \rightarrow \neg s$	(C_4 – C_7 + introdução da condicional)

Note na Tabela 1.51 que, a partir da introdução da hipótese q (inclusive ela), as fórmulas são endentadas à direita na coluna, para indicar a duração do argumento hipotético, ou seja, a parte da prova na qual a hipótese é considerada. Como argumentos hipotéticos são construídos com a introdução de uma hipótese com um escopo de atuação na prova, no que segue a seqüência de fórmulas que identifica a prova da conclusão não é apresentada. A própria tabela evidencia a prova, com o escopo da validade da hipótese assinalada via deslocamento à direita.

Observação 1.25 A regra da introdução da condicional pode ser enunciada como: dada a derivação de uma wff β a partir de uma hipótese α, pode-se descartar a hipótese e inferir a wff $\alpha \rightarrow \beta$. A regra é estabelecida pelo Teorema da Dedução.

Teorema 1.3 (Teorema da Dedução) Sejam α e β duas wffs e $\delta_1, \delta_2, \delta_3,...$ premissas. Se juntos α, $\delta_1, \delta_2, \delta_3,...$ logicamente implicam β, então $\delta_1, \delta_2, \delta_3,...$ logicamente implicam $\alpha \rightarrow \beta$.

Observação 1.26 As regras de inferência da Tabela 1.44 com o Teorema da Dedução formam um sistema completo.

Exemplo 1.42 Usando dedução natural, o argumento a seguir está provado na Tabela 1.52.

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

Tabela 1.52 Prova usando a regra de introdução da condicional.

C_1	$p \rightarrow q$	premissa
C_2	$q \rightarrow r$	premissa
C_3	p	hipótese
C_5	q	(C_1 + C_3 + *modus ponens*)
C_6	r	(C_2 + C_5 + *modus ponens*)
C_8	$p \rightarrow r$	(C_3 – C_6 + introdução da condicional)

Exemplo 1.43 Considere a prova do argumento $(p \wedge q) \rightarrow r \mid\!- p \rightarrow (q \rightarrow r)$.

Este exemplo ilustra a situação na qual a conclusão a ser derivada é uma expressão condicional que tem, como conseqüente, outra expressão condicional. A derivação emprega a mesma estratégia anterior. O antecedente da condicional mais externa é assumido como hipótese. Como, no caso, o conseqüente é outra expressão condicional, pode-se assumir como uma nova hipótese o antecedente dessa última condicional e derivar sua conclusão, como mostra a Tabela 1.53.

Tabela 1.53 Prova usando a regra de introdução da condicional.

C_1	$(p \wedge q) \rightarrow r$	premissa
C_2	p	hipótese
C_3	q	hipótese
C_4	$p \wedge q$	(C_2 + C_3 + conjunção)
C_5	r	(C_1 + C_4 + *modus ponens*)
C_6	$q \rightarrow r$	(C_3 – C_5 + introdução da condicional)
C_7	$p \rightarrow (q \rightarrow r)$	(C_2 – C_6 + introdução da condicional)

Note que C_2 é o antecedente da conclusão, inserido na prova como hipótese. Para derivar a conclusão (que é a condicional $q \rightarrow r$), a partir desse ponto, coloca-se como hipótese o antecedente da condicional, isto é, q. Como é uma nova hipótese, ela requer um deslocamento à direita para indicar que é a segunda hipótese vigente. Como a conclusão r é derivada (C_5), a hipótese q pode ser descartada e a condicional $q \rightarrow r$ pode ser inferida pela regra de introdução da condicional. Foi mostrado então que $q \rightarrow r$ segue da hipótese original p. A hipótese p permanece vigente até ser descartada e a conclusão desejada $(q \rightarrow r)$ ser inferida.

Observação 1.27 Algumas wffs são passíveis de serem provadas sem suposições (premissas) – são os teoremas. A prova de um teorema se inicia com uma ou mais hipóteses que serão descartadas pela introdução da condicional, como mostra o Exemplo 1.44.

Exemplo 1.44 A Tabela 1.54 mostra a prova de $\vdash p \rightarrow (p \vee q)$ usando a regra de introdução da condicional.

Tabela 1.54 Prova de $\vdash p \rightarrow (p \vee q)$.

C_1	p	hipótese
C_2	$p \vee q$	C_1 + adição
C_3	$p \rightarrow (p \vee q)$	($C_1 - C_2$ + introdução da condicional)

Exemplo 1.45 A Tabela 1.55 mostra a prova de $p \vee q \vdash q \vee p$ usando a regra de introdução da condicional.

Tabela 1.55 Prova de $p \vee q \vdash q \vee p$.

C_1	$p \vee q$	premissa
C_2	p	hipótese
C_3	$q \vee p$	(C_2 + adição)
C_4	$p \rightarrow (q \vee p)$	($C_2 - C_3$ + introdução da condicional)
C_5	q	hipótese
C_6	$q \vee p$	(C_5 + adição)
C_7	$q \rightarrow (q \vee p)$	($C_5 - C_6$ + introdução da condicional)
C_8	$(q \vee p) \vee (q \vee p)$	($C_1 + C_4 + C_7$ + dilema construtivo)
C_9	$(q \vee p)$	(C_8 + equivalência - idempotência)

Exemplo 1.46 A Tabela 1.56 mostra a prova de $(p \wedge q) \vee (p \wedge r) \vdash p \wedge (q \vee r)$ usando a regra de introdução da condicional.

Tabela 1.56 Prova de $(p \wedge q) \vee (p \wedge r) \vdash p \wedge (q \vee r)$.

C_1	$(p \wedge q) \vee (p \wedge r)$	premissa
C_2	$(p \wedge q)$	hipótese
C_3	p	(C_2 + simplificação)
C_4	q	(C_2 + simplificação)
C_5	$q \vee r$	(C_4 + adição)
C_6	$p \wedge (q \vee r)$	($C_3 + C_5$ + adição)
C_7	$(p \wedge q) \rightarrow p \wedge (q \vee r)$	($C_2 - C_6$ + introdução da condicional)
C_8	$(p \wedge r)$	hipótese
C_9	p	(C_8 + simplificação)
C_{10}	r	(C_8 + simplificação)

Tabela 1.56 *Continuação...*

C_{11}	$q \lor r$	(C_{10} + adição)
C_{12}	$p \land (q \lor r)$	(C_{11} + adição)
C_{13}	$(p \land r) \to p \land (q \lor r)$	($C_8 - C_{12}$ + introdução da condicional)
C_{14}	$p \land (q \lor r)$	($C_1 + C_7 + C_{13}$ + dilema construtivo)

Observação 1.28 Outra estratégia de prova que usa o raciocínio hipotético é conhecida como *redução ao absurdo* ou prova indireta. Para provar uma conclusão, a conclusão negada é inserida como hipótese e a estratégia busca derivar uma contradição – a derivação da contradição evidencia que a hipótese negada é falsa, o que permite inferir que a conclusão segue das premissas. Como visto anteriormente, uma contradição é qualquer wff no padrão $\alpha \land (\neg\alpha)$. A wff pode ser um átomo ou, então, uma expressão composta. A regra da redução ao absurdo é estabelecida como: dada a derivação de uma contradição a partir de uma hipótese α, pode-se descartar a hipótese e inferir $\neg\alpha$.

Exemplo 1.47 A Tabela 1.57 mostra a prova do argumento $p \to q, \neg q \vdash \neg p$ usando a redução ao absurdo.

Tabela 1.57 Prova de $p \to q, \neg q \vdash \neg p$ por redução ao absurdo.

C_1	$p \to q$	premissa
C_2	$\neg q$	premissa
C_3	p	hipótese
C_4	q	($C_1 + C_3$ + *modus ponens*)
C_5	$q \land \neg q$	($C_4 + C_2$ + adição)
C_6	$\neg p$	($C_3 - C_5$ + redução ao absurdo)

Exemplo 1.48 A Tabela 1.58 mostra a prova do argumento $p \to q, \neg q \vdash \neg p$ usando a redução ao absurdo.

Tabela 1.58 Prova de $p \to q, \neg q \vdash \neg p$ usando redução ao absurdo.

C_1	$p \to q$	premissa
C_2	$\neg q$	premissa
C_3	p	hipótese
C_4	q	($C_1 + C_3$ + *modus ponens*)
C_5	$q \land \neg q$	($C_4 + C_2$ + adição)
C_6	$\neg p$	($C_3 - C_5$ + redução ao absurdo)

Observação 1.29 Quando um esquema de raciocínio hipotético é usado, alguns cuidados devem ser tomados:

1. cada hipótese introduz, em uma prova, um deslocamento nas fórmulas inferidas a partir de então, até o ponto de a hipótese ser descartada pela aplicação da introdução da condicional ou da redução ao absurdo;
2. nenhuma ocorrência de uma fórmula deslocada pode ser usada em qualquer regra aplicada após o término do deslocamento. Isso garante que a fórmula derivada da hipótese não seja usada depois que a hipótese for descartada;
3. se duas ou mais hipóteses estiverem ativas simultaneamente, então a ordem na qual elas são descartadas deve ser a ordem inversa na qual elas foram introduzidas;
4. uma prova não está completa até que todas as hipóteses sejam descartadas.

1.8 PROVA AUTOMÁTICA DE TEOREMAS - ALGORITMO DE WANG

Lógica e Prova Automática de Teoremas são de significativa importância na área de Inteligência Artificial – a primeira por ser uma linguagem formal utilizada na expressão de conhecimento e representação de problemas e, a segunda, por ser uma possível forma de obter soluções de muitos desses problemas de maneira automática.

A Lógica Proposicional, embora aplicável a um número restrito de problemas, fornece uma ferramenta simples, com a qual os conceitos básicos de prova automática de teoremas podem ser ilustrados. Nesta seção, é apresentado o método sintático para a prova automática de teoremas da Lógica Proposicional, conhecido como Algoritmo de Wang, proposto e descrito em Wang (1960, 1964).

A referência Monard & Nicoletti (1990) apresenta e discute em detalhes duas conhecidas implementações deste algoritmo (ver Coelho & Cotta (1988)), escritas na linguagem de programação Prolog, abordando tanto a estrutura de dados utilizada quanto a eficiência de execução de cada uma delas, procurando evidenciar seus aspectos positivos e negativos. Com base na discussão, é proposta e descrita uma terceira implementação em Prolog do algoritmo de Wang, que introduz diversas otimizações em relação às duas outras versões descritas na literatura. As três implementações são comparadas com relação à eficiência medida em tempo de execução, usando um conjunto de 50 teoremas e 21 não-teoremas. As medidas obtidas evidenciaram a superioridade da implementação proposta na referência que, em média, executa aproximadamente duas vezes mais rápido que a mais eficiente das outras duas implementações.

O algoritmo de Wang espera como entrada uma wff, e, usando um conjunto de oito regras sintáticas, conclui-se que a wff fornecida é ou não um teorema. Dependendo dos conectivos presentes na wff o algoritmo a quebra em subexpressões que, recursivamente, devem ser provadas teoremas também.

Seis das oito regras usadas pelo algoritmo são manipulações simbólicas fundamentadas no conectivo principal de subfórmulas. As duas regras restantes regulam os critérios de parada do algoritmo: uma detecta quando uma expressão é um teorema, e a outra quando a expressão não é um teorema. A entrada para o algoritmo pode ser a wff fornecida como um todo ou, então, já dividida entre as subfórmulas que representam as premissas e as que representam as conclusões.

Antes da discussão do algoritmo são apresentadas algumas equivalências lógicas na forma de exemplos, que subsidiam o entendimento da motivação para o estabelecimento de algumas das regras utilizadas no Algoritmo de Wang.

Exemplo 1.49 A Tabela 1.59 mostra a equivalência entre as fórmulas α: $(p \wedge \neg q) \rightarrow r$ e β: $p \rightarrow (r \vee q)$, evidenciando que a fórmula $\alpha \leftrightarrow \beta$ é uma tautologia.

Tabela 1.59 Tabela-verdade da tautologia $\alpha \leftrightarrow \beta$ para α: $(p \wedge \neg q) \rightarrow r$ e β: $p \rightarrow (r \vee q)$.

	p	q	r	¬q	p ∧ ¬q	α	r ∨ q	β	α ↔ β
I_1	v	v	v	f	f	v	v	v	v
I_2	v	v	f	f	f	v	v	v	v
I_3	v	f	v	v	v	v	v	v	v
I_4	v	f	f	v	v	f	f	f	v
I_5	f	v	v	f	f	v	v	v	v
I_6	f	v	f	f	f	f	f	f	v
I_7	f	f	v	v	f	v	v	v	v
I_8	f	f	f	v	f	f	f	v	v

A Tabela 1.60 mostra a equivalência entre as fórmulas α: $p \rightarrow (r \vee \neg q)$ e β: $(p \wedge q) \rightarrow r$, evidenciando que a fórmula $\alpha \leftrightarrow \beta$ é uma tautologia.

Tabela 1.60 Tabela-verdade da tautologia $\alpha \leftrightarrow \beta$ para α: $p \rightarrow (r \vee \neg q)$ e β: $(p \wedge q) \rightarrow r$.

	p	q	r	¬q	r ∨ ¬q	α	p ∧ q	β	α ↔ β
I_1	v	v	v	f	v	v	v	v	v
I_2	v	v	f	f	f	f	v	f	v
I_3	v	f	v	v	v	v	f	v	v
I_4	v	f	f	v	v	v	f	v	v
I_5	f	v	v	f	v	v	f	v	v
I_6	f	v	f	f	f	f	f	v	v
I_7	f	f	v	v	v	v	f	v	v
I_8	f	f	f	v	v	f	f	v	v

Exemplo 1.50 A Tabela 1.61 mostra a equivalência entre as fórmulas α: $(p \vee q) \rightarrow r$ e β: $(p \rightarrow r) \wedge (q \rightarrow r)$, evidenciando que a fórmula $\alpha \leftrightarrow \beta$ é uma tautologia.

Tabela 1.61 Tabela-verdade da tautologia $\alpha \leftrightarrow \beta$ para α: $(p \vee q) \rightarrow r$ e β: $(p \rightarrow r) \wedge (q \rightarrow r)$.

	p	q	r	$p \vee q$	α	$p \rightarrow r$	$q \rightarrow r$	β	$\alpha \leftrightarrow \beta$
I_1	v	v	v	v	v	v	v	v	v
I_2	v	v	f	v	f	f	f	f	v
I_3	v	f	v	v	v	v	v	v	v
I_4	v	f	f	v	f	f	v	f	v
I_5	f	v	v	v	v	v	v	v	v
I_6	f	v	f	v	f	v	f	f	v
I_7	f	f	v	f	v	v	v	v	v
I_8	f	f	f	f	v	v	v	v	v

Exemplo 1.51 A Tabela 1.62 mostra a equivalência entre as fórmulas α: $p \rightarrow (q \wedge r)$ e β: $(p \rightarrow q) \wedge (p \rightarrow r)$, evidenciando que a fórmula $\alpha \leftrightarrow \beta$ é uma tautologia.

Tabela 1.62 Tabela-verdade da tautologia $\alpha \leftrightarrow \beta$ para α: $p \rightarrow (q \wedge r)$ e β: $(p \rightarrow q) \wedge (p \rightarrow r)$.

	p	q	r	$q \wedge r$	α	$p \rightarrow q$	$p \rightarrow r$	β	$\alpha \leftrightarrow \beta$
I_1	v	v	v	v	v	v	v	v	v
I_2	v	v	f	f	f	v	f	f	v
I_3	v	f	v	f	f	f	v	f	v
I_4	v	f	f	f	f	f	f	f	v
I_5	f	v	v	v	v	v	v	v	v
I_6	f	v	f	f	v	v	v	v	v
I_7	f	f	v	f	v	v	v	v	v
I_8	f	f	f	f	v	v	v	v	v

Exemplo 1.52 A Tabela 1.63 mostra que a fórmula α: $(p \wedge q) \rightarrow (q \vee r)$ é uma tautologia.

Tabela 1.63 Tabela-verdade da tautologia α: $(p \wedge q) \rightarrow (q \vee r)$.

	p	q	r	$p \wedge q$	$q \vee r$	α
I_1	v	v	v	v	v	v
I_2	v	v	f	v	v	v
I_3	v	f	v	f	v	v
I_4	v	f	f	f	f	v

Tabela 1.63 *Continuação...*

	p	q	r	$p \wedge q$	$q \vee r$	α
I_5	f	v	v	v	v	v
I_6	f	v	f	f	v	v
I_7	f	f	v	f	v	v
I_8	f	f	f	f	v	v

O Algoritmo de Wang se aplica a uma expressão no esquema

$$p_1, p_2, p_3,..., p_n \Rightarrow c_1, c_2,..., c_k \qquad (1.5)$$

no qual:

1. os p_is ($1 \leq i \leq n$) são identificados como premissas e os c_js ($1 \leq j \leq k$), como conclusões;
2. o símbolo $\Rightarrow$ é a implicação de Wang, que tem a semântica da implicação lógica e que, sintaticamente, funciona como um separador entre as premissas e as conclusões durante a manipulação de subfórmulas pelo algoritmo;
3. as vírgulas entre as premissas são interpretadas como o conectivo lógico $\wedge$ e as vírgulas entre as conclusões são interpretadas como o conectivo lógico $\vee$.

O esquema (1.5) é então interpretado como a wff:

$$p_1 \wedge p_2 \wedge p_3 \wedge .... \wedge p_n \rightarrow c_1 \vee c_2 \vee .... \vee c_k \qquad (1.6)$$

Em termos de valor-verdade – abordagem semântica –, tem-se um teorema se o valor-verdade da wff (1.6) for v. Embora no algoritmo de Wang de prova automática de teorema os valores-verdade não sejam utilizados, uma vez que se trata de uma abordagem puramente sintática, uma compreensão semântica se faz necessária para um completo entendimento do algoritmo.

Para iniciar a prova de um teorema usando o algoritmo de Wang, todas as premissas são escritas à esquerda do símbolo $\Rightarrow$ e a conclusão é escrita à direita do símbolo $\Rightarrow$. Note que tanto premissas quanto conclusões, quando inseridas no esquema, são separadas por vírgulas. As vírgulas que separam as premissas são "entendidas" como o operador lógico $\wedge$ e as que separam as conclusões, como o operador lógico $\vee$. Por exemplo, se as premissas forem: $(p \rightarrow q)$, $(q \rightarrow r)$ e $\neg r$ e se a conclusão for $\neg p$, o esquema inicial de representação fica:

$$(p \rightarrow q), (q \rightarrow r), \neg r \Rightarrow \neg p$$

A seguir, o algoritmo aplica um conjunto de transformações às subfómulas, de maneira a dividilas em outras mais simples que, por sua vez, serão recursivamente submetidas ao mesmo processo.

O algoritmo de Wang sempre termina em um número finito de passos, provando ou não o teorema em questão.

Como comentado anteriormente, o Algoritmo de Wang consiste na aplicação de regras de transformação e de condições de parada. A seguir, são descritas as oito regras que compõem o algoritmo; seis regras são de transformação, identificadas como R_1 a R_6 e as duas regras restantes são de término, nomeadas de R_7 e R_8. O objetivo dos procedimentos recursivos – regras de transformação – é remover conectivos de maneira que as regras de término possam ser aplicadas.

Regra R_1: se uma das fórmulas tem a forma $\neg\alpha$, então se pode tirar a negação e mover a fórmula para o outro lado do símbolo $\Rightarrow$. A wff α pode ser qualquer uma e estar em qualquer um dos lados de $\Rightarrow$, como mostrado a seguir. O Exemplo 1.49 mostra que tal "movimentação" é viável uma vez que não interfere no valor semântico da fórmula.

$p \vee q, \neg(r \wedge s), p \vee r \Rightarrow s, q$	$p, q, r \Rightarrow \neg s, q$
$\Downarrow R_1$	$\Downarrow R_1$
$p \vee q, p \vee r \Rightarrow s, q, (r \wedge s)$	$p, q, r, s \Rightarrow q$

Regra R_2: se uma fórmula à esquerda de $\Rightarrow$ tem a forma $\alpha \wedge \beta$, ou se uma forma à direita de $\Rightarrow$ tem a forma $\alpha \vee \beta$, o conectivo pode ser substituído por uma vírgula. O conectivo $\wedge$ a ser substituído deve ser o conectivo principal da sentença à esquerda de $\Rightarrow$. Observação análoga vale para o conectivo $\vee$ com relação à sentença à direita de $\Rightarrow$.

$p, r \wedge (\neg p \vee s) \Rightarrow \neg p \wedge \neg r$	$r \Rightarrow \neg s \vee q$
$\Downarrow R_2$	$\Downarrow R_2$
$p, r, (\neg p \vee s) \Rightarrow \neg p \wedge \neg r$	$r \Rightarrow \neg s, q$

Regra R_3: se uma fórmula à esquerda de $\Rightarrow$ tem a forma $\alpha \vee \beta$, pode-se remover o operador $\vee$ e separar seus dois operandos, de forma a dividir o teorema sendo provado em dois novos teoremas. Cada um dos dois teoremas obtidos pela separação deve ser provado individualmente. O $\vee$ deve ser o conectivo principal à esquerda. O Exemplo 1.50 mostra, em um caso particular, a equivalência entre a expressão lógica e a conjunção das duas subexpressões obtidas pela divisão.

$$
\begin{array}{ccc}
 & r, \neg p \vee s \Rightarrow t \vee \neg s & \\
\swarrow & R_3 & \searrow \\
r, \neg p \Rightarrow t \vee \neg s & & r, s \Rightarrow t \vee \neg s
\end{array}
$$

Regra R_4: se uma fórmula à direita de $\Rightarrow$ tem a forma $\alpha \wedge \beta$, pode-se remover o operador $\wedge$ e separar seus dois operandos, de forma a dividir o teorema original em dois novos teoremas. Cada um dos dois teoremas obtidos pela separação deve ser provado individualmente. O $\wedge$ deve ser o conectivo principal à direita. O Exemplo 1.51 mostra, em um caso particular, a equivalência entre a expressão lógica e a conjunção das duas subexpressões obtidas pela divisão.

$$\begin{array}{ccc} & r, t \Rightarrow \neg q, \neg r \wedge s & \\ \swarrow & R_4 & \searrow \\ r, t \Rightarrow \neg q, \neg r & & r, t \Rightarrow \neg q, s \end{array}$$

É importante observar que a aplicação das regras R_3 ou R_4 provoca um crescimento exponencial do número de teoremas a serem provados. Em geral, se existirem um total de m operadores $\vee$ e $\wedge$ como conectivos principais à esquerda e à direita de $\Rightarrow$, respectivamente, um total de 2^m novos teoremas resultam como conseqüência da aplicação dessas duas regras.

Regra R_5: uma fórmula em qualquer nível, que tenha a forma $(\alpha \rightarrow \beta)$, pode ser substituída pela fórmula equivalente $(\neg\alpha \vee \beta)$, eliminando com isso a implicação.

$p \rightarrow q, r \wedge (\neg p \vee s) \Rightarrow r, s$	$t, w \Rightarrow p \rightarrow q, \neg s \vee q$
$\Downarrow R_5$	$\Downarrow R_5$
$\neg p \vee q, r \wedge (\neg p \vee s) \Rightarrow r, s$	$t, w \Rightarrow \neg p \vee q, \neg s, q$

Regra R_6: uma fórmula em qualquer nível, que tenha a forma $(\alpha \leftrightarrow \beta)$, pode ser substituída pela fórmula equivalente $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$, eliminando com isso a dupla implicação.

$r, (\neg p \vee s), p \leftrightarrow q \Rightarrow r, s$	$t, w \Rightarrow p \leftrightarrow q, \neg s \vee q$
$\Downarrow R_6$	$\Downarrow R_6$
$r, (\neg p \vee s), (p \rightarrow q) \wedge (q \rightarrow p) \Rightarrow r, s$	$t, w \Rightarrow (p \rightarrow q) \wedge (q \rightarrow p), \neg s, q$

As regras R_1, R_2, R_3, R_4, R_5 e R_6 devem ser aplicadas quantas vezes forem necessárias para remover os conectivos $\leftrightarrow$, $\rightarrow$, $\neg$, $\wedge$ e $\vee$, até que:

Regra R_7: um teorema é considerado provado se alguma fórmula α ocorre em ambos os lados de $\Rightarrow$. Tal teorema é chamado de axioma. Nenhuma transformação será mais necessária nesse teorema, muito embora possam existir outros a serem provados. O teorema original não estará provado até que todos os teoremas obtidos a partir dele tenham sido provados. Portanto, esta regra deve ser verificada para todo novo teorema que eventualmente resulte da aplicação das regras de transformação.

O Exemplo 1.52 mostra uma situação de ocorrência de uma mesma fórmula em ambos os lados de uma implicação lógica.

$r, (\neg p \vee s), p \Rightarrow p, r, s$	$(s \vee q), t, w \Rightarrow (s \vee q)$
$\Downarrow R_7$	$\Downarrow R_7$
teorema	teorema

Regra R_8: um teorema é provado inválido se todas as fórmulas que nele comparecem são símbolos atômicos individuais – isto é, não existem mais conectivos – e um mesmo símbolo não ocorre em ambos os lados de $\Rightarrow$. Se um teorema como este é encontrado, o algoritmo termina. A conclusão inicial não é uma conseqüência lógica das premissas, ou seja, a expressão fornecida ao algoritmo não é um teorema.

$r, s, p \Rightarrow q, w$
$\Downarrow R_8$
não-teorema

O Algoritmo de Wang sempre converge para a solução de um dado problema. Toda aplicação de uma transformação conduz a algum progresso no sentido de eliminar um conectivo e, assim, diminuir sintaticamente o tamanho do teorema – mesmo que com isso sejam criados outros teoremas, como acontece no caso da aplicação das regras R_3 e R_4. A Tabela 1.64 resume os padrões de transformações do Algoritmo de Wang.

Tabela 1.64 Regras do Algoritmo de Wang.

R_1	$(\ldots,\neg\alpha,\ldots\Rightarrow\ldots,\beta)$	torna-se	$(\ldots\Rightarrow\ldots,\beta,\alpha)$
R_1	$(\alpha,\ldots\Rightarrow\ldots,\neg\beta,\ldots)$	torna-se	$(\beta,\alpha,\ldots\Rightarrow\ldots)$
R_2	$(\ldots,\alpha \wedge \beta,\ldots\Rightarrow\ldots)$	torna-se	$(\ldots,\alpha,\beta,\ldots\Rightarrow\ldots)$
R_2	$(\ldots\Rightarrow\ldots,\alpha \vee \beta,\ldots)$	torna-se	$(\ldots\Rightarrow\ldots,\alpha,\beta,\ldots)$
R_3	$(\ldots,\alpha \vee \beta,\ldots\Rightarrow\ldots)$	torna-se	$(\ldots,\alpha,\ldots\Rightarrow\ldots)$ e $(\ldots,\beta,\ldots\Rightarrow\ldots)$
R_4	$(\ldots\Rightarrow\ldots,\alpha \wedge \beta,\ldots)$	torna-se	$(\ldots\Rightarrow\ldots,\alpha,\ldots)$ e $(\ldots\Rightarrow\ldots,\beta,\ldots)$
R_5	$(\ldots\Rightarrow\ldots,\alpha \rightarrow \beta,\ldots)$	torna-se	$(\ldots\Rightarrow\ldots,\neg\alpha \vee \beta,\ldots)$
R_5	$(\ldots,\alpha \rightarrow \beta,\ldots\Rightarrow\ldots)$	torna-se	$(\ldots,\neg\alpha \vee \beta,\ldots\Rightarrow\ldots)$
R_6	$(\ldots,\alpha \leftrightarrow \beta,\ldots\Rightarrow\ldots)$	torna-se	$(\ldots,(\alpha \rightarrow \beta) \wedge (\alpha \rightarrow \beta),\ldots\Rightarrow\ldots)$
R_6	$(\ldots\Rightarrow\ldots,\alpha \leftrightarrow \beta,\ldots)$	torna-se	$(\ldots\Rightarrow\ldots,(\alpha \rightarrow \beta) \wedge (\alpha \rightarrow \beta),\ldots)$
R_7	$(\ldots,\alpha,\ldots\Rightarrow\ldots,\alpha,\ldots)$	torna-se	v, ou seja, é um teorema.

Exemplo 1.53 A Figura 1.2 mostra os passos do algoritmo de Wang para provar que a conclusão $(p \wedge q)$ segue das duas premissas (p_1) $(p \wedge q) \rightarrow r$ e (p_2) q. Durante a prova, o algoritmo evidencia que $(p \wedge q) \rightarrow r, q \Rightarrow p \wedge q$ não é um teorema, ao também evidenciar que uma das subexpressões geradas durante a prova $(r, q \Rightarrow p)$ não é um teorema.

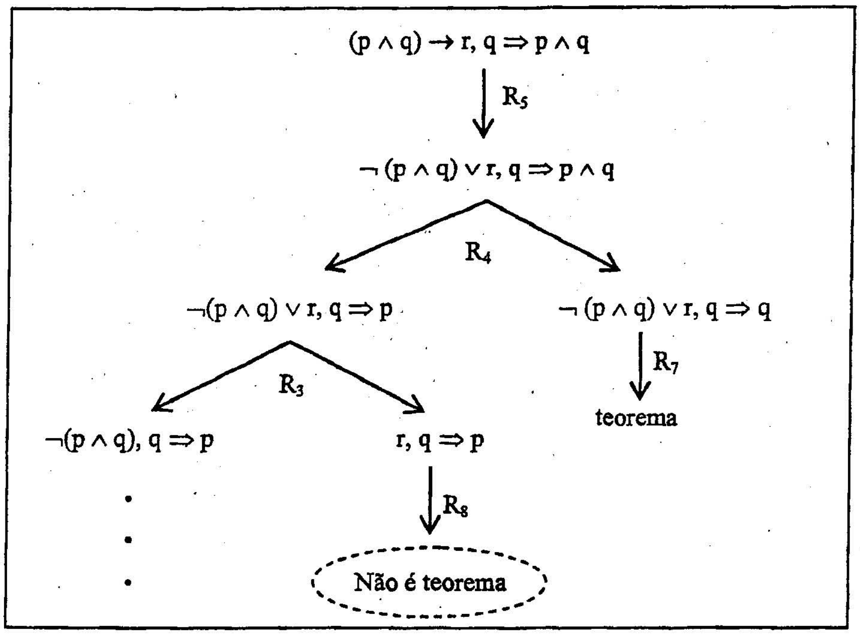


Figura 1.2 A expressão $((p \wedge q) \rightarrow r \wedge q) \rightarrow (p \wedge q)$ não é um teorema, uma vez que, durante sua tentativa de prova, uma das subexpressões geradas não é qualificada como teorema.

Como pode ser evidenciado na Tabela 1.65, a expressão $(p \wedge q)$ não é conseqüência lógica das premissas $(p \wedge q) \rightarrow r$ e q.

Tabela 1.65 Tabela-verdade que evidencia que $p \wedge q$ não é conseqüência lógica de $(p \wedge q) \rightarrow r$ e q.

	p	q	r	$(p \wedge q)$	$(p \wedge q) \rightarrow r$	$((p \wedge q) \rightarrow r) \wedge q$	$((p \wedge q) \rightarrow r) \wedge q \rightarrow (p \wedge q)$
I_1	v	v	v	v	v	v	v
I_2	v	v	f	v	f	f	v
I_3	v	f	v	f	v	f	v
I_4	v	f	f	f	v	f	v
I_5	f	v	v	f	v	v	f
I_6	f	v	f	f	v	v	f
I_7	f	f	v	f	v	f	v
I_8	f	f	f	f	v	f	v

Exemplo 1.54 A Figura 1.3 ilustra o uso do algoritmo de Wang para provar que a conclusão $\neg p$ segue das três premissas (p_1) $p \to q$, (p_2) $q \to r$ e (p_3) $\neg r$.

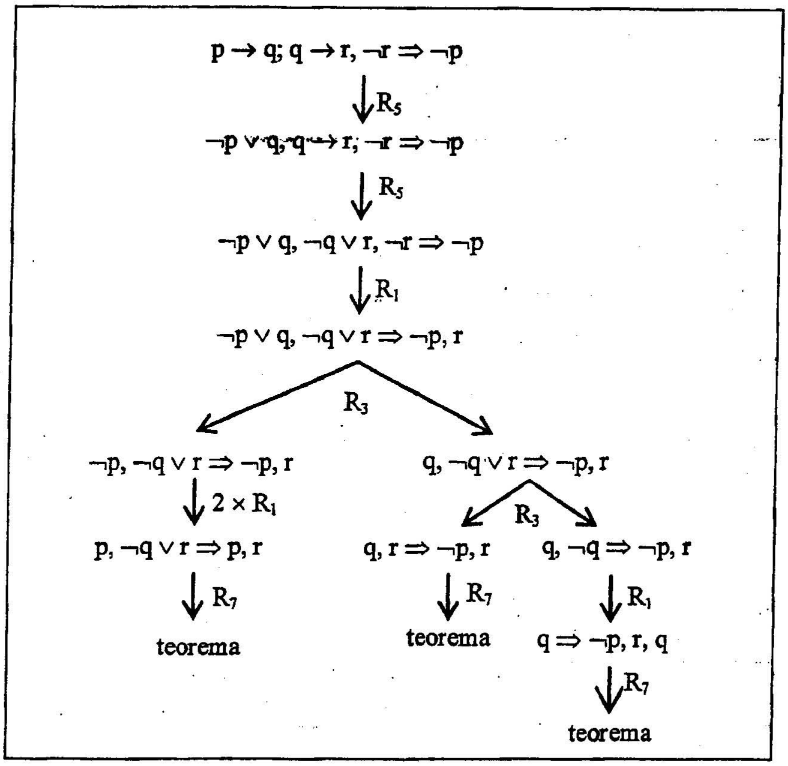


Figura 1.3 Prova da validade do teorema $((p \to q) \wedge (q \to r) \wedge \neg r) \to (\neg p)$ usando Wang.

Exemplo 1.55 A Figura 1.4 ilustra o uso do algoritmo de Wang para provar que a expressão $(p \wedge q) \leftrightarrow (p \wedge p)$ é um teorema. Note que, nesse caso, o algoritmo é iniciado com a expressão toda colocada à direita do símbolo de Wang ($\Rightarrow$).

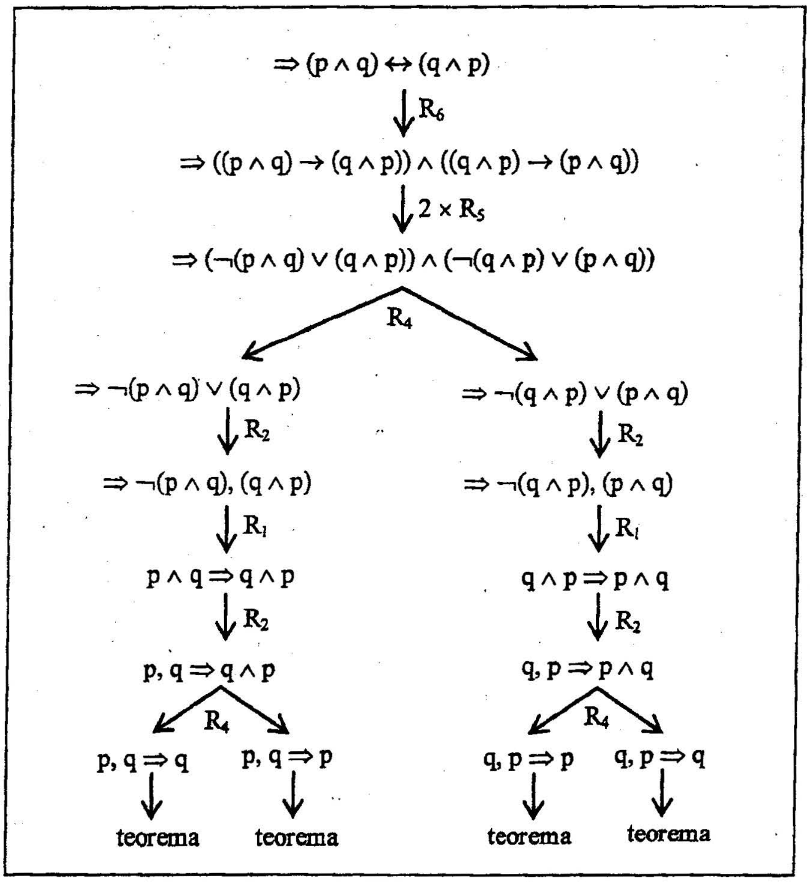


Figura 1.4 Prova da validade do teorema $(p \wedge q) \leftrightarrow (p \wedge p)$ usando o algoritmo de Wang.

1.9 RESOLUÇÃO E PROCEDIMENTOS DE PROVA POR RESOLUÇÃO

O procedimento de prova por resolução é bem geral. Ele é um método sintático de prova que se fundamenta no uso de uma simples regra de inferência, o que torna sua aplicação fácil, vantajosa e computacionalmente conveniente.

Resolução pode ser aplicada apenas àquelas wffs denominadas cláusulas – que, como visto, são wffs que consistem em uma disjunção de literais, i.e., disjunção de átomos/átomos negados. A regra da resolução, quando aplicável, é aplicada a um par de cláusulas-pais e produz, como resultado, uma cláusula derivada, chamada de resolvente – a regra da resolução permite combinar duas fórmulas em uma terceira, por meio da eliminação de átomos complementares. O princípio de resolução em Lógica Proposicional é estabelecido no Teorema 1.4.

Definição 1.17 Considere duas cláusulas α e β (conjuntos de literais). Se existe um literal p tal que $p \in \alpha$ e $\neg p \in \beta$, então o *resolvente* $(\alpha,\beta;p)$ de α e β com relação ao literal p (ou $\neg p$) é a cláusula $(\alpha - \{p\}) \cup (\beta - \{\neg p\})$.

Exemplo 1.56 Considere duas cláusulas α e β representadas como conjuntos de literais. Se α: $\{\neg p, r\}$ e β: $\{q, \neg r\}$ então resolvente(α,β;r): $\{\neg p, q\}$. Se α: $\{\neg p, q, r\}$ e β: $\{\neg q, \neg r\}$ então resolvente(α,β;r): $\{\neg p, q, \neg q\}$ e resolvente(α,β;q): $\{\neg p, r, \neg r\}$.

Teorema 1.4 (Princípio de Resolução para a Lógica Proposicional) Considere duas cláusulas α e β e seja p um literal tal que $p \in \alpha$ e $\neg p \in \beta$. Então:

$$\{\alpha,\beta\} \models \text{resolvente}(\alpha,\beta;p)$$

ou seja, o resolvente de duas cláusulas α e β é conseqüência lógica das duas cláusulas.

Exemplo 1.57 Considere as duas cláusulas C_1: $\neg p \vee q$ e C_2: $r \vee \neg q$. Note que a fórmula atômica q comparece em C_1 (como q) e em C_2 (como $\neg q$), ou seja, q está ocorrendo em C_1 como literal positivo e em C_2 como literal negativo, o que é uma pré-condição para o uso de resolução. A regra da resolução aplicada a C_1 e C_2 produz como resolvente(C_1,C_2;q) a cláusula C_3: $\neg p \vee r$, como mostra a Figura 1.5.

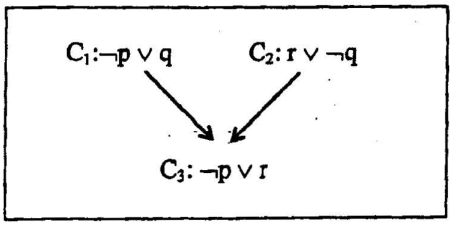


Figura 1.5 Resolução aplicada às cláusulas C_1 e C_2, produzindo a cláusula C_3:resolvente(C_1,C_2;q).

Exemplo 1.58 Considere duas wffs: α: $\neg p \rightarrow q$ e β: $q \rightarrow r$ às quais se pretende aplicar resolução. A regra da resolução é aplicada a cláusulas e, conseqüentemente, as fórmulas em questão devem estar representadas como cláusulas. A maneira de fazer isso é reescrevê-las na Forma Normal Conjuntiva, como foi visto na Seção 1.6.1. Portanto, FNC(α): $p \vee q$ e FNC(β): $\neg q \vee r$. A Figura 1.6 mostra o uso de resolução nas duas wffs.

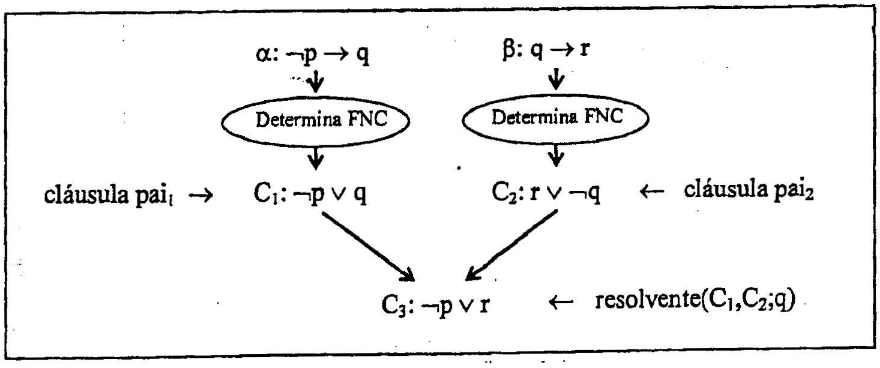


Figura 1.6 Resolução aplicada a duas wffs cujas respectivas expressões na forma normal conjuntiva têm apenas uma cláusula.

O uso de resolução para a prova de teoremas está conjugado à estratégia de prova por redução ao absurdo. A prova pode ser conduzida de duas maneiras: (1) negação da conclusão e (2) negação de todo o teorema. Os procedimentos são praticamente os mesmos; no caso (1), entretanto, lida-se com cada premissa individualmente e com a conclusão negada e, no caso (2), lida-se com todo o teorema negado. Os passos para o caso (1) são:

1. achar para cada premissa e para a conclusão negada (adotada como premissa) a respectiva FNC, como descrito na Seção 1.6.1;
2. cada premissa é agora uma conjunção de uma ou mais cláusulas. Individualizar cada cláusula;
3. cada cláusula é uma disjunção de um ou mais literais; estão, portanto, na forma correta para a aplicação de resolução. Procurar, então, por duas cláusulas que contenham o mesmo átomo com sinais opostos. Nas duas cláusulas a seguir, p é o átomo em questão. Em C_1 ele aparece negado (i.e., um literal negativo) e em C_2 ele aparece sem o sinal de negação (i.e., um literal positivo).

   $$C_1: q \vee r \vee t \vee \neg p \text{ e } C_2: r \vee p \vee \neg s$$

   Aplicando resolução a C_1 e C_2, o literal em questão é eliminado, e o que resta em ambas as cláusulas é combinado em uma nova cláusula; no caso, a cláusula C_3: $q \vee r \vee t \vee r \vee p \vee \neg s$ passa a ser também uma nova candidata ao uso de resolução, junto a todas as anteriores. A regra é bastante simples de ser aplicada, mesmo porque ela pode ser vista como um simples processo de cancelamento;
4. continuar o processo descrito no item 3 até que se tenha derivado um átomo qualquer e, também, a negação desse átomo (por exemplo, até que se tenha derivado o átomo p e também a sua negação $\neg p$). Ao aplicar resolução a essas duas cláusulas, i.e., C_i: p e C_j: $\neg p$ (ou seja, cláusulas descritas por um único e mesmo literal que comparece em cada uma delas com sinais opostos), obtém-se a cláusula vazia, denotada pelo átomo *nil*, que representa uma contradição, finalizando, então, o uso da resolução para a prova da conclusão. O absurdo (representado pela contradição da derivação de p e também de $\neg p$) decorre da suposição de a conclusão não seguir das premissas. Portanto, a conclusão segue das premissas.

Comparando com a prova da Lógica clássica que faz uso de várias regras de inferência, o método de resolução tem várias vantagens, entre elas:

1. não é necessário o uso de equivalências para rearranjar $p \vee q$ como $q \vee p$, etc. Isso se deve ao fato de que todas as fórmulas envolvidas são colocadas na FNC antes de o método começar a ser aplicado e, particularmente, porque para o método é indiferente a posição, na cláusula, do átomo a ser eliminado;

2. existe apenas uma regra de inferência a ser aplicada, i.e., a resolução; e
3. é fácil ser mecanizado.

A linguagem Prolog está fundamentada no princípio de resolução aplicado a cláusulas de Horn e utiliza em suas provas a estratégia de busca em profundidade.

Exemplo 1.59 Prove, usando resolução, que a conclusão $r \vee s$ segue das premissas $p \vee q$, $p \rightarrow r$ e $q \rightarrow s$.

1. Converta cada uma das premissas para a FNC e individualize cada uma das cláusulas obtidas, escrevendo-a em uma linha separadamente.

FNC($p \vee q$):	$p \vee q$
FNC($p \rightarrow r$):	$(p \rightarrow r) \equiv \neg p \vee r$
FNC($q \rightarrow s$):	$\neg q \vee s$

Tem-se, pois, as seguintes cláusulas:

$$C_1: p \vee q$$

$$C_2: \neg p \vee r$$

$$C_3: \neg q \vee s$$

2. Negue a conclusão e ache a FNC da conclusão negada. Tem-se, pois, que:

Conclusão negada: $\neg(r \vee s)$	
FNC($\neg(r \vee s)$):	$\neg(r \vee s) \equiv \neg r \wedge \neg s$

ou seja, a FNC da conclusão negada é constituída por duas cláusulas, C_4: $\neg r$ e C_5: $\neg s$. Essas duas cláusulas são, então, adicionadas às demais, que descrevem as premissas. O conjunto todo de cláusulas passa então a ser $\{C_1, C_2, C_3, C_4, C_5\}$, e a regra de resolução é aplicada a pares de cláusulas que contenham literais de sinais opostos, como mostra a Tabela 1.66.

Tabela 1.66 Uso de resolução com negação da conclusão

Cláusulas		Comentário
C_1:	$p \vee q$	Cláusula da 1ª premissa
C_2:	$\neg p \vee r$	Cláusula da 2ª premissa
C_3:	$\neg q \vee s$	Cláusula da 3ª premissa

Tabela 1.66 *Continuação...*

Cláusulas		Comentário
C_4:	$\neg r$	Cláusula da negação da conclusão
C_5:	$\neg s$	Cláusula da negação da conclusão
C_6:	$\neg p$	Resolvente da resolução de C_2 e C_4
C_7:	q	Resolvente da resolução de C_1 e C_6
C_8:	$\neg q$	Resolvente da resolução de C_3 e C_5
C_9:	*nil*	Resolvente da resolução de C_7 e C_8

Observação 1.30 A refutação mostrada na Tabela 1.66 pode também ser mostrada por meio da construção de um grafo direcionado acíclico (DAG), como mostra a Figura 1.17:

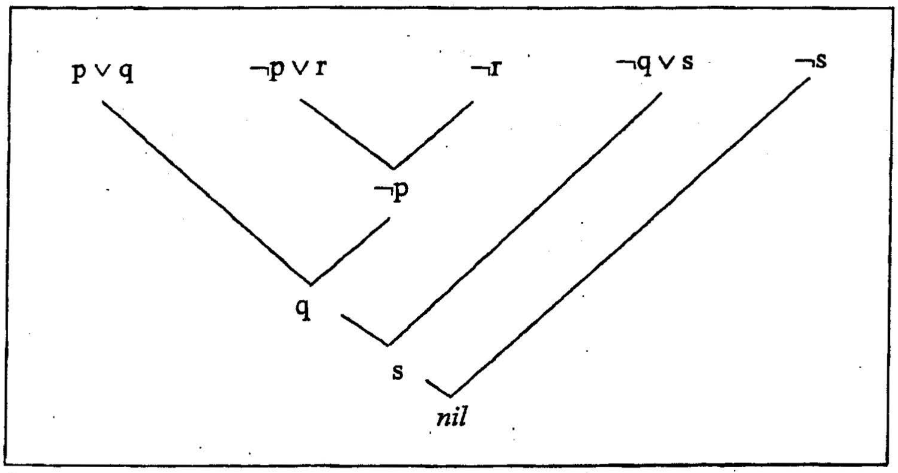


Figura 1.7 Árvore de refutação, cujas folhas são as cláusulas que definem as premissas e a conclusão negada. A cada nível uma cláusula é adicionada como um vértice da árvore por meio do uso de resolução a duas cláusulas pai.

Exemplo 1.60 Verificar se o argumento $\neg p \rightarrow q$, $q \rightarrow r$, $\neg r \vee s$, $\neg s \mid\!- p$ é válido (ou não) usando (1) regras de inferência; (2) o princípio de resolução (com negação da conclusão); e (3) o princípio de resolução com negação do teorema todo.

(1) Uso de regras de inferência. A Tabela 1.67 exibe a prova da conclusão como estabelecida na Definição 1.16. Note que no uso de regras de inferência para derivar a conclusão de um argumento, a notação C_i é usada para indicar as fórmulas que são premissas e as que são derivadas durante o processo, sem que essa notação signifique que a fórmula em questão seja uma cláusula. Isso muda, entretanto, quando o processo de resolução é utilizado, quando C_i indica uma cláusula.

Tabela 1.67 Construção da prova de p por meio de regras de inferência.

Tem-se	C_1	$\neg p \rightarrow q$	premissa
	C_2	$q \rightarrow r$	premissa
	C_3	$\neg r \vee s$	premissa
	C_4	$\neg s$	premissa
Deduz-se	C_5	$\neg r$	(C_3 + C_4 + silogismo disjuntivo)
	C_6	$\neg q \vee r$	(C_3 + equivalência lógica)
	C_7	$\neg q$	(C_5 + C_6 + silogismo disjuntivo)
	C_8	$p \vee q$	(C_1: equivalência)
	C_9	p	(C_7 + C_8 + silogismo disjuntivo)

A seqüência $C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9$ é uma prova da conclusão p e o argumento $\neg p \rightarrow q$, $q \rightarrow r$, $\neg r \vee s$, $\neg s \vdash p$ é válido.

(2) Prova do argumento usando o princípio de resolução (negando a conclusão)

FNC($\neg p \rightarrow q$):	$(\neg p \rightarrow q) \equiv \neg(\neg p) \vee q \equiv p \vee q$
FNC($q \rightarrow r$):	$(q \rightarrow r) \equiv \neg q \vee r$
FNC($\neg r \vee s$):	$\neg r \vee s$
FNC($\neg s$): $\neg s$	$\neg s$
Conclusão negada: $\neg p$ FNC($\neg p$):	$\neg p$

A Tabela 1.68 mostra a prova usando resolução e a Figura 1.18 mostra o DAG associado à refutação.

Tabela 1.68 Uso de resolução com negação da conclusão.

Cláusulas		Comentário
C_1:	$p \vee q$	Cláusula da 1ª premissa
C_2:	$\neg q \vee r$	Cláusula da 2ª premissa
C_3:	$\neg r \vee s$	Cláusula da 3ª premissa
C_4:	$\neg s$	Cláusula da 4ª premissa
C_5:	$\neg p$	Cláusula da negação da conclusão
C_6:	q	Resolvente da resolução de C_1 e C_5

Tabela 1.68 *Continuação...*

Cláusulas		Comentário
C_7:	r	Resolvente da resolução de C_2 e C_6
C_8:	s	Resolvente da resolução de C_3 e C_7
C_9:	*nil*	Resolvente da resolução de C_4 e C_9

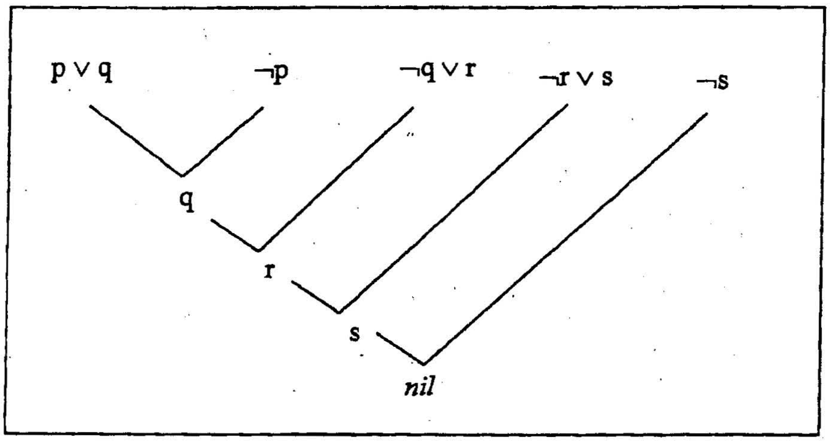


Figura 1.8 Árvore de refutação associada à Tabela 1.68.

A prova por resolução não é única. A Tabela 1.69 mostra uma outra prova, diferente da mostrada na Tabela 1.68.

Tabela 1.69 Uso de resolução com negação da conclusão.

Cláusulas		Comentário
C_1:	$p \vee q$	Cláusula da 1ª premissa
C_2:	$\neg q \vee r$	Cláusula da 2ª premissa
C_3:	$\neg r \vee s$	Cláusula da 3ª premissa
C_4:	$\neg s$	Cláusula da 4ª premissa
C_5:	$\neg p$	Cláusula da negação da conclusão
C_6:	$p \vee r$	Resolvente da resolução de C_1 e C_2
C_7:	$p \vee s$	Resolvente da resolução de C_3 e C_6
C_8:	p	Resolvente da resolução de C_4 e C_7
C_9:	*nil*	Resolvente da resolução de C_5 e C_8

(3) Prova do argumento usando o princípio de resolução (negando todo o teorema), ou seja, negando: $((\neg p \rightarrow q) \wedge (q \rightarrow r) \wedge (\neg r \vee s) \wedge \neg s) \rightarrow p$ e colocando a fórmula resultante na FNC. Tem-se, pois, que:

$$\neg(((\neg p \rightarrow q) \wedge (q \rightarrow r) \wedge (\neg r \vee s) \wedge \neg s) \rightarrow p) \equiv$$

$$\neg(\neg((\neg p \rightarrow q) \wedge (q \rightarrow r) \wedge (\neg r \vee s) \wedge \neg s) \vee p) \equiv$$

$$\neg(\neg((\neg p \rightarrow q) \wedge (q \rightarrow r) \wedge (\neg r \vee s) \wedge \neg s)) \wedge \neg p \equiv$$

$$((\neg p \rightarrow q) \wedge (q \rightarrow r) \wedge (\neg r \vee s) \wedge \neg s) \wedge \neg p \equiv$$

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

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

$$C_1: p \vee q$$

$$C_2: \neg q \vee r$$

$$C_3: \neg r \vee s$$

$$C_4: \neg s$$

$$C_5: \neg p$$

que são as mesmas obtidas quando da negação da conclusão (Tabela 1.68).

Exemplo 1.61 Considere novamente o argumento que foi provado usando regras de inferência, no Exemplo 1.34, representado em Lógica Proposicional por:

$$p \rightarrow q, q \rightarrow r, \neg r \vee p \vdash p \leftrightarrow q$$

O mesmo argumento é provado na Tabela 1.70, usando resolução com a negação da conclusão. Para isso, a FNC de cada uma das premissas e da conclusão negada são determinadas, como segue:

FNC($p \rightarrow q$):	$(p \rightarrow q) \equiv \neg p \vee q$
FNC($q \rightarrow r$):	$(q \rightarrow r) \equiv \neg q \vee r$
FNC($\neg r \vee p$):	$\neg r \vee p$

Conclusão negada: $\neg(p \leftrightarrow q)$ FNC($\neg(p \leftrightarrow q)$):	$\neg(p \leftrightarrow q) \equiv$ $\neg((p \rightarrow q) \wedge (q \rightarrow p)) \equiv$ $\neg(p \rightarrow q) \vee (\neg(q \rightarrow p)) \equiv$ $\neg(\neg p \vee q) \vee (\neg(\neg q \vee p)) \equiv$ $(\neg(\neg p) \wedge (\neg q)) \vee (\neg(\neg q) \wedge (\neg p)) \equiv$ $(p \wedge (\neg q)) \vee (q \wedge (\neg p)) \equiv$ $(p \vee (q \wedge (\neg p)) \wedge ((\neg q) \vee (q \wedge (\neg p)) \equiv$ $((p \vee q) \wedge (p \vee \neg p)) \wedge ((\neg q) \vee q) \wedge ((\neg q) \vee (\neg p)) \equiv$ $(p \vee q) \wedge$ verdade $\wedge$ verdade $\wedge (\neg q \vee (\neg p)) \equiv$ $(p \vee q) \wedge (\neg q \vee (\neg p))$

Tabela 1.70 Uso de resolução com negação da conclusão.

Cláusulas		Comentário
C_1:	$\neg p \vee q$	Cláusula da 1ª premissa
C_2:	$\neg q \vee r$	Cláusula da 2ª premissa
C_3:	$\neg r \vee p$	Cláusula da 3ª premissa
C_4:	$p \vee q$	Cláusula da negação da conclusão
C_5:	$\neg q \vee \neg p$	Cláusula da negação da conclusão
C_6:	$\neg q \vee p$	Resolvente da resolução de C_2 e C_3
C_7:	$\neg p \vee \neg p \equiv \neg p$	Resolvente da resolução de C_1 e C_5
C_8:	$\neg q$	Resolvente da resolução de C_6 e C_7
C_9:	p	Resolvente da resolução de C_4 e C_7
C_{10}:	*nil*	Resolvente da resolução de C_7 e C_9

Exemplo 1.62 Considere novamente o argumento do Exemplo 1.39, representado em Lógica Proposicional por:

$$p \rightarrow q, r \rightarrow s, (q \vee s) \rightarrow t, \neg t \mid\!- \neg p \wedge \neg r$$

A seguir, o argumento é provado usando resolução com a negação da conclusão. Para a prova, a FNC de cada uma das premissas e da conclusão negada são determinadas.

FNC($p \to q$):	$(p \to q) \equiv \neg p \vee q$
FNC($r \to s$):	$(r \to s) \equiv \neg r \vee s$
FNC($(q \vee s) \to t$) :	$((q \vee s) \to t) \equiv$ $\neg(q \vee s) \vee t \equiv$ $(\neg q \wedge (\neg s)) \vee t \equiv$ $(\neg q \vee t) \wedge (\neg s \vee t)$
FNC($\neg t$):	$\neg t$
Conclusão negada: $\neg(\neg p \wedge \neg r)$ FNC($\neg(\neg p \wedge \neg r)$):	$\neg(\neg p \wedge \neg r) \equiv$ $\neg(\neg p) \vee \neg(\neg r) \equiv$ $p \vee r$

A Tabela 1.71 mostra a prova usando resolução e a Figura 1.19 mostra a correspondente árvore de refutação. Outra prova é mostrada na Tabela 1.72.

Tabela 1.71 Uso de resolução com negação da conclusão.

Cláusulas		Comentário
C_1:	$\neg p \vee q$	Cláusula da 1ª premissa
C_2:	$\neg r \vee s$	Cláusula da 2ª premissa
C_3:	$\neg q \vee t$	Cláusula da 3ª premissa
C_4:	$\neg s \vee t$	Cláusula da 3ª premissa
C_5:	$\neg t$	Cláusula da 4ª premissa
C_6:	$p \vee r$	Cláusula da negação da conclusão
C_7:	$\neg p \vee t$	Resolvente da resolução de C_1 e C_3
C_8:	$\neg p$	Resolvente da resolução de C_5 e C_7
C_9:	$\neg s$	Resolvente da resolução de C_4 e C_5
C_{10}:	$\neg r$	Resolvente da resolução de C_2 e C_9
C_{11}:	p	Resolvente da resolução de C_6 e C_{10}
C_{12}:	*nil*	Resolvente da resolução de C_8 e C_{11}

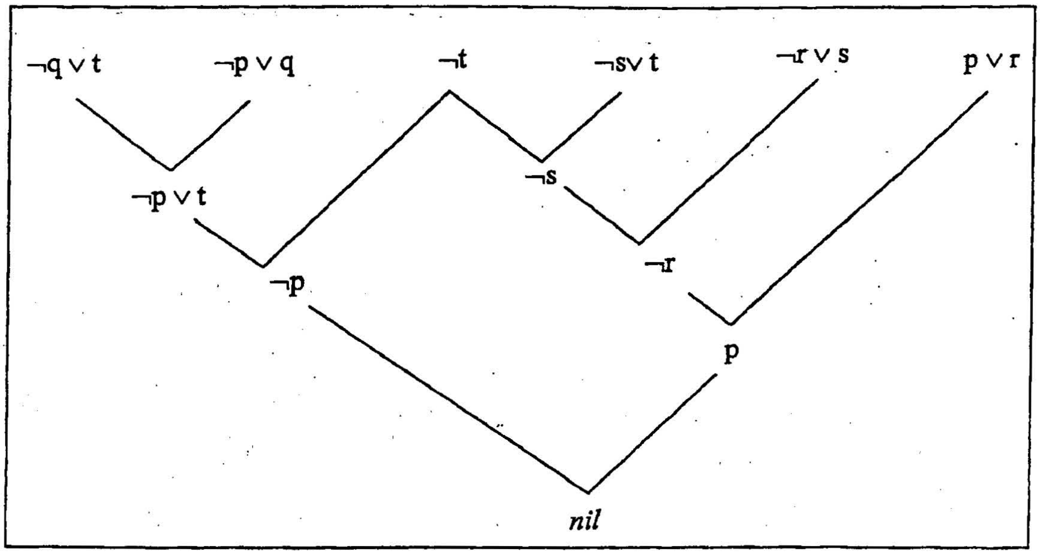


Figura 1. 9 Árvore de refutação associada à Tabela 1.68.

Tabela 1.72 Uso de resolução com negação da conclusão.

Cláusulas		Comentário
C_1:	$\neg p \vee q$	Cláusula da 1ª premissa
C_2:	$\neg r \vee s$	Cláusula da 2ª premissa
C_3:	$\neg q \vee t$	Cláusula da 3ª premissa
C_4:	$\neg s \vee t$	Cláusula da 3ª premissa
C_5:	$\neg t$	Cláusula da 4ª premissa
C_6:	$p \vee r$	Cláusula da negação da conclusão
C_7:	$\neg s$	Resolvente da resolução de C_4 e C_5
C_8:	$\neg r$	Resolvente da resolução de C_2 e C_7
C_9:	p	Resolvente da resolução de C_6 e C_8
C_{10}:	q	Resolvente da resolução de C_1 e C_9
C_{11}:	$\neg q$	Resolvente da resolução de C_3 e C_5
C_{12}:	*nil*	Resolvente da resolução de C_{10} e C_{11}

Para o uso de resolução com negação do teorema, o teorema:

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

é negado e reescrito na FND. Como pode ser visto a seguir, a negação do teorema produz o mesmo conjunto de cláusulas que o produzido quando da negação apenas da conclusão.

$$\neg(((p \to q) \land (r \to s) \land (q \lor s) \to t \land \neg t) \to (\neg p \land \neg r)) \equiv$$

$$\neg(\neg((p \to q) \land (r \to s) \land (q \lor s) \to t \land \neg t)) \lor (\neg p \land \neg r)) \equiv$$

$$((p \to q) \land (r \to s) \land (q \lor s) \to t \land \neg t) \land \neg(\neg p \land \neg r)) \equiv$$

$$(\neg p \lor q) \land (\neg r \lor s) \land (\neg(q \lor s) \lor t) \land \neg t \land (p \lor r) \equiv$$

$$(\neg p \lor q) \land (\neg r \lor s) \land ((\neg q) \land (\neg s)) \lor t) \land \neg t \land (p \lor r) \equiv$$

$$(\neg p \lor q) \land (\neg r \lor s) \land (\neg q \lor t) \land (\neg s \lor t) \land \neg t \land (p \lor r)$$

C_1: $\neg p \lor q$

C_2: $\neg r \lor s$

C_3: $\neg q \lor t$

C_4: $\neg s \lor t$

C_5: $\neg t$

C_6: $p \lor r$

2. A ÁLGEBRA DE BOOLE

2.1 CONCEITOS INICIAIS

Definição 2.1 Seja A um conjunto tal que $|A| = n$. Se $\langle a_1, a_2, ..., a_m\rangle \in A^m$, a m-tupla é uma *m-amostra* de A. Se todos os a_i da m-amostra forem distintos, então a m-amostra é uma *m-permutação* de A. Particularmente, uma n-permutação é chamada simplesmente de *permutação* de A.

Observação 2.1 Dado um conjunto A, a seguinte notação é adotada: $S_m(A)$: conjunto de todas as m-amostras de A; $P_m(A)$: conjunto de todas as m-permutações de A; e $P(A)$: conjunto de todas as permutações de A.

Exemplo 2.1

(a) Considere o conjunto $A=\{a,b,c\}$. A Tabela 2.1 mostra os conjuntos $S_2(A)$, $P_2(A)$, $S_3(A)$ e $P_3(A)$, respectivamente;

(b) Para todo k, $P_k(A) \subseteq S_k(A)$. Especificamente, $P_1(A) = S_1(A)$, e $P_k(A) \subset S_k(A)$, para $k > 1$. Caso $k > |A|$, então $P_k(A) = \varnothing$.

Tabela 2.1 Conjuntos das 2-amostras, das 2-permutações, das 3-amostras e das permutações do conjunto A = {a, b, c}.

$S_2(A)$	{ <a,a>, <a,b>, <a,c>, <b,a>, <b,b>, <b,c>, <c,a>, <c,b>, <c,c> }
$P_2(A)$	{ <a,b>, <a,c>, <b,a>, <b,c>, <c,a>, <c,b> }
$S_3(A)$	{ <a,a,a>, <a,a,b>, <a,a,c>, <a,b,a>, <a,b,b>, <a,b,c>, <a,c,a>, <a,c,b>,<a,c,c>, <b,a,a>, <b,a,b>, <b,a,c>, <b,b,a>, <b,b,a>, <b,b,c>, <b,c,a>, <b,c,b>, <b,c,c>, <c,a,a>, <c,a,b>, <c,a,c>, <c,b,a>, <c,b,b>, <c,b,c>, <c,c,a>, <c,c,b>, <c,c,c> }
$P_3(A) = P(A)$	{ <a,b,c>, <a,c,b>, <b,a,c>, <b,c,a>, <c,a,b>, <c,b,a>}

Definição 2.2 (Álgebra de Boole B) Considere:

Conjunto B
Operações binárias $\oplus$ e $\otimes$ com relação às quais B é fechado.
Operação unária $'$ com relação à qual B é fechado.
Elementos distintos 0 e 1 de B (por distintos entende-se que, enquanto os símbolos a, b, c... representam quaisquer elementos, os símbolos 0 e 1 representam eles próprios).

Axiomas:

Para todo a,b,c $\in$ B,

1A – $a \oplus b = b \oplus a$

1B – $a \otimes b = b \otimes a$

2A – $a \oplus (b \oplus c) = (a \oplus b) \oplus c$

2B – $a \otimes (b \otimes c) = (a \otimes b) \otimes c$

3A – $a \oplus (b \otimes c) = (a \oplus b) \otimes (a \oplus c)$

3B – $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$

4A – $a \oplus 0 = a$

4B – $a \otimes 1 = a$

5A – $a \oplus a' = 1$

5B – $a \otimes a' = 0$

Definição 2.3 Seja S uma assertiva na álgebra booleana B. Se em S os símbolos $\oplus$, $\otimes$, 1 e 0 forem substituídos de acordo com:

Símbolo	Torna-se
$\oplus$	$\otimes$
$\otimes$	$\oplus$
1	0
0	1

a assertiva resultante é a sentença *dual* de S.

Exemplo 2.2 Cada axioma em cada um dos cinco pares de axiomas da Álgebra de Boole (Definição 2.1) é, de fato, dual do outro. A Tabela 2.2 mostra exemplos de assertivas duais.

Tabela 2.2 Assertivas e suas duais.

S	Dual de S
$a \oplus b = b \oplus a$	$a \otimes b = b \otimes a$
$a \otimes b = b \otimes a$	$a \oplus b = b \oplus a$
$1 \oplus (a \oplus b) \otimes c = 0$	$0 \otimes (a \otimes b) \oplus c = 1$
$0 \oplus a = 1$	$1 \otimes a = 0$

Teorema 2.1 (Princípio da dualidade) Se S for um teorema na Álgebra de Boole B, então seu dual é também um teorema em B.

Teorema 2.2 Em uma álgebra booleana B:

(1) se, para todo a, $a \oplus b = a$, então $b = 0$;

(2) se, para todo a, $a \otimes b = a$, então $b = 1$.

Prova de (1): Considere que para todo $a \in B$, $a \oplus b = a$. Particularmente,

(i) $a \oplus b_1 = a$

(ii) $a \oplus b_2 = a$

Substituindo a por b_2 em (i) e a por b_1 em (ii) (o que pode ser feito, uma vez que a assertiva foi estabelecida para todo a), tem-se, respectivamente,

(iii) $b_2 \oplus b_1 = b_2$

(iv) $b_1 \oplus b_2 = b_1$

De (iv), usando o axioma 1A pode-se escrever que:

(v) $b_2 \oplus b_1 = b_1$

De (iii) e (v) segue que $b_2 = b_1$. Assim, o elemento b em $a \oplus b = a$ é único. Pelo axioma 4A, entretanto, $a \oplus 0 = a$. Portanto, o único elemento é o 0. A prova de (2) segue do resultado estabelecido em (1), usando o princípio da dualidade.

Observação 2.2 Os axiomas 4A e 4B estabelecem as propriedades do 0 e do 1, mas não estabelecem que o 0 e o 1 são os únicos elementos que têm essas propriedades. Considere o axioma 4A o qual estabelece que, para todo $a \in B$, $a \oplus 0 = a$. Note que a possibilidade de existir outro elemento $b \in B$, tal que, para todo $a \in B$, $a \oplus b = a$, é deixada em aberto. A parte (1) do Teorema 2.2, entretanto, exclui a possibilidade de existência desse outro elemento. O Teorema 2.2 estabelece que 0 e 1 são os únicos elementos de B, no sentido de que, nas assertivas $a \oplus b = a$ e $a \otimes c = a$, o elemento b só pode ser 0 e o elemento c só pode ser 1. O Teorema 2.3 estabelece a unicidade de a′ para o elemento $a \in B$ (isto é, para todo $a \in B$, existe a′, uma vez que B é fechado sob a operação ′).

Teorema 2.3 Em uma álgebra booleana B, para todo a, se $a \oplus b = 1$ e $a \otimes b = 0$, então $b = a'$.

Teorema 2.4 Em uma álgebra booleana B,

(1) para todo a, $a \oplus a = a$;

(2) para todo a, $a \oplus 1 = 1$;

(3) para todo a e b, $a \oplus (a \otimes b) = a$.

Prova de (1):

$$
\begin{aligned}
a \oplus a &= (a \oplus a) \otimes 1 && (4B)\\
&= (a \oplus a) \otimes (a \oplus a') && (5A)\\
&= a \oplus (a \otimes a') && (3A)\\
&= a \oplus 0 && (5B)\\
&= a
\end{aligned}
$$

Prova de (2):

$$
\begin{aligned}
a \oplus 1 &= a \oplus (a \oplus a') && (5A)\\
&= (a \oplus a) \oplus a' && (2A)\\
&= a \oplus a' && (1)\\
&= 1 && (5A)
\end{aligned}
$$

Teorema 2.5 Em uma álgebra booleana B,

(1) para todo a, $a \otimes a = a$;

(2) para todo a, $a \otimes 0 = 0$;

(3) para todo a e b, $a \otimes (a \oplus b) = a$.

Prova: Segue a partir do Teorema 2.4, usando o princípio de dualidade.

Teorema 2.6 Em uma álgebra booleana B,

(1) para todo a, $(a')' = a$;

(2) $0' = 1$ e $1' = 0$;

(3) para todo a e b, $(a \oplus b)' = a' \otimes b'$ e $(a \otimes b)' = a' \oplus b'$.

Os dois teoremas estabelecidos em (3): $(a \oplus b)' = a' \otimes b'$ e $(a \otimes b)' = a' \oplus b'$ são conhecidos como *Leis De Morgan.*

2.2 FUNÇÕES BOOLEANAS

Apesar de a definição de álgebra de Boole dada na Seção 2.1 ter sido bem geral, o tópico de funções e fórmulas booleanas será desenvolvido utilizando a álgebra de Boole dada pelo sistema $<\{0,1\},\oplus,\otimes,',0,1>$.

Teorema 2.7 O sistema $<\{0,1\},\oplus,\otimes,',0,1>$ é uma álgebra de Boole.

Prova: O sistema dado é precisamente a álgebra de Boole $<B,\oplus,\otimes,',0,1>$ da Seção 2.1, com o conjunto B tendo apenas dois elementos, os elementos distintos 0 e 1. Para provar que o sistema é uma álgebra de Boole, é suficiente provar que $\{0,1\}$ é fechado sob $\oplus$, $\otimes$ e $'$. Na prova serão usados os seguintes resultados:

$$\text{para todo a, } a \oplus a = a;$$

$$\text{para todo a, } a \oplus 1 = 1;$$

$$\text{para todo a, } a \otimes a = a;$$

$$\text{para todo a, } a \otimes 0 = 0; \text{ e}$$

$$0' = 1 \text{ e } 1' = 0.$$

Como os únicos elementos de B são 0 e 1, tem-se que:

$0 \oplus 0 = 0$	$0 \otimes 0 = 0$	$0' = 1$
$0 \oplus 1 = 1$	$0 \otimes 1 = 0$	$1' = 0$
$1 \oplus 0 = 1$	$1 \otimes 0 = 0$	
$1 \oplus 1 = 1$	$1 \otimes 1 = 1$	

Operações são funções, isto é,

$$\oplus: \{0,1\}^2 \to \{0,1\} \text{ é a função } \{<0,0,0>, <0,1,1>, <1,0,1>, <1,1,1>\}$$

$$\otimes: \{0,1\}^2 \to \{0,1\} \text{ é a função } \{<0,0,0>, <0,1,0>, <1,0,0>, <1,1,1>\}$$

$$': \{0,1\} \to \{0,1\} \text{ é a função } \{<0,1>, <1,0>\}$$

Essas três funções são definidas no conjunto $\{0,1\}$ e sobre o conjunto $\{0,1\}$. O conjunto $\{0,1\}$ é, pois, fechado sob essas operações.

Definição 2.4 Seja a álgebra de Boole $<B,\oplus,\otimes,',0,1>$. A função $f(x_1,x_2,...,x_n)$, total de B^n em B ($f: B^n \to B$) é uma *função booleana de n variáveis*.

Exemplo 2.3 Seja $B = \{0,1\}$. Como $|B| = 2$, $|B^n| = 2^n$. O número de funções totais $f:B^n \to B$ é dado por 2^{2^n}. Para n = 1, as quatro funções estão mostradas na Tabela 2.3 e, para n = 2, as 16 funções estão mostradas na Tabela 2.4. A Figura 2.1 mostra um diagrama de cada uma das quatro funções totais da Tabela 2.3.

Tabela 2.3 Quatro funções totais de $\{0,1\} \rightarrow \{0,1\}$.

<0,0>	<0,1>	<1,0>	<1,1>	
X		X		$f_{00} = f_0$
X			X	$f_{01} = f_1$
	X	X		$f_{10} = f_2$
	X		X	$f_{11} = f_3$

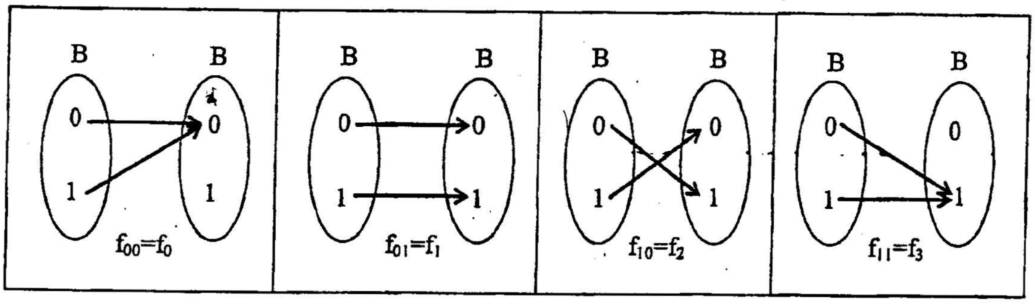


Figura 2.1 As quatro possíveis funções totais de $\{0,1\}$ em $\{0,1\}$, indexadas de acordo com as imagens dos elementos de B.

Observação 2.3 Note nas Tabelas 2.3 e 2.4 que cada função é indexada com um número inteiro. Cada inteiro é a representação decimal do conjunto de valores binários associados pela função. Considere a Tabela 2.4 e a função total identificada por $f_{13}: B^2 \rightarrow B$. O número inteiro 13 tem por representação binária a seqüência de dígitos 1101 e, portanto, $f_{13} = f_{1101} = \{<0,0,1>, <0,1,1>, <1,0,0>, <1,1,1>\}$.

Tabela 2.4 Dezesseis funções totais de $\{0,1\}^2 \rightarrow \{0,1\}$.

<0,0,0>	<0,0,1>	<0,1,0>	<0,1,1>	<1,0,0>	<1,0,1>	<1,1,0>	<1,1,1>	funções
X		X		X		X		$f_{0000} = f_0$
X		X		X			X	$f_{0001} = f_1$
X		X			X	X		$f_{0010} = f_2$
X		X			X		X	$f_{0011} = f_3$
X			X	X		X		$f_{0100} = f_4$
X			X	X			X	$f_{0101} = f_5$
X			X		X	X		$f_{0110} = f_6$
X			X		X		X	$f_{0111} = f_7$
	X	X		X		X		$f_{1000} = f_8$
	X	X		X			X	$f_{1001} = f_9$
	X	X			X	X		$f_{1010} = f_{10}$

Tabela 2.4 *Continuação...*

<0,0,0>	<0,0,1>	<0,1,0>	<0,1,1>	<1,0,0>	<1,0,1>	<1,1,0>	<1,1,1>	funções
	X	X			X		X	$f_{1011} = f_{11}$
	X		X	X		X		$f_{1100} = f_{12}$
	X		X	X			X	$f_{1101} = f_{13}$
	X		X		X	X		$f_{1110} = f_{14}$
	X		X		X		X	$f_{1111} = f_{15}$

Definição 2.5 Seja a álgebra de Boole $<B,\oplus,\otimes,',0,1>$. Sejam f, g e h funções booleanas de n variáveis, $f: B^n \rightarrow B$, $g: B^n \rightarrow B$ e $h: B^n \rightarrow B$. As seguintes *igualdades funcionais* são definidas:

1. $f \oplus g = h$ se e somente se, para todo elemento do conjunto de amostras $S_n(B)$, o resultado da combinação das imagens sob f e g, de acordo com a interpretação de $\oplus$ em B, é igual à imagem do elemento sob h;
2. $f \otimes g = h$ (análogo à (i));
3. $f' = h$ se e somente se a imagem de qualquer elemento de $S_n(B)$ sob h é igual ao complemento (imagem por ') da imagem deste elemento sob f.

Exemplo 2.4 Considere as 16 funções totais definidas na Tabela 2.4 e, particularmente, considere as funções:

$$f_4 = f_{0100} = \{<0,0,0>, <0,1,1>, <1,0,0>, <1,1,0>\}$$

$$f_8 = f_{1000} = \{<0,0,1>, <0,1,0>, <1,0,0>, <1,1,0>\}$$

Pode-se, pois, escrever que:

$$f_4 \oplus f_8 = f_{0100} = \{<0,0,1>, <0,1,1>, <1,0,0>, <1,1,0>\} = f_{1100} = f_{12}.$$

Segue, pois, a igualdade funcional: $f_4 \oplus f_8 = f_{12}$.
De maneira semelhante, considerando as funções:

$$f_5 = f_{0101} = \{<0,0,0>, <0,1,1>, <1,0,0>, <1,1,1>\}$$

$$f_9 = f_{1001} = \{<0,0,1>, <0,1,0>, <1,0,0>, <1,1,1>\}$$

tem-se que $f_5 \otimes f_9 = \{<0,0,0>, <0,1,0>, <1,0,0>, <1,1,1>\} = f_{0001} = f_1$. Segue, então, a igualdade funcional $f_5 \otimes f_9 = f_1$. Focalizando a operação ' considere, por exemplo, a função:

$$f_{13} = f_{1101} = \{<0,0,1>, <0,1,1>, <1,0,0>, <1,1,1>\}.$$

Tem-se, então,

$$f'_{13} = f_{0010} = \{\langle 0,0,0\rangle, \langle 0,1,0\rangle, \langle 1,0,1\rangle, \langle 1,1,0\rangle\} = f_{0010} = f_2.$$

Segue que $f'_{13} = f_2$.

Teorema 2.8 Seja $\langle B,\oplus,\otimes,',0,1\rangle$ uma álgebra de Boole. Sejam f e g funções booleanas totais de $B^n \to B$. São funções:

1. $f \oplus g$;

2. $f \otimes g$; e

3. f'.

Observação 2.4 A Definição 2.5 contempla a definição de igualdade funcional. O Teorema 2.8, a seguir, prova que $f \oplus g$, $f \otimes g$ e f' são de fato funções, como definidas na Definição 2.5. É preciso, pois, provar que se $f \oplus g = h_1$ e $f \oplus g = h_2$ então $h_1 = h_2$.

Teorema 2.9 Seja $\langle B,\oplus,\otimes,',0,1\rangle$ uma álgebra de Boole. Então, o sistema $\langle F_n,\oplus,\otimes,',f_0,f_1\rangle$, no qual F_n é o conjunto de todas as funções totais de B^n em B e as três operações definidas como na Definição 2.5, é uma álgebra de Boole.

Exemplo 2.5 Considere B = {0,1}, bem como o sistema $\langle F_2,\oplus,\otimes,',f_0,f_1\rangle$, no qual F_2 é o conjunto de todas as funções totais de $\{0,1\}^2$ em {0,1}. Temos $|F_2| = 16$, como definidas na Tabela 2.4. As funções f_0 e f_1 são as funções cujos contradomínios são {0} e {1} respectivamente que, no caso do exemplo em questão, são as funções f_0 e f_{15}. Pode ser facilmente verificado que F_2 é fechado sob as operações e que seus elementos satisfazem os axiomas. Considere, por exemplo, o axioma 4A que estabelece que, para qualquer elemento $a \in F_2$, $a \otimes 1 = a$. Uma investigação exaustiva da verificação do axioma está mostrada na Tabela 2.5.

Tabela 2.5 Verificação da validade do axioma 4A ($a \otimes 1 = a$), com $1 = f_1 = \{\langle 0,0,1\rangle, \langle 0,1,1\rangle, \langle 1,0,1\rangle, \langle 1,1,1\rangle\}$.

$f_0 = \{\langle 0,0,0\rangle, \langle 0,1,0\rangle, \langle 1,0,0\rangle, \langle 1,1,0\rangle\} \otimes f_1$	=	$\{\langle 0,0,0\rangle, \langle 0,1,0\rangle, \langle 1,0,0\rangle, \langle 1,1,0\rangle\}$	=	f_0
$f_1 = \{\langle 0,0,0\rangle, \langle 0,1,0\rangle, \langle 1,0,0\rangle, \langle 1,1,1\rangle\} \otimes f_1$	=	$\{\langle 0,0,0\rangle, \langle 0,1,0\rangle, \langle 1,0,0\rangle, \langle 1,1,1\rangle\}$	=	f_1
$f_2 = \{\langle 0,0,0\rangle, \langle 0,1,0\rangle, \langle 1,0,1\rangle, \langle 1,1,0\rangle\} \otimes f_1$	=	$\{\langle 0,0,0\rangle, \langle 0,1,0\rangle, \langle 1,0,1\rangle, \langle 1,1,0\rangle\}$	=	f_2
$f_3 = \{\langle 0,0,0\rangle, \langle 0,1,0\rangle, \langle 1,0,1\rangle, \langle 1,1,1\rangle\} \otimes f_1$	=	$\{\langle 0,0,0\rangle, \langle 0,1,0\rangle, \langle 1,0,1\rangle, \langle 1,1,1\rangle\}$	=	f_3
$f_4 = \{\langle 0,0,0\rangle, \langle 0,1,1\rangle, \langle 1,0,0\rangle, \langle 1,1,0\rangle\} \otimes f_1$	=	$\{\langle 0,0,0\rangle, \langle 0,1,1\rangle, \langle 1,0,0\rangle, \langle 1,1,0\rangle\}$	=	f_4
$f_5 = \{\langle 0,0,0\rangle, \langle 0,1,1\rangle, \langle 1,0,0\rangle, \langle 1,1,1\rangle\} \otimes f_1$	=	$\{\langle 0,0,0\rangle, \langle 0,1,1\rangle, \langle 1,0,0\rangle, \langle 1,1,1\rangle\}$	=	f_5
$f_6 = \{\langle 0,0,0\rangle, \langle 0,1,1\rangle, \langle 1,0,1\rangle, \langle 1,1,0\rangle\} \otimes f_1$	=	$\{\langle 0,0,0\rangle, \langle 0,1,1\rangle, \langle 1,0,1\rangle, \langle 1,1,0\rangle\}$	=	f_6
$f_7 = \{\langle 0,0,0\rangle, \langle 0,1,1\rangle, \langle 1,0,1\rangle, \langle 1,1,1\rangle\} \otimes f_1$	=	$\{\langle 0,0,0\rangle, \langle 0,1,1\rangle, \langle 1,0,1\rangle, \langle 1,1,1\rangle\}$	=	f_7
$f_8 = \{\langle 0,0,1\rangle, \langle 0,1,0\rangle, \langle 1,0,0\rangle, \langle 1,1,0\rangle\} \otimes f_1$	=	$\{\langle 1,0,0\rangle, \langle 0,1,0\rangle, \langle 1,0,0\rangle, \langle 1,1,0\rangle\}$	=	f_8

Tabela 2.5 *Continuação...*

$f_9 = \{<0,0,1>, <0,1,0>, <1,0,0>, <1,1,1>\} \otimes f_1$	=	$\{<1,0,0>, <0,1,0>, <1,0,0>, <1,1,1>\}$	=	f_9
$f_{10} = \{<0,0,1>, <0,1,0>, <1,0,1>, <1,1,0>\} \otimes f_1$	=	$\{<1,0,0>, <0,1,0>, <1,0,1>, <1,1,0>\}$	=	f_{10}
$f_{11} = \{<0,0,1>, <0,1,0>, <1,0,1>, <1,1,1>\} \otimes f_1$	=	$\{<1,0,0>, <0,1,0>, <1,0,1>, <1,1,1>\}$	=	f_{11}
$f_{12} = \{<0,0,1>, <0,1,1>, <1,0,0>, <1,1,0>\} \otimes f_1$	=	$\{<1,0,0>, <0,1,1>, <1,0,0>, <1,1,0>\}$	=	f_{12}
$f_{13} = \{<0,0,1>, <0,1,1>, <1,0,0>, <1,1,1>\} \otimes f_1$	=	$\{<1,0,0>, <0,1,1>, <1,0,0>, <1,1,1>\}$	=	f_{13}
$f_{14} = \{<0,0,1>, <0,1,1>, <1,0,1>, <1,1,0>\} \otimes f_1$	=	$\{<1,0,0>, <0,1,1>, <1,0,1>, <1,1,0>\}$	=	f_{14}
$f_{15} = \{<0,0,1>, <0,1,1>, <1,0,1>, <1,1,1>\} \otimes f_1$	=	$\{<1,0,0>, <0,1,1>, <1,0,1>, <1,1,1>\}$	=	f_{15}

Exemplo 2.6 Considere o sistema $<F_3,\oplus,\otimes,',f_0,f_1>$, no qual F_3 é o conjunto de todas as funções totais de $\{0,1\}^3$ em $\{0,1\}$. Temos $|F_3| = 256$. As funções f_0 e f_1 são as funções cujos contradomínios são $\{0\}$ e $\{1\}$ respectivamente que, no caso do exemplo em questão, são as funções f_0 e f_{255}:

$$0 = f_0 = \{<0,0,0,0>, <0,0,1,0>, <0,1,1,0>, <0,1,1,0>, <1,0,0,0>, <1,0,1,0>, <1,1,0,0>, <1,1,1,0>\}$$

$$1 = f_1 = \{<0,0,0,1>, <0,0,1,1>, <0,1,1,1>, <0,1,1,1>, <1,0,0,1>, <1,0,1,1>, <1,1,0,1>, <1,1,1,1>\}$$

Em razão da maneira como são enumeradas, qualquer uma delas pode ser facilmente descrita. Por exemplo, a função f_{85} é dada por:

$$f_{85} = \{<0,0,0,\mathbf{0}>, <0,0,1,\mathbf{1}>, <0,1,1,\mathbf{0}>, <0,1,1,\mathbf{1}>, <1,0,0,\mathbf{0}>, <1,0,1,\mathbf{1}>, <1,1,0,\mathbf{0}>, <1,1,1,\mathbf{1}>\}$$

uma vez que a representação do número decimal 85 em binário é 01010101.

Suponha outra situação, na qual é conhecida a descrição da função, dada por:

$$f_? = \{<0,0,0,\mathbf{1}>, <0,0,1,\mathbf{0}>, <0,1,1,\mathbf{0}>, <0,1,1,\mathbf{0}>, <1,0,0,\mathbf{1}>, <1,0,1,\mathbf{0}>, <1,1,0,\mathbf{0}>, <1,1,1,\mathbf{1}>\}$$

Como 10001001 é a representação em binário do número decimal 137, resulta que a função $f_?$ em questão é a função f_{137}.

2.3 FÓRMULAS BOOLEANAS

Considere a expressão $(x_1 \oplus x_2)' \oplus (x_1' \otimes x_2')$, na qual x_1 e x_2 são variáveis que representam elementos de B. Apesar da tendência de abordar tal expressão como função, ela certamente não é um conjunto de pares ordenados e, conseqüentemente, não é uma função. A expressão é uma fórmula (ou forma). A razão pela qual as fórmulas são tratadas, algumas vezes, como funções se deve ao fato da existência de uma associação entre uma fórmula com n variáveis e uma função total de B^n em B. Essa associação é tratada nesta subseção.

Definição 2.6 Seja a álgebra de Boole $<B,\oplus,\otimes,',0,1>$ e sejam $x_1, x_2,\ldots, x_n$ variáveis representando elementos de B. Uma *fórmula booleana* é definida recursivamente como:

1. 0 e 1 são fórmulas booleanas;
2. a variável x_i, $i = 1,\ldots, n$ é uma fórmula booleana;
3. se α é uma fórmula booleana, então ~~(α) também o é~~;
4. Se α é uma fórmula booleana, então α' também o é;
5. Se α e β são fórmulas booleanas, então $\alpha \oplus \beta$ também o é;
6. Se α e β são fórmulas booleanas, então $\alpha \otimes \beta$ também o é;
7. Somente as expressões dadas pelas afirmações 1. a 6. são fórmulas booleanas.

Exemplo 2.7 Considere a expressão:

$$(1)\ (x_1 \oplus x_2)' \oplus (x_1' \otimes x_2')$$

A expressão (1) é uma fórmula booleana, se cada uma das expressões,

$$(1.1)\ (x_1 \oplus x_2)' \text{ e}$$

$$(1.2)\ (x_1' \otimes x_2')$$

for uma fórmula booleana.

A expressão (1.1) é uma fórmula booleana se a expressão (1.1.1) $(x_1 \oplus x_2)$ for uma fórmula booleana. A expressão (1.1.1) é uma fórmula booleana se a expressão (1.1.1.1) $x_1 \oplus x_2$ for uma fórmula booleana. A expressão (1.1.1.1) é uma fórmula booleana uma vez que x_1 e x_2 são fórmulas booleanas. Isso faz com que a expressão (1.1.1) seja uma fórmula booleana e, por sua vez, faz com que a expressão (1.1.1) seja uma fórmula booleana que, por sua vez, faz com que a expressão (1.1) seja uma fórmula booleana.

A mesma abordagem pode ser usada para ~~mostrar que a expressão~~ (1.2) é uma fórmula booleana. Portanto, a expressão (1) em questão é uma fórmula booleana válida.

Definição 2.7 Considere a álgebra de Boole $<B,\oplus,\otimes,',0,1>$ e considere uma fórmula de Boole com n variáveis $x_1, x_2,\ldots, x_n$ que representam elementos de B. O conjunto dos elementos das n-amostras de B, $S_n(B)$, é chamado de *conjunto de valores de atribuição,* e a i-ésima coordenada de uma atribuição está associada à variável x_i.

Exemplo 2.8 Considere a fórmula booleana $x_1 \otimes (x_1 \oplus (x_2 \otimes x_3))$ e a atribuição $<1,0,1>$. Com essa atribuição a fórmula fica: $1 \otimes (1 \oplus (0 \otimes 1))$ que, calculada usando as definições de operações para elementos de $\{0,1\}$, como na prova do Teorema 2.7, resulta em: $1 \otimes (1 \oplus (0 \otimes 1)) = 1$. Pode-se, pois, escrever que: "A associação entre as coordenadas de uma atribuição e as variáveis permite a redução de uma fórmula a um valor". Essa redução, chamada de cálculo, estabelece uma ligação entre fórmulas e funções. No caso, um elemento da função total de B^3 em B, associada à fórmula $x_1 \otimes (x_1 \oplus (x_2 \otimes x_3))$, é, por exemplo, <<1,0,1>,1> ou, notado de maneira simplificada, <1,0,1,1>. Cálculos realizados levando em consideração cada um dos elementos de $S_n(B)$ (valores de atribuição) fornecem todos os elementos da função, como mostrado na Tabela 2.6.

Tabela 2.6 Cálculo da fórmula booleana $x_1 \otimes (x_1 \oplus (x_2 \otimes x_3))$ e especificação da função booleana a ela associada. Note que f é o conjunto das oito 4-uplas

$<x_1,x_2,x_3>$	$\alpha = x_2 \otimes x_3$	$\beta = x_1 \oplus \alpha$	$x_1 \otimes \beta$	f	f
<0,0,0>	0	0	0	{<<0,0,0>,0>	{<0,0,0,0>
<0,0,1>	0	0	0	<<0,0,1>,0>	<0,0,1,0>
<0,1,0>	0	0	0	<<0,1,0>,0>	<0,1,0,0>
<0,1,1>	1	1	0	<<0,1,1>,0>	<0,1,1,0>
<1,0,0>	0	1	1	<<1,0,0>,1>	<1,0,0,1>
<1,0,1>	0	1	1	<<1,0,1>,1>	<1,0,1,1>
<1,1,0>	0	1	1	<<1,1,0>,1>	<1,1,0,1>
<1,1,1>	1	1	1	<<1,1,1>,1>}	<1,1,1,1>}

Definição 2.8 Duas fórmulas booleanas são *equivalentes* se seus valores são iguais, para todo elemento do conjunto de atribuição.

Exemplo 2.9 Na Tabela 2.6 pode ser observado que as fórmulas booleanas x_1 e $x_1 \otimes (x_1 \oplus (x_2 \otimes x_3))$ são equivalentes. A equivalência das fórmulas permite escrever que $x_1 = x_1 \otimes (x_1 \oplus (x_2 \otimes x_3))$. Uma vez que as fórmulas são equivalentes, as funções a elas associadas são iguais.

Note que apesar de x_1 ser uma fórmula definida usando apenas uma variável (x_1 no caso), a função associada a ela deve ser total de $\{0,1\}^3$ em $\{0,1\}$, no caso específico de igualdade entre as funções associadas a x_1 e a $x_1 \otimes (x_1 \oplus (x_2 \otimes x_3))$. Para que se possa afirmar que funções correspondentes às fórmulas booleanas são iguais, é preciso considerar o mesmo conjunto de atribuição.

Definição 2.9 Seja a álgebra de Boole $<B,\oplus,\otimes,',0,1>$ e $\alpha(x_1, x_2,..., x_n)$ uma fórmula booleana nas variáveis $x_1, x_2,..., x_n$, que representam elementos de B. Os elementos do conjunto das n-amostras $S_n(\{0,1\})$, chamados de valores binários de atribuição, são denotados por t^n_m, ($m = 0,...,2^n - 1$). A amostra $<m_1,m_2,...,m_n>$ é denotada por t^n_m quando o número binário $m_1m_2...m_n$ for igual a m (número decimal).

Seja F o conjunto das fórmulas $\alpha(m_1, m_2,..., m_n)$ e seja a função $q: S_n \rightarrow F$ tal que $q(t^n_m) = \alpha(m_1, m_2,..., m_n)$ se e somente se $t^n_m = <m_1,m_2,...,m_n>$. Define-se, a seguir, a função de avaliação binária v:

$F \rightarrow \{0,1\}$ tal que $v(\alpha(m_1, m_2,..., m_n)) = 0$ se e somente se $\alpha(m_1, m_2,..., m_n) = 0$ quando calculada de acordo com as regras do Teorema 2.7 e $v(\alpha(m_1, m_2,..., m_n)) = 1$ se e somente se $\alpha(m_1, m_2,..., m_n) = 1$ quando calculada de acordo com as regras do Teorema 2.7.

Exemplo 2.10 A notação introduzida na Definição 2.9 é particularizada, no exemplo que segue, para $n = 3$. Considere:

$$B^3 = \{<0,0,0>, <0,0,1>, <0,1,0>, <0,1,1>, <1,0,0>, <1,0,1>, <1,1,0>, <1,1,1>\}$$

e

$$t_0^3 = <0,0,0> \quad t_4^3 = <1,0,0>$$
$$t_1^3 = <0,0,1> \quad t_5^3 = <1,0,1>$$
$$t_2^3 = <0,1,0> \quad t_6^3 = <1,1,0>$$
$$t_3^3 = <0,1,1> \quad t_7^3 = <1,1,1>$$

e as funções q e v como mostradas na Figura 2.2.

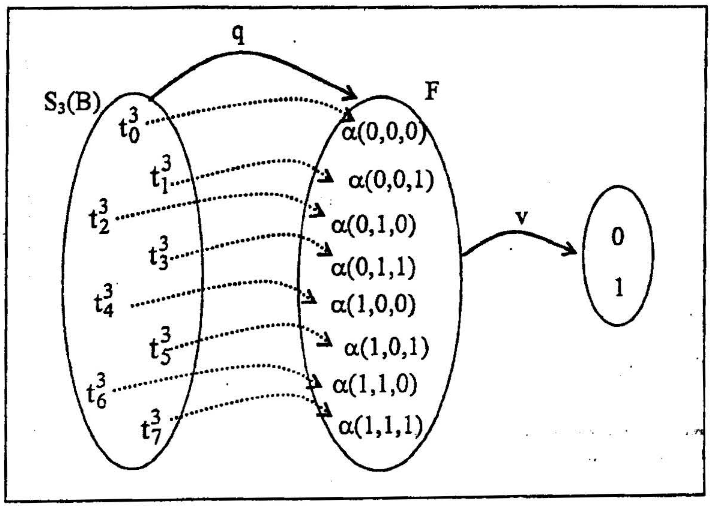


Figura 2.2 Representação pictórica das funções q e v da Definição 2.9.

Exemplo 2.11 Considere $\alpha(x_1, x_2, x_3) = x_1 \otimes (x_1 \oplus (x_2 \otimes x_3))$. Então, $t_5^3 = <1,0,1>$ e $q(t_5^3) = \alpha(<1,0,1>) = 1 \otimes (1 \oplus (0 \otimes 1))$ e $v(q(t_5^3)) = 1$.

Se $B = \{0,1\}$, então o conjunto $\{< t_m^3, v(q(t_m^3))>, m = 0,...,7\}$ é a função associada com a fórmula booleana $x_1 \otimes (x_1 \oplus (x_2 \otimes x_3))$. Escrevendo por extenso, tem-se que a função é:

$\{< t_0^3, v(q(t_0^3))>, < t_1^3, v(q(t_1^3))>, < t_2^3, v(q(t_2^3))>, < t_3^3, v(q(t_3^3))>, < t_4^3, v(q(t_4^3))>, < t_5^3, v(q(t_5^3))>, < t_6^3, v(q(t_6^3))>, < t_7^3, v(q(t_7^3))>\}$.

Exemplo 2.12 Considere $\alpha(x_1, x_2, x_3) = (x_1 \oplus x_2) \otimes (x_1 \oplus x_3')$. O cálculo binário da fórmula está mostrado na Tabela 2.7.

Tabela 2.7 Cálculo da fórmula booleana $(x_1 \oplus x_2) \otimes (x_1 \oplus x_3')$ e especificação da função booleana a ela associada. Note que a função é o conjunto das 8 4-uplas da coluna f.

m	t^3_m	$q(t^3_m)$	$v(q(t^3_m))$	f
0	<0,0,0>	$(0 \oplus 0) \otimes (0 \oplus 0')$	0	{<0,0,0,0>
1	<0,0,1>	$(0 \oplus 0) \otimes (0 \oplus 1')$	0	<0,0,1,0>
2	<0,1,0>	$(0 \oplus 1) \otimes (0 \oplus 0')$	1	<0,1,0,1>
3	<0,1,1>	$(0 \oplus 1) \otimes (0 \oplus 1')$	0	<0,1,1,0>
4	<1,0,0>	$(1 \oplus 0) \otimes (1 \oplus 0')$	1	<1,0,0,1>
5	<1,0,1>	$(1 \oplus 0) \otimes (1 \oplus 1')$	1	<1,0,1,1>
6	<1,1,0>	$(1 \oplus 1) \otimes (1 \oplus 0')$	1	<1,1,0,1>
7	<1,1,1>	$(1 \oplus 1) \otimes (1 \oplus 1')$	1	<1,1,1,1>}

Exemplo 2.13 Considere $\alpha(x_1, x_2, x_3) = (x_1 \otimes x_2)' \oplus (x_2 \oplus x_3)'$. O cálculo binário da fórmula está mostrado na Tabela 2.8, bem como a função booleana a ela associada.

Tabela 2.8 Cálculo da fórmula booleana $(x_1 \otimes x_2)' \oplus (x_2 \oplus x_3)'$ e especificação da função booleana a ela associada. Note que a função é o conjunto das 8 4-uplas da coluna f.

m	t^3_m	$q(t^3_m)$	$v(q(t^3_m))$	f
0	<0,0,0>	$(0 \otimes 0)' \oplus (0 \oplus 0)'$	1	{<0,0,0,1>
1	<0,0,1>	$(0 \otimes 0)' \oplus (0 \oplus 1)'$	1	<0,0,1,1>
2	<0,1,0>	$(0 \otimes 1)' \oplus (1 \oplus 0)'$	1	<0,1,0,1>
3	<0,1,1>	$(0 \otimes 1)' \oplus (1 \oplus 1)'$	1	<0,1,1,1>
4	<1,0,0>	$(1 \otimes 0)' \oplus (0 \oplus 0)'$	1	<1,0,0,1>
5	<1,0,1>	$(1 \otimes 0)' \oplus (0 \oplus 1)'$	1	<1,0,1,1>
6	<1,1,0>	$(1 \otimes 1)' \oplus (1 \oplus 0)'$	1	<1,1,0,1>
7	<1,1,1>	$(1 \otimes 1)' \oplus (1 \oplus 1)'$	1	<1,1,1,1>}

Definição 2.10 Seja a álgebra de Boole $\langle B,\oplus,\otimes,',0,1\rangle$ e sejam as variáveis $x_1, x_2,..., x_n$, e considere que a_i representa x_i ou x_i'.

A fórmula $a_1 \otimes a_2 \otimes ... \otimes a_n$ é denominada *minterm* (polinômio minimal, produto completo, produto fundamental) e a fórmula $a_1 \oplus a_2 \oplus ... \oplus a_n$ é denominada *maxterm* (polinômio máximo, soma completa ou soma fundamental), de n variáveis.

Convenção: $x_i^1 = x_i$ e $x_i^0 = x_i'$ e, sob essa convenção, a seguinte notação será adotada:

$$\text{minterm: } x_i^{m_1} \otimes x_i^{m_2} \otimes \ldots \otimes x_i^{m_n} \text{ será representado por } min_m{}^n \text{ e}$$

$$\text{maxterm: } x_i^{m_1}{}' \oplus x_i^{m_2}{}' \oplus \ldots \oplus x_i^{m_n}{}' \text{ será representado por } max_m{}^n,$$

quando o número binário $m_1m_2m_3...m_n = m$. Os a_is são chamados de literais.

Exemplo 2.14 No caso particular de n = 3, tem-se:

$$\begin{cases} x_1' \otimes x_2' \otimes x_3' = x_1^0 \otimes x_2^0 \otimes x_3^0 = min_0{}^3 \\ x_1' \otimes x_2' \otimes x_3 = x_1^0 \otimes x_2^0 \otimes x_3^1 = min_1{}^3 \\ x_1' \otimes x_2 \otimes x_3' = x_1^0 \otimes x_2^1 \otimes x_3^0 = min_2{}^3 \\ x_1' \otimes x_2 \otimes x_3 = x_1^0 \otimes x_2^1 \otimes x_3^1 = min_3{}^3 \\ x_1 \otimes x_2' \otimes x_3' = x_1^1 \otimes x_2^0 \otimes x_3^0 = min_4{}^3 \\ x_1 \otimes x_2' \otimes x_3 = x_1^1 \otimes x_2^0 \otimes x_3^1 = min_5{}^3 \\ x_1 \otimes x_2 \otimes x_3' = x_1^1 \otimes x_2^1 \otimes x_3^0 = min_6{}^3 \\ x_1 \otimes x_2 \otimes x_3 = x_1^1 \otimes x_2^1 \otimes x_3^1 = min_7{}^3 \text{ e} \end{cases}$$

$$\begin{cases} x_1' \oplus x_2' \oplus x_3' = x_1^0 \oplus x_2^0 \oplus x_3^0 = max_7{}^3 \\ x_1' \oplus x_2' \oplus x_3 = x_1^0 \oplus x_2^0 \oplus x_3^1 = max_6{}^3 \\ x_1' \oplus x_2' \oplus x_3' = x_1^0 \oplus x_2^1 \oplus x_3^0 = max_5{}^3 \\ x_1' \oplus x_2 \oplus x_3 = x_1^0 \oplus x_2^1 \oplus x_3^1 = max_4{}^3 \\ x_1 \oplus x_2' \oplus x_3' = x_1^1 \oplus x_2^0 \oplus x_3^0 = max_3{}^3 \\ x_1 \oplus x_2' \oplus x_3 = x_1^1 \oplus x_2^0 \oplus x_3^1 = max_2{}^3 \\ x_1 \oplus x_2 \oplus x_3' = x_1^1 \oplus x_2^1 \oplus x_3^0 = max_1{}^3 \\ x_1 \oplus x_2 \oplus x_3 = x_1^1 \oplus x_2^1 \oplus x_3^1 = max_0{}^3 \end{cases}$$

Teorema 2.10 Seja a álgebra de Boole $\langle B,\oplus,\otimes,',0,1\rangle$ e $x_1, x_2,..., x_n$ variáveis que representam elementos de B. Toda fórmula booleana $\alpha(x_1, x_2,..., x_n)$ de n variáveis é equivalente a uma expansão em minterms dada por:

$$\alpha(x_1, x_2,..., x_n) = \bigoplus_{m=0}^{2^n-1} v(q(t_m^n)) \otimes min_m{}^n$$

Exemplo 2.15 Considere $\alpha(x_1, x_2, x_3) = (x_1 \otimes x_2') \oplus (x_2' \otimes (x_3 \oplus x_1'))$. Usando o Teorema 2.10, essa fórmula é equivalente à expansão em minterms:

$$(x_1 \otimes x_2') \oplus (x_2' \otimes (x_3 \oplus x_1')) = \bigoplus_{m=0}^{2^3-1} v(q(t_m^3)) \otimes min_m^{\,3} =$$

$$= v(q(t_0^3)) \otimes min_0^{\,3} \oplus v(q(t_1^3)) \otimes min_1^{\,3} \oplus v(q(t_2^3)) \otimes min_2^{\,3} \oplus v(q(t_3^3)) \otimes min_3^{\,3} \oplus$$
$$v(q(t_4^3)) \otimes min_4^{\,3} \oplus v(q(t_5^3)) \otimes min_5^{\,3} \oplus v(q(t_6^3)) \otimes min_6^{\,3} \oplus v(q(t_7^3)) \otimes min_7^{\,3} =$$

$$= v(q(\alpha(0,0,0)) \otimes min_0^{\,3} \oplus v(q(\alpha(0,0,1))) \otimes min_1^{\,3} \oplus v(q(\alpha(0,1,0))) \otimes min_2^{\,3} \oplus$$
$$v(q(\alpha(0,1,1))) \otimes min_3^{\,3} \oplus v(q(\alpha(1,0,0))) \otimes min_4^{\,3} \oplus v(q(\alpha(1,0,1))) \otimes min_5^{\,3} \oplus$$
$$v(q(\alpha(1,1,0))) \otimes min_6^{\,3} \oplus v(q(\alpha(1,1,1))) \otimes min_7^{\,3} =$$

$$= (x_1^0 \otimes x_2^0 \otimes x_3^0) \oplus (x_1^0 \otimes x_2^0 \otimes x_3^1) \oplus (x_1^1 \otimes x_2^0 \otimes x_3^0) \oplus (x_1^1 \otimes x_2^0 \otimes x_3^1) =$$
$$= (x_1' \otimes x_2' \otimes x_3') \oplus (x_1' \otimes x_2' \otimes x_3) \oplus (x_1 \otimes x_2' \otimes x_3') \oplus (x_1 \otimes x_2' \otimes x_3).$$

Teorema 2.11 Para uma dada fórmula booleana $\alpha(x_1, x_2,..., x_n)$, a equivalente expansão em minterm, definida pelo Teorema 2.10, é única.

Teorema 2.12 Seja a álgebra de Boole $\langle B,\oplus,\otimes,',0,1\rangle$ e $x_1, x_2,..., x_n$ variáveis que representam elementos de B. Toda fórmula booleana $\alpha(x_1, x_2,..., x_n)$ de n variáveis é equivalente a uma expansão em maxterms dada por:

$$\alpha(x_1, x_2, ..., x_n) = \bigotimes_{m=0}^{2^n-1} v(q(t_m^n)) \oplus max_m^{\,n}$$

Observação 2.5

1. Uma vez que as fórmulas normais expandidas são únicas, elas são chamadas de fórmulas normais ou fórmulas canônicas.
2. A expansão canônica do Teorema 2.10 é chamada de *forma normal disjuntiva.*
3. A expansão canônica do Teorema 2.12 é chamada de *forma normal conjuntiva.*
4. Uma fórmula em que os literais são combinados apenas via operador $\otimes$ é chamada de *produto simples* ou *produto*. Se o operador for $\oplus$, a fórmula é chamada de *soma simples* ou *soma.*

Observação 2.6 Uma conseqüência óbvia da unicidade da forma normal é que duas fórmulas booleanas são equivalentes se elas têm a mesma forma normal. Suas funções associadas são, conseqüentemente, iguais.

Observação 2.7 Exames das expressões canônicas mostram que duas fórmulas booleanas $\alpha(x_1, x_2,..., x_n)$ e $\beta(x_1, x_2,..., x_n)$ têm a mesma forma normal, isto é, têm a mesma função associada, se $v(\alpha(x_1, x_2,..., x_n)) = v(\beta(x_1, x_2,..., x_n))$ para todas as n-amostras $<m_1,m_2,...,m_n>$ do conjunto $\{0,1\}$.

Observação 2.8

1. Na forma normal disjuntiva, o termo $min_m{}^n$ para o qual o correspondente $v(q(t_m^n))$ é 0 não precisa ser incluído.
2. Na forma normal conjuntiva, o termo $max_m{}^n$ para o qual o correspondente $v(q(t_m^n))$ é 1 não precisa ser incluído.

Exemplo 2.16 Considere a fórmula booleana $\alpha(x_1, x_2, x_3) = (x_1 \oplus x_2) \otimes (x_1 \oplus x_3')$ e ache as correspondentes expansões em minterms e maxterms.

Usando o Teorema 2.10, essa fórmula é equivalente à expansão em minterms:

$$(x_1 \oplus x_2') \otimes (x_1 \oplus x_3') = \bigoplus_{m=0}^{2^3-1} v(q(t_m^3)) \otimes min_m{}^3 =$$

$$= v(q(t_0^3)) \otimes min_0{}^3 \oplus v(q(t_1^3)) \otimes min_1{}^3 \oplus v(q(t_2^3)) \otimes min_2{}^3 \oplus v(q(t_3^3)) \otimes min_3{}^3 \oplus$$
$$v(q(t_4^3)) \otimes min_4{}^3 \oplus v(q(t_5^3)) \otimes min_5{}^3 \oplus v(q(t_6^3)) \otimes min_6{}^3 \oplus v(q(t_7^3)) \otimes min_7{}^3 =$$

$$= v(\alpha(0,0,0)) \otimes min_0{}^3 \oplus v(\alpha(0,0,1)) \otimes min_1{}^3 \oplus v(\alpha(0,1,0)) \otimes min_2{}^3 \oplus v(\alpha(0,1,1)) \otimes$$
$$min_3{}^3 \oplus v(\alpha(1,0,0)) \otimes min_4{}^3 \oplus v(\alpha(1,0,1)) \otimes min_5{}^3 \oplus v(\alpha(1,1,0)) \otimes min_6{}^3 \oplus$$
$$v(\alpha(1,1,1)) \otimes min_7{}^3.$$

A Tabela 2.9 lista os valores $v(q(t_m^3))$, os quais, uma vez obtidos, serão substituídos na expressão anterior.

Tabela 2.9 Cálculo para a expansão da fórmula booleana $(x_1 \oplus x_2) \otimes (x_1 \oplus x_3')$.

t_m^3	x_1	x_2	x_3	x_3'	$x_1 \oplus x_2$	$x_1 \oplus x_3'$	$(x_1 \oplus x_2) \otimes (x_1 \oplus x_3')$	$v(q(t_m^3))$
t_0^3	0	0	0	1	0	1	0	0←
t_1^3	0	0	1	0	0	0	0	0←
t_2^3	0	1	0	1	1	1	1	1
t_3^3	0	1	1	0	1	0	0	0←

Tabela 2.9 *Continuação...*

t_4^3	1	0	0	1	1	1	1	1
t_5^3	1	0	1	0	1	1	1	1
t_6^3	1	1	0	1	1	1	1	1
t_7^3	1	1	1	0	1	1	1	1

Os minterms min_0^3, min_1^3 e min_3^3 não entram na expansão normal disjuntiva. Vem, pois, que:

$(x_1 \oplus x_2) \otimes (x_1 \oplus x_3') = min_2^3 \oplus min_4^3 \oplus min_5^3 \oplus min_6^3 \oplus min_7^3 =$

$= (x_1' \otimes x_2 \otimes x_3') \oplus (x_1 \otimes x_2' \otimes x_3') \oplus (x_1 \otimes x_2' \otimes x_3) \oplus (x_1 \otimes x_2 \otimes x_3') \oplus (x_1 \otimes x_2 \otimes x_3)$.

Já a expansão em maxterms fica:

$(x_1 \oplus x_2) \otimes (x_1 \oplus x_3') = max_0^3 \otimes max_1^3 \otimes max_3^3 = (x_1 \oplus x_2 \oplus x_3) \otimes (x_1 \oplus x_2 \oplus x_3') \otimes$
$(x_1 \oplus x_2' \oplus x_3')$.

Exemplo 2.17 Em algumas situações pode ser necessário obter a fórmula associada a uma dada função booleana. Considere, por exemplo, a função booleana $f: \{0,1\}^2 \rightarrow \{0,1\}$ dada por: $f = \{<0,0,0>, <0,1,1><1,0,1><1,1,0>\}$. Portanto, considerando que:

$$v(\alpha(0,0)) = 0 \quad v(t_0^2) = 0$$

$$v(\alpha(0,1)) = 1 \quad v(t_1^2) = 1$$

$$v(\alpha(1,0)) = 1 \quad v(t_2^2) = 1$$

$$v(\alpha(1,1)) = 0 \quad v(t_3^2) = 0$$

tem-se:

$\alpha(x_1,x_2) = v(t_0^2) \otimes min_0^2 \oplus v(t_1^2) \otimes min_1^2 \oplus v(t_2^2) \otimes min_2^2 \oplus v(t_3^2) \otimes min_3^2 = min_1^2 \oplus$
$min_2^2 = (x_1' \otimes x_2) \oplus (x_1 \otimes x_2')$ e que

$\alpha(x_1,x_2) = v(t_0^2) \oplus max_0^2 \otimes v(t_1^2) \oplus max_1^2 \otimes v(t_2^2) \oplus max_2^2 \otimes v(t_3^2) \oplus max_3^2 = max_0^2 \oplus$
$max_3^2 = (x_1 \oplus x_2) \otimes (x_1' \oplus x_2')$.

Exemplo 2.18 As Tabelas 2.10 e 2.11 mostram que as duas fórmulas booleanas:

$$\alpha = ((x_1 \otimes x_3) \oplus (x_2 \oplus x_3'))'$$

$$\beta = (x_1' \otimes x_3) \oplus (x_2' \otimes x_3')$$

são equivalentes.

A forma normal disjuntiva de α e β é: $(x_1' \otimes x_2' \otimes x_3') \oplus (x_1' \otimes x_2' \otimes x_3) \oplus (x_1' \otimes x_2 \otimes x_3) \oplus (x_1 \otimes x_2' \otimes x_3')$.

A forma normal conjuntiva de α e β é: $(x_1 \oplus x_2' \oplus x_3) \otimes (x_1' \oplus x_2 \oplus x_3') \otimes (x_1' \oplus x_2' \oplus x_3) \otimes (x_1' \oplus x_2' \oplus x_3')$.

Tabela 2.10 $\alpha = ((x_1 \otimes x_3) \oplus (x_2 \oplus x_3'))' = (\alpha_1 \oplus \alpha_2)'$.

m	t^3_m	$v(\alpha_1)$	$v(\alpha_2)$	$v(\alpha_1 \oplus \alpha_2)$	$v(\alpha)$
0	<0,0,0>	0	0	0	1
1	<0,0,1>	0	0	0	1
2	<0,1,0>	0	1	1	0
3	<0,1,1>	0	0	0	1
4	<1,0,0>	0	0	0	1
5	<1,0,1>	1	0	1	0
6	<1,1,0>	0	1	1	0
7	<1,1,1>	1	0	1	0

Tabela 2.11 $\beta = (x_1' \otimes x_3) \oplus (x_2' \otimes x_3')$.

m	t^3_m	$v(x_1' \otimes x_3)$	$v(x_2' \otimes x_3')$	$v(\beta)$
0	<0,0,0>	0	1	1
1	<0,0,1>	1	0	1
2	<0,1,0>	0	0	0
3	<0,1,1>	1	0	1
4	<1,0,0>	0	1	1
5	<1,0,1>	0	0	0
6	<1,1,0>	0	0	0
7	<1,1,1>	0	0	0

2.4 MINIMIZAÇÃO DE EXPRESSÕES BOOLEANAS

Uma vez que uma fórmula booleana pode ser representada por diferentes expressões, em algumas aplicações (por exemplo, em projetos de circuitos) é importante usar uma representação que de alguma forma seja minimal, particularmente se tal expressão será usada várias vezes.

Minimização implica redução de uma expressão booleana a alguma forma minimal. Qualquer estratégia para minimizar expressões booleanas tem por base resultados da Álgebra de Boole. Geralmente a redução algébrica de funções booleanas não é um processo trivial e exige certo grau de experiência; isso se torna mais evidente à medida que a complexidade da função a ser minimizada aumenta. O termo minimal diz respeito a uma determinada forma de representação – no contexto em questão, trata-se da forma normal disjuntiva. Os métodos de minimização são algoritmos de simplificação que minimizam uma forma normal de maneira peculiar. Dada uma forma normal disjuntiva, o método de minimização encontra uma expressão equivalente, representada como uma soma de produtos, que contém o menor número de produtos e, de todas as somas de produto contendo esse número de produtos, aquela em que os produtos têm o menor número de literais. Existem várias técnicas que viabilizam e facilitam o processo de minimização. Particularmente conhecidas são o algoritmo de Quine-McCluskey e os mapas de Karnaugh; ambos os métodos têm a mesma funcionalidade e estão fundamentados no conceito de implicativo direto. O algoritmo de Quine-McCluskey, entretanto, é escalonável e facilmente automatizado. O mapa de Karnaugh é construído a partir da forma normal que deve ser minimizada, e a fórmula minimal é encontrada por meio do exame visual do mapa. Essa técnica é complicada de ser implementada computacionalmente e, como um método manual, ela se torna complicada quando o número de variáveis excede a 5. Nesta seção, serão apresentados e discutidos ambos, o Algoritmo de Quine-McCluskey e os mapas de Karnaugh, com ênfase no primeiro método.

2.4.1 Considerações iniciais

Considere uma fórmula booleana com três variáveis (n=3), $\alpha(x_1, x_2, x_3)$, cuja avaliação está mostrada na Tabela 2.12.

Tabela 2.12 Avaliação de uma fórmula booleana com três variáveis.

	x_1	x_2	x_3	$v(q(t_m^n))$
t_0^3	0	0	0	1
t_1^3	0	0	1	0
t_2^3	0	1	0	0
t_3^3	0	1	1	0
t_4^3	1	0	0	0
t_5^3	1	0	1	0
t_6^3	1	1	0	0
t_7^3	1	1	1	0

Como estabelecido pelo Teorema 2.10, a fórmula avaliada na Tabela 2.12 é equivalente a uma expansão em minterms dada por:

$$\alpha(x_1, x_2, x_3) = \bigoplus_{m=0}^{2^n-1} v(q(t_m^n)) \otimes min_m^n$$

que, no caso em questão, produz:

$\alpha(x_1, x_2, x_3) = 1 \otimes min_0^3 \oplus 0 \otimes min_1^3 \oplus 0 \otimes min_2^3 \oplus 0 \otimes min_3^3 \oplus 0 \otimes min_4^3 \oplus 0 \otimes min_5^3 \oplus 0 \otimes min_6^3 \oplus 0 \otimes min_7^3 = min_0^3 = x_1^0 x_2^0 x_3^0 = x_1' x_2' x_3'$.

Considere, agora, a expansão em minterms de uma fórmula booleana $\beta(x_1, x_2, x_3)$, cuja avaliação está mostrada na Tabela 2.13.

Tabela 2.13 Avaliação da fórmula booleana $\beta(x_1, x_2, x_3)$.

	x_1	x_2	x_3	$v(q(t_m^n))$
t_0^3	0	0	0	0
t_1^3	0	0	1	0
t_2^3	0	1	0	0
t_3^3	0	1	1	0
t_4^3	1	0	0	1
t_5^3	1	0	1	1
t_6^3	1	1	0	0
t_7^3	1	1	1	0

Tem-se que $\beta(x_1, x_2, x_3) = \bigoplus_{m=0}^{2^n-1} v(q(t_m^n)) \otimes min_m^n = 0 \otimes min_0^3 \oplus 0 \otimes min_1^3 \oplus 0 \otimes min_2^3 \oplus 0 \otimes min_3^3 \oplus 1 \otimes min_4^3 \oplus 1 \otimes min_5^3 \oplus 0 \otimes min_6^3 \oplus 0 \otimes min_7^3 = min_4^3 \oplus min_5^3 = x_1^1 x_2^0 x_3^0 \oplus x_1^1 x_2^0 x_3^1 = x_1 x_2' x_3' \oplus x_1 x_2' x_3 = x_1 x_2'$ (ver Observação 2.9).

Observação 2.9 Note que:

(a) na expressão de um minterm, o operador $\otimes$ está implícito;

(b) na fórmula $x_1 x_2' x_3' \oplus x_1 x_2' x_3$, a variável x_3 é redundante e, portanto, pode ser eliminada. Segue a justificativa.

$$x_1 x_2' x_3' \oplus x_1 x_2' x_3 = (\underbrace{x_1 \otimes x_2'}_{a} \otimes \underbrace{x_3'}_{b}) \oplus (\underbrace{x_1 \otimes x_2'}_{a} \otimes \underbrace{x_3}_{c})$$

O Axioma 3B estabelece que: $(a \otimes b) \oplus (a \otimes c) = a \otimes (b \oplus c)$. Para o exemplo em questão, pode-se escrever que a expressão anterior é igual a: $a \otimes (b \oplus c) = a \otimes (x_3' \oplus x_3) = a \otimes 1 = a = x_1 \otimes x_2'$.

Para qualquer fórmula booleana α,

$$(\alpha \otimes x) \oplus (\alpha \otimes x') = \alpha \otimes (x \oplus x') = \alpha \otimes 1 = \alpha \tag{2.1}$$

O uso do resultado (2.1), portanto, fornece uma maneira de redução de 2 termos a 1 termo apenas – como visto no exemplo em questão, a variável x_3 é redundante. Essa abordagem pode ser usada na minimização de fórmulas booleanas.

Considere uma terceira fórmula booleana $\lambda(x_1, x_2, x_3)$ cuja avaliação está mostrada na Tabela 2.14.

Tabela 2.14 Avaliação da fórmula booleana $\lambda(x_1, x_2, x_3)$.

	x_1	x_2	x_3	$v(q(t_m^n))$
t_0^3	0	0	0	0
t_1^3	0	0	1	1
t_2^3	0	1	0	0
t_3^3	0	1	1	1
t_4^3	1	0	0	1
t_5^3	1	0	1	1
t_6^3	1	1	0	0
t_7^3	1	1	1	0

Tem-se, então, que $\lambda(x_1, x_2, x_3) = \bigoplus_{m=0}^{2^n-1} v(q(t_0^3)) \otimes \min_m{}^n = 0 \otimes \min_0{}^3 \oplus 1 \otimes \min_1{}^3 \oplus 0 \otimes \min_2{}^3 \oplus$
$1 \otimes \min_3{}^3 \oplus 1 \otimes \min_4{}^3 \oplus 1 \otimes \min_5{}^3 \oplus 0 \otimes \min_6{}^3 \oplus 0 \otimes \min_7{}^3 = \min_1{}^3 \oplus \min_3{}^3 \oplus \min_4{}^3 \oplus \min_5{}^3$
$= x_1^0 x_2^0 x_3^1 \oplus x_1^0 x_2^1 x_3^1 \oplus x_1^1 x_2^0 x_3^0 \oplus x_1^1 x_2^0 x_3^1 = x_1' x_2' x_3 \oplus x_1' x_2 x_3 \oplus x_1 x_2' x_3' \oplus x_1 x_2' x_3 = x_1' x_3 \oplus x_1 x_2'$
(ver Observação 2.9 (b)).

Definição 2.11

1. Uma fórmula booleana α *cobre* outra fórmula booleana β se $v(\alpha) = 1$ para todos os valores de atribuição que tornam $v(\beta) = 1$.
2. Se α cobre β e β for um produto simples, então β é um *implicativo* de α.
3. Se β for um implicativo de α e nenhuma fórmula obtida de β pela remoção de literais for um implicativo de α, então β é um *implicativo direto* de α.

Exemplo 2.19 Considere a fórmula α: $x_1x_2 \oplus x_3x_4$ e a fórmula β: $x_1x_2x_3'$. Como pode ser visto na Tabela 2.15, α cobre β e, como β é um produto, β é um implicativo de α. Uma vez que a expressão x_1x_2 é também um implicativo de α, β não é um implicativo direto. Os produtos x_1x_2 e x_3x_4 são os dois implicativos diretos de α.

Tabela 2.15 Cálculo da fórmula booleana α: $x_1x_2 \oplus x_3x_4$, β: $x_1x_2x_3'$.

$<x_1,x_2,x_3,x_4>$	x_3'	x_1x_2	x_3x_4	α	β
<0,0,0,0>	1	0	0	0	0
<0,0,0,1>	1	0	0	0	0
<0,0,1,0>	0	0	0	0	0
<0,0,1,1>	0	0	1	1	0
<0,1,0,0>	1	0	0	0	0
<0,1,0,1>	1	0	0	0	0
<0,1,1,0>	0	0	0	0	0
<0,1,1,1>	0	0	1	1	0
<1,0,0,0>	1	0	0	0	0
<1,0,0,1>	1	0	0	0	0
<1,0,1,0>	0	0	0	0	0
<1,0,1,1>	0	0	1	1	0
<1,1,0,0>	1	1	0	1	1
<1,1,0,1>	1	1	0	1	1
<1,1,1,0>	0	1	0	1	0
<1,1,1,1>	0	1	1	1	0

Exemplo 2.20 Considere a fórmula α: $x_1x_2 \oplus x_1x_2'x_3 \oplus x_1'x_2'x_3$. Como pode ser visto na Tabela 2.16, os implicativos diretos de α são os produtos: x_1x_2, x_1x_3, $x_2'x_3$.

Tabela 2.16 Implicativos diretos da fórmula α: $x_1x_2 \oplus x_1x_2'x_3 \oplus x_1'x_2'x_3$.

$<x_1,x_2,x_3>$	x_1'	x_2'	x_1x_2	$x_1x_2'x_3$	$x_1'x_2'x_3$	α	x_1x_3	$x_2'x_3$
<0,0,0>	0	1	0	0	0	0	0	0
<0,0,1>	0	1	0	0	0	1	0	1
<0,1,0>	0	0	0	0	0	0	0	0
<0,1,1>	0	0	0	0	0	0	0	0
<1,0,0>	1	1	0	0	0	0	0	0
<1,0,1>	1	1	0	1	1	1	1	1
<1,1,0>	0	0	1	0	0	1	0	0
<1,1,1>	0	0	1	0	0	1	1	0

2.4.2 O Algoritmo de Quine-McCluskey

Antes da apresentação e discussão do algoritmo, considere o Exemplo 2.21.

Exemplo 2.21 Seja uma forma normal definida pelo seguinte conjunto de minterms:

$$P_1 = \{\mathbf{x_1'x_2'x_3'x_4}, x_1'x_2x_3x_4', x_1x_2'x_3x_4', x_1x_2'x_3x_4, x_1x_2x_3x_4', x_1x_2x_3x_4\}$$

O resultado (2.1) é aplicável aos seguintes pares de minterms:

$$x_1'x_2x_3x_4',\quad x_1x_2x_3x_4';$$

$$x_1x_2'x_3x_4',\quad x_1x_2'x_3x_4;$$

$$x_1x_2'x_3x_4',\quad x_1x_2x_3x_4';$$

$$x_1x_2'x_3x_4,\quad x_1x_2x_3x_4;$$

$$x_1x_2x_3x_4',\quad x_1x_2x_3x_4.$$

Note que o primeiro minterm de P_1 foi marcado em negrito para indicar que não se qualificou para participar de qualquer aplicação do resultado estabelecido em (2.1). A aplicação de (2.1) aos pares identificados produz o conjunto de produtos:

$$P_2 = \{\mathbf{x_2x_3x_4'}, x_1x_2'x_3, x_1x_3x_4', x_1x_3x_4, x_1x_2x_3\}$$

Considerando o conjunto P_2, o resultado (2.1) é agora aplicável aos pares:

$$x_1x_2'x_3,\quad x_1x_2x_3;$$

$$x_1x_3x_4',\quad x_1x_3x_4.$$

Note que o primeiro minterm de P_2 foi marcado em negrito para indicar que não se qualificou para participar de qualquer aplicação de (2.1). Para ambos os pares de P_2, o produto reduzido por (2.1) é:

$$P_3 = \{\mathbf{x_1x_3}\}$$

Em P_1 e P_2 todos os produtos, exceto $x_1'x_2'x_3'x_4$ e $x_2x_3x_4'$, foram usados. Portanto, o conjunto dos implicativos diretos é:

$$\{x_1'x_2'x_3'x_4, x_2x_3x_4', x_1x_3\}$$

Observação 2.10 Note que o teorema de redução é aplicável a um par de produtos apenas se um dos produtos contém x_i e o outro, um x_i', e todos os outros literais são iguais. Essa observação é utilizada no algoritmo de Quine-McCluskey, descrito no Procedimento 2.1, que é uma sistematização do procedimento mostrado no Exemplo 2.21.

Procedimento 2.1 Algoritmo de Quine-McCluskey.

1. Representar todo minterm $x_1^{m_1} x_2^{m_2} \dots x_n^{m_n}$ da forma normal disjuntiva pelo número binário $m_1m_2\dots m_n$.
2. $i \leftarrow n$. Dividir o conjunto dos números binários em grupos: G_0^i, G_1^i, ..., G_n^i, atribuindo o número contendo k 1s ao grupo G_k^i.
3. Comparar todo número no grupo G_k^i com todo número no grupo G_{k+1}^i, para k = 0, 1,..., n – 1. Aplicar o resultado $\alpha x \oplus \alpha x' = \alpha$ sempre que possível, isto é, sempre que os dígitos comparados forem idênticos, exceto por um único dígito, que ocupa uma determinada posição. Quando o resultado for aplicável:
   - 3.1 marcar ambos os números comparados;
   - 3.2 gerar um novo número pegando o número em G_k^i e substituindo o zero na posição em que este difere do outro por um hífen;
   - 3.3 colocar o novo número formado no grupo G_k^{i-1} caso esse grupo ainda não contenha o número em questão.
4. $i \leftarrow i - 1$. Se mais do que um grupo não for vazio, retorne ao Passo 3.
5. Pare. Os números não marcados em todos os grupos representam o conjunto dos implicativos diretos.

Exemplo 2.22 Aplicação do algoritmo Quine-McCluskey à forma normal definida pelo conjunto de minterms min_1^4, min_6^4, min_{10}^4, min_{11}^4, min_{14}^4, min_{15}^4, cujas representações binárias são, respectivamente: 0001, 0110, 1010, 1011, 1110, 1111.

A Tabela 2.17 mostra o funcionamento do algoritmo. Os grupos da primeira coluna são G_1^4, G_2^4, G_3^4 e G_4^4. O algoritmo começa comparando o 0001 de G_1^4 com o 0110 de G_2^4. Como esses números diferem em três posições, o resultado (2.1) não se aplica. A seguir, são comparados 0001 com 1010. Novamente, o (2.1) não se aplica, e as possíveis comparações entre os dois primeiros grupos terminam.

A seguir, são realizadas as comparações entre os números dos grupos G_2^4 e G_3^4. O resultado (2.1) não se aplica ao par 0110 e 1011, mas se aplica ao par 0110 e 1110, uma vez que diferem apenas no primeiro dígito. Ambos os números, 0110 e 1011, são marcados (*flag* passa a ser 1), e o número 0110 é alterado, gerando o novo número _110, que é introduzido na segunda coluna de números da tabela, como parte do grupo G_2^3.

A seguir, 1010 é comparado com 1011 e 1110 na seqüência. O resultado (2.1) é aplicável em ambas as comparações; ambos os números são marcados, e os números 101_ e 1_10 são introduzidos na segunda coluna de números da tabela. As comparações entre os elementos de G_2^4 e de G_3^4 terminaram, e os números _110, 101_ e 1_10, criados a partir das comparações bem-sucedidas, formam o grupo G_2^3. Os outros dois números na segunda coluna de números, que definem G_3^3, são os resultados das comparações de 1011 com 1111 e de 1110 com 1111, respectivamente.

O Passo 4 é então executado – como existem dois grupos de números na segunda coluna de números da tabela, o Passo 3 volta a ser realizado. Entre as seis possíveis comparações dos três números em G_2^3 com os dois números de G_3^3, o resultado (2.1) é aplicável aos pares:

1. 101_ e 111_, que provoca a marcação de ambos os números e a criação do número 1_1_ na terceira coluna de números da tabela (grupo G_2^2) e
2. 1_10 e 1_11, que provoca a marcação de ambos os números, mas não a inclusão de 1_1_ na terceira coluna de números da tabela (grupo G_2^2), uma vez que esse padrão já se encontra lá.

Uma vez que a terceira coluna contém apenas um grupo G_2^2, o algoritmo pára. Os números não marcados são: 0001 em G_1^4, _110 em G_2^3 e 1_1_ em G_2^2. Os implicativos diretos representados por esses números são $x_1'x_2'x_3'x_4$, $x_2x_3x_4'$ e x_1x_3 respectivamente.

Tabela 2.17 *Trace* do Quine-McCluskey para uma fórmula representada pelos minterms $min_1{}^4$, $min_6{}^4$, $min_{10}{}^4$, $min_{11}{}^4$, $min_{14}{}^4$, $min_{15}{}^4$.

Grupos (i = 4)		*Flag*	Grupos (i = 3)		*Flag*	Grupos (i = 2)		*Flag*
G_1^4	0001							
G_2^4	0110 1010	1 1	G_2^3	_110 101_ 1_10	 1 1	G_2^2	1_1_	
G_3^4	1011 1110	1 1	G_3^3	1_11 111_	1 1			
G_4^4	1111	1						

Exemplo 2.23 Na Tabela 2.18 está o *trace* da aplicação do algoritmo Quine-McCluskey à forma normal definida pelo conjunto de 16 minterms: $min_0{}^5$, $min_1{}^5$, $min_2{}^5$, $min_8{}^5$, $min_9{}^5$, $min_{16}{}^5$, $min_{17}{}^5$, $min_{18}{}^5$, $min_{20}{}^5$, $min_{21}{}^5$, $min_{24}{}^5$, $min_{25}{}^5$, $min_{26}{}^5$, $min_{27}{}^5$, $min_{28}{}^5$, $min_{29}{}^5$.

Tabela 2.18 *Trace* do Quine-McCluskey para uma fórmula representada por 16 minterms.

Grupos (i = 5)		*Flag*	Grupos (i = 4)		*Flag*	Grupos (i = 3)		*Flag*	Grupos (i = 2)		*Flag*
G_0^5	00000	1	G_0^4	0000_	1	G_0^3	0_00_	1	G_0^2	__00_	
				000_0	1		_000_	1			
				0_000	1		_00_0				
				_0000	1		__000	1			
G_1^5	00001	1	G_1^4	0_001	1	G_1^3	__001	1	G_1^2	1__0_	
	00010	1		_0001	1		_100_	1			
	01000	1		_0010	1		10_0_	1			
	10000	1		0100_	1		1_00_	1			
				_1000	1		1_0_0				
				1000_	1		1__00	1			
				100_0	1						
				10_00	1						
				1_000	1						
G_2^5	01001	1	G_2^4	_1001	1	G_2^3	1__01	1			
	10001	1		10_01	1		1_10_	1			
	10010	1		1_001	1		110__				
	10100	1		1_010	1		11_0_	1			
	11000	1		1010_	1						
				1_100	1						
				1100_	1						
				110_0	1						
				11_00	1						
G_3^5	10101	1	G_3^4	1_101	1						
	11001	1		110_1	1						
	11010	1		11_01	1						
	11100	1		1101_	1						
				1110_	1						

Tabela 2.18 *Continuação...*

Grupos (i = 5)		*Flag*	Grupos (i = 4)		*Flag*	Grupos (i = 3)		*Flag*	Grupos (i = 2)		*Flag*
G_4^5	11011 11101	1 1									

Após os implicativos diretos terem sido encontrados, é gerada a soma minimal dos produtos, via construção da tabela de cobertura. Constrói-se uma tabela em que as linhas correspondem aos minterms da forma normal que foi entrada para o Quine-McCluskey e as colunas, aos implicativos diretos encontrados. Tanto as linhas quanto as colunas são nomeadas usando representação binária.

Para o preenchimento de determinadas posições da tabela, o seguinte procedimento é adotado: considere um determinado minterm (ocupando linha i) e um determinado implicativo direto (ocupando coluna j). Sem levar em consideração os hífens da representação do implicativo direto, comparar os dígitos restantes com os correspondentes dígitos na representação do minterm. Se forem os mesmos, então o implicativo direto cobre o minterm e o número 1 é inserido na tabela, na posição [i,j]. Esse procedimento é repetido para todos os pares minterms–implicativos diretos. O resultado, para o Exemplo 2.23 (Tabela 2.18), está mostrado na Tabela 2.19.

Note que, se α cobre β e se β cobre α, então α e β são equivalentes. A forma normal cobre todo implicativo direto e, portanto, toda soma de implicativos diretos. Conseqüentemente, uma soma minimal de implicativos diretos que juntos cobrem todos os minterms caracteriza uma soma minimal de produtos equivalente à forma normal. Esse conjunto minimal é determinado a partir da tabela de cobertura, por meio da seleção do menor número de colunas, de maneira que, considerando as colunas selecionadas, exista pelo menos um 1 em toda linha, e o número de literais nos implicativos diretos das colunas selecionadas é o menor possível para este número de colunas.

Se uma linha contém apenas um único 1, então o implicativo associado à coluna correspondente é chamado de *implicativo direto essencial*. Todo implicativo direto essencial deve comparecer no conjunto minimal. Na Tabela 2.19 quatro implicativos diretos são essenciais: _00_0 (único 1 na linha 3), 110_ _ (único 1 na linha 15), _ _00_ (único 1 na linha 2 e também nas linhas 4 e 6) e 1_ _0_ (único 1 nas linhas 9, 11, 14 e 16). As correspondentes colunas são 1, 3, 4 e 5 respectivamente. Como os quatro implicativos diretos essenciais cobrem toda a forma normal, $x_2'x_3'x_5' \oplus x_1x_2x_3' \oplus x_3'x_4' \oplus x_1x_4'$ é a soma minimal de produtos equivalente à forma normal.

Tabela 2.19 Tabela de cobertura para o Exemplo 2.23 com os implicativos diretos essenciais assinalados com a flecha.

			↓		↓	↓	↓
			_00_0	1_0_0	110__	__00_	1__0_
	1	00000	1			1	
	2	00001				**1**	
	3	00010	**1**				
M	4	01000				**1**	
I	5	10000	1	1		1	1
N	6	01001				**1**	
T	7	10001				1	1
E	8	10010	1	1			
R	9	10100					**1**
M	10	11000		1	1	1	1
S	11	10101					**1**
	12	11001			1	1	1
	13	11010		1	1		
	14	11100					**1**
	15	11011			**1**		
	16	11101					**1**

Geralmente, a determinação da soma minimal de produtos não é simples. Alguns minterms podem não ser cobertos por qualquer dos implicativos diretos essenciais. A forma minimal, então, deve conter pelo menos um implicativo direto adicional aos implicativos diretos essenciais.

Pode também acontecer de nenhum implicativo direto ser essencial (como mostra a Tabela 2.20); nesse caso, o problema de minimização tem, em geral, mais de uma solução. Nos casos mais simples a solução pode ser encontrada por inspeção. É fácil ver na Tabela 2.20 que $x_1'x_2'x_3' \oplus x_1x_2'x_4' \oplus x_1'x_2x_4 \oplus x_1x_2x_3$ é uma possível solução do problema.

Tabela 2.20 Tabela de cobertura com ausência de implicativos diretos essenciais.

			000_	_000	0_01	10_0	01_1	1_10	_111	111_
M	1	0000	1	1						
I	2	0001	1		1					
N	3	1000		1		1				
T	4	0101			1		1			
E	5	1010				1		1		
R	6	0111					1		1	
M	7	1110						1		1
S	8	1111							1	1

Exemplo 2.24 Considere uma fórmula booleana definida pelo conjunto de minterms min_1^4, min_4^4, min_6^4, min_9^4, min_{11}^4, min_{12}^4, min_{13}^4, min_{15}^4. A Tabela 2.21 mostra o uso do algoritmo de Quine-McCluskey, cujas representações binárias dos minterms são, respectivamente: 0001, 0100, 0110, 1001, 1011, 1100, 1101 e 1111.

Tabela 2.21 *Trace* do Quine-McCluskey para a fórmula representada pelos minterms min_1^4, min_4^4, min_6^4, min_9^4, min_{11}^4, min_{12}^4, min_{13}^4, min_{15}^4.

Grupos (i = 4)		*Flag*	Grupos (i = 3)		*Flag*	Grupos (i = 2)		*Flag*
G_1^4	0001 0100	1 1	G_1^3	_001 01_0 _100				
G_2^4	0110 1001 1100	1 1 1	G_2^3	10_1 1_01 110_	1 1	G_2^2	1__1	
G_3^4	1011 1101	1 1	G_3^3	1_11 11_1	1 1			
G_4^4	1111	1						

Os implicativos diretos são as expressões que não tiveram suas "*flags*", no caso, as expressões: $x_2'x_3'x_4$, $x_1'x_2x_4'$, $x_2x_3'x_4'$, $x_1x_2x_3'$ e x_1x_4. A Tabela 2.22 é a tabela de cobertura, para evidenciar os implicativos diretos essenciais e, assim, construir as FND minimais.

Tabela 2.22 Tabela de cobertura, para evidenciar os implicativos diretos essenciais.

		_001	01_0	_100	110_	1__1
M	0001	1				
I	0100		1	1		
N	0110		1			
T	1001	1				1
E	1100			1	1	
R	1011					1
M	1101				1	1
S	1111					1

Como visto anteriormente, para o preenchimento da tabela de cobertura o procedimento é: sem levar em consideração os hífens na representação dos implicativos diretos, compare o restante dos dígitos com os correspondentes dígitos do minterm. Se os dígitos coincidirem, então o implicativo direto cobre o minterm e um 1 é colocado na intersecção da linha e coluna relativas a ambos. Esse processo é repetido para todo par de minterm–implicativo direto. Note que os essenciais são: _001 (sem sua inclusão o minterm 0001 não fica coberto), 01_0 (sem sua inclusão o minterm 0110 não fica coberto) e 1__1 (sem sua inclusão os minterms 1011 e 111 não ficam cobertos). Com a inclusão dos três essenciais, ficam cobertos todos os minterms, com exceção do 1100. Portanto, um implicativo direto precisa ser incluído para a cobertura desse minterm; existem duas opções: o _100 ou o 110_, o que dá origem a duas possíveis FND minimais: $x_2'x_3'x_4 \oplus x_2'x_3x_4' \oplus x_1x_4 \oplus x_2x_3'x_4'$ ou $x_2'x_3'x_4 \oplus x_2'x_3x_4' \oplus x_1x_4 \oplus x_1x_2x_3'$.

Exemplo 2.25 Considere uma fórmula booleana definida pelo conjunto de minterms min_0^4, min_4^4, min_5^4, min_6^4, min_7^4, min_8^4, min_9^4, min_{10}^4, min_{11}^4, min_{14}^4, min_{15}^4. A Tabela 2.23 mostra o uso do algoritmo de Quine-McCluskey, cujas representações binárias dos minterms são, respectivamente: 0000, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1110, 1111.

Tabela 2.23 *Trace* do Quine-McCluskey para a função representada pelos minterms min_0^4, min_4^4, min_5^4, min_6^4, min_7^4, min_8^4, min_9^4, min_{10}^4, min_{11}^4, min_{14}^4, min_{15}^4.

Grupos (i = 4)		*Flag*	Grupos (i = 3)		*Flag*	Grupos (i = 2)		*Flag*
G_0^4	0000	1	G_0^3	0_00 _000				
G_1^4	0100 1000	1 1	G_1^3	010_ 01_0 100_ 10_0	1 1 1 1	G_1^2	01_ _ 10_ _	
G_2^4	0101 0110 1001 1010	1 1 1 1	G_2^3	01_1 011_ _110 10_1 101_ 1_10	1 1 1 1 1 1	G_2^2	_11_ 1_1_	
G_3^4	0111 1011 1110	1 1 1	G_3^3	_111 1_11 111_	1 1 1			
G_4^4	1111	1						

Os implicativos diretos são as expressões que não tiveram suas "*flags*" 1. No caso, são as expressões $x_1'x_3'x_4'$, $x_2'x_3'x_4'$, $x_1'x_2$, x_1x_2', x_2x_3 e x_1x_3 correspondentes aos padrões: 0_00, _000; 01_ _, 10_ _, _11_ e 1_1_. A Tabela 2.24 é a tabela de cobertura, para evidenciar os implicativos diretos essenciais e, assim, construir as FND minimais.

Analisando a Tabela 2.24, note que os essenciais são: 01_ _ (sem sua inclusão o minterm 0101 não fica coberto) e o 10_ _ (sem sua inclusão o minterm 1001 não fica coberto). Com a inclusão dos dois essenciais, ficam cobertos apenas oito minterms dos 11 existentes. Existem várias possibilidades para a cobertura dos três restantes e, em razão desse fato, existem 4 possíveis fnd minimais, que são, respectivamente:

1. $x_1'x_2 \oplus x_1x_2' \oplus x_2x_3 \oplus x_1'x_3'x_4'$

2. $x_1'x_2 \oplus x_1x_2' \oplus x_2x_3 \oplus x_2'x_3'x_4'$

3. $x_1'x_2 \oplus x_1x_2' \oplus x_1x_3 \oplus x_1'x_3'x_4'$

4. $x_1'x_2 \oplus x_1x_2' \oplus x_1x_3 \oplus x_2'x_3'x_4'$

Tabela 2.24 Tabela de cobertura, para evidenciar implicativos diretos essenciais.

MINTERMS	0_00	_000	01_ _	10_ _	_11_	1_1_
0000	1	1				
0100	1		1			
1000		1		1		
0101			**1**			
0110			1		1	
1001				**1**		1
1010				1		
0111			1		1	
1011				1		1
1110					1	1
1111					1	1

2.4.3 Mapas de Karnaugh

Um mapa de Karnaugh pode ser genericamente definido como uma disposição ordenada de posições com atribuições, tais que a diferença entre quaisquer duas posições adjacentes representa uma mudança no valor de apenas uma variável. O mapa deve conter uma célula para cada possível combinação de variáveis.

Um mapa para 2 variáveis deve conter 4 células, uma vez que existem 2^2 combinações diferentes dos possíveis valores das 2 variáveis. Para 3 variáveis, 8 células e para n variáveis, 2^n células. A Figura 2.3 mostra três maneiras diferentes de definir cada célula de uma fórmula booleana com 3 variáveis (8 células), e a Figura 2.4 mostra duas das representações para uma fórmula booleana com 4 variáveis. Para simplificar a notação, em vez de as variáveis booleanas serem referenciadas como $x_1, x_2, \ldots, x_n$, como tem sido feito até esse ponto, elas serão notadas com as letras maiúsculas: A, B, C... para simplificar a construção do mapa.

		AB			
		00	01	11	10
C	0	A'B'C'	A'BC'	ABC'	AB'C'
	1	A'B'C	A'BC	ABC	AB'C

		AB			
		00	01	11	10
C	0	000	010	110	100
	1	001	011	111	101

		AB			
		00	01	11	10
C	0	0	2	6	4
	1	1	3	7	5

Figura 2.3 Três maneiras diferentes de definir um mapa de Karnaugh para uma fórmula booleana com três variáveis (A, B e C).

		AB			
		00	01	11	10
CD	00	0	4	12	8
	01	1	5	13	9
	11	3	7	15	11
	10	2	6	14	10

		AB			
		00	01	11	10
CD	00	0000	0100	1100	1000
	01	0001	0101	1101	1001
	11	0011	0111	1111	1011
	10	0010	0110	1110	1010

Figura 2.4 Duas maneiras diferentes de definir um mapa de Karnaugh para uma fórmula booleana com quatro variáveis.

Qualquer uma das representações é válida e sua adoção depende da preferência do usuário. O processo de minimização usando o mapa é fundamentado no reconhecimento de padrões básicos. A presença de 1s em células adjacentes identifica imediatamente a presença de variáveis redundantes. A Figura 2.5 ilustra alguns exemplos de minimização em um mapa com 3 variáveis. Note que o agrupamento de 2 células elimina uma variável. O mapa é construído de tal maneira que as extremidades podem ser consideradas adjacentes.

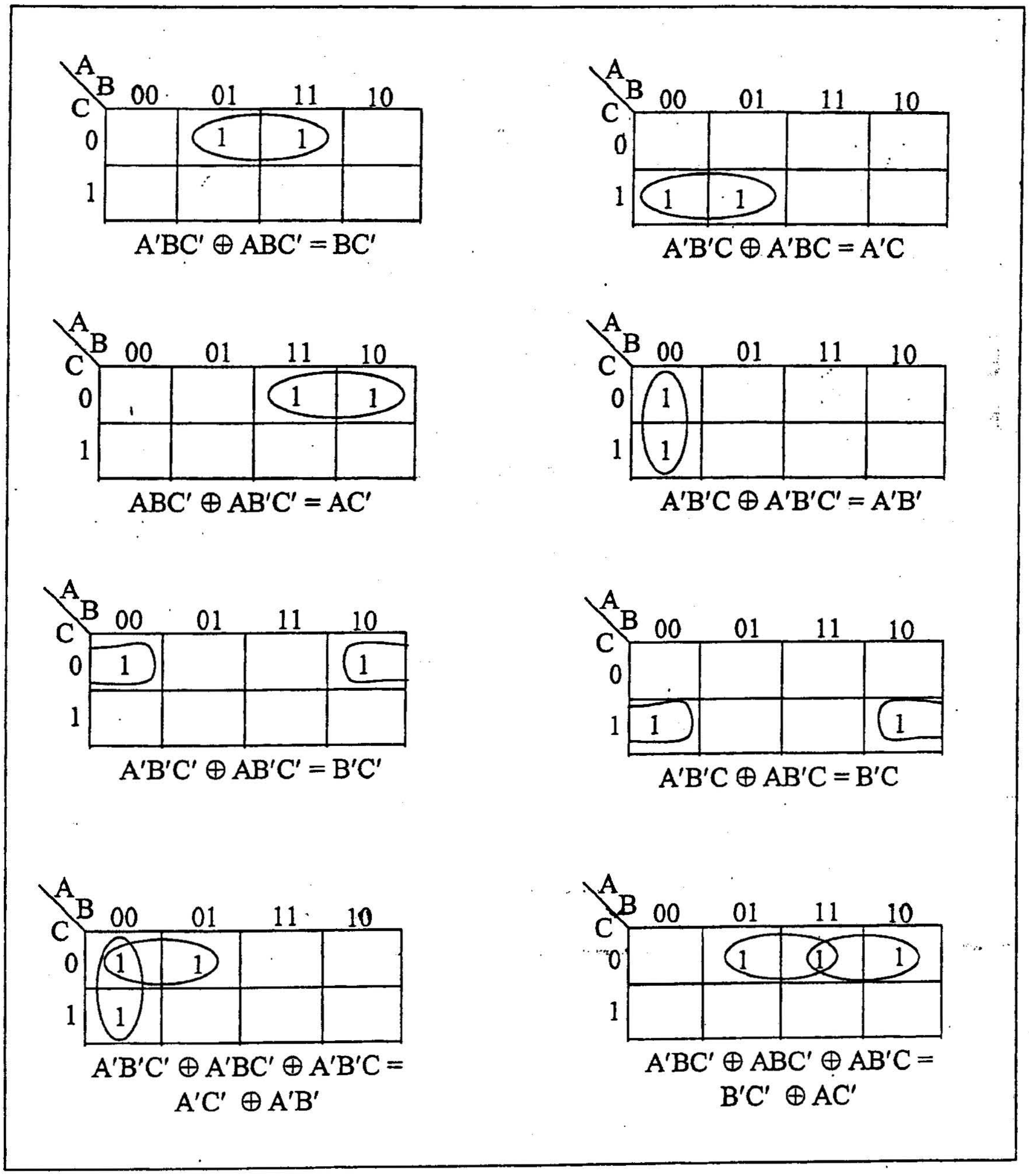


Figura 2.5 Exemplos de minimização em fórmulas booleanas com três variáveis.

A Figura 2.6 ilustra uma fórmula booleana, cuja representação ocupa apenas uma célula de um mapa com 4 variáveis. As Figuras 2.7 e 2.8 ilustram o agrupamento de duas células em um mapa com 4 variáveis, e a Figura 2.9 ilustra o agrupamento de 4 células em um mapa com 4 variáveis.

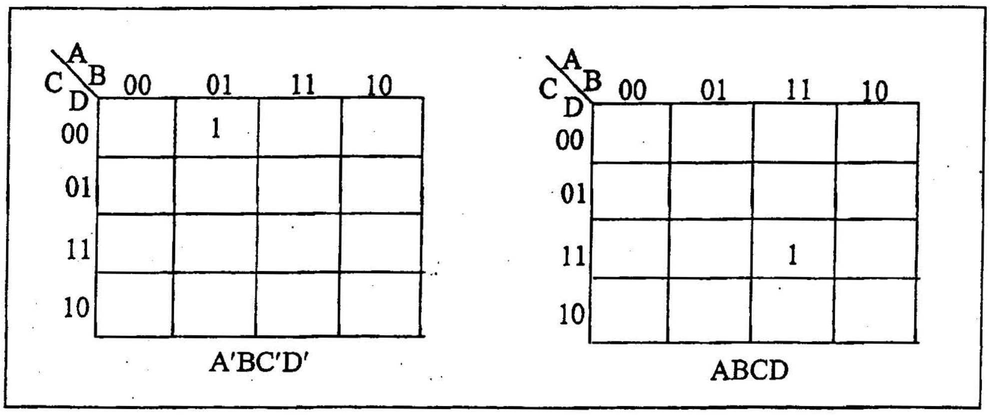


Figura 2.6 Representação de fórmula que ocupa apenas uma célula de um mapa com 4 variáveis.

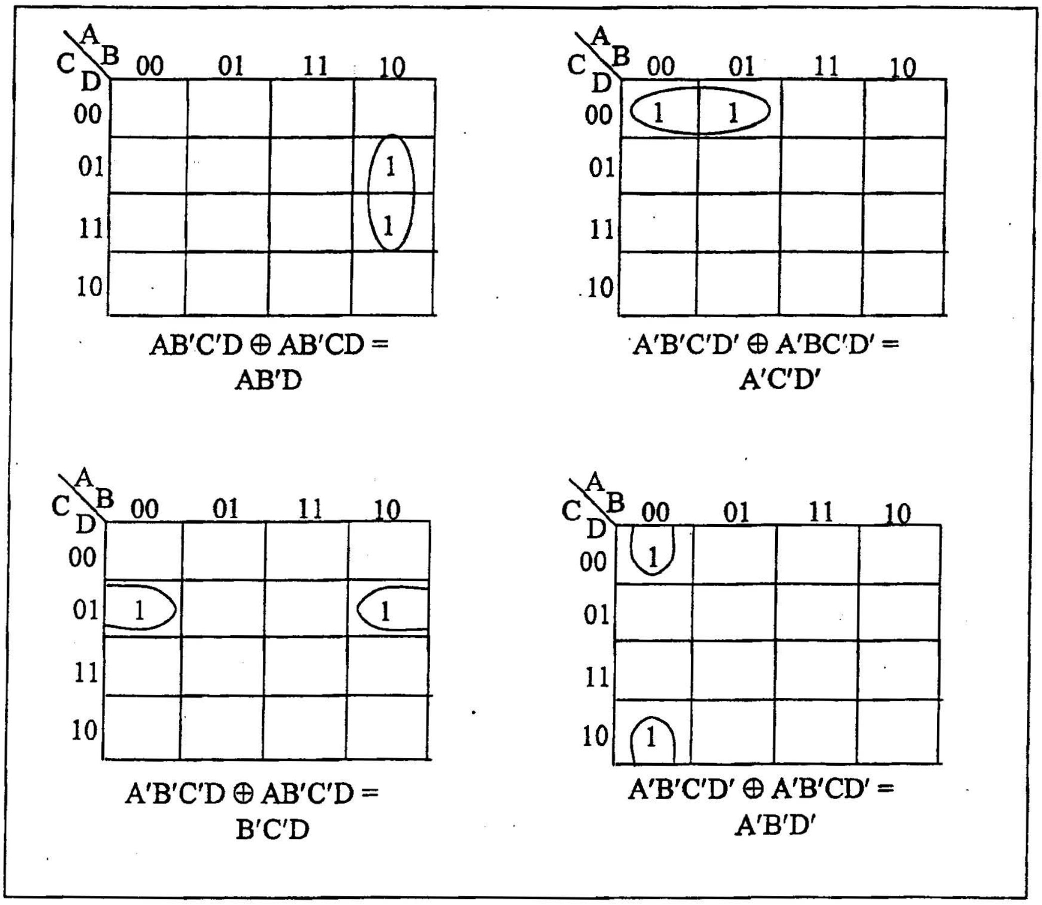
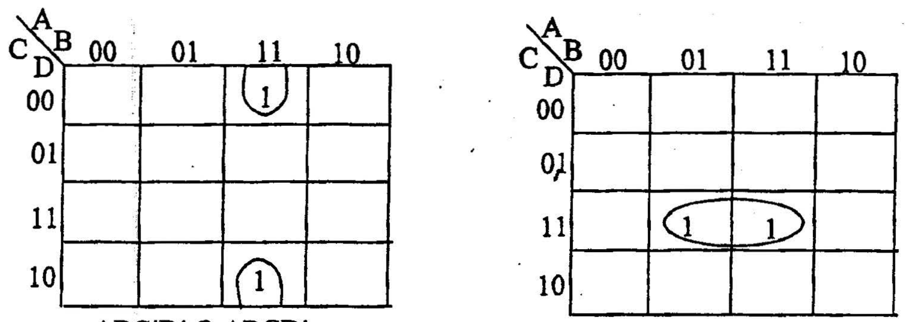


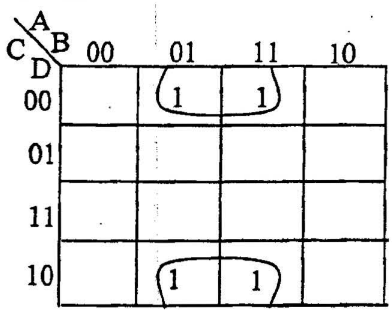
Figura 2.7 Exemplos de minimização de fórmulas que ocupam 2 células de um mapa com 4 variáveis.



ABC'D' ⊕ ABCD' =
ABD'

A'BCD ⊕ ABCD =
BCD

Figura 2.8 Exemplos de minimização de fórmulas que ocupam 2 células de um mapa com 4 variáveis.



A'BC'D' ⊕ ABC'D' ⊕ A'BCD' ⊕ ABCD'=
BC'D' ⊕ BCD' =
= BD'

Figura 2.9 Exemplo de minimização de fórmula que ocupa 4 células de um mapa com 4 variáveis.

3. A LÓGICA DE PREDICADOS

Conforme visto no Capítulo 1, a Lógica Proposicional é um formalismo interessante, mas não muito expressivo, que, em muitas situações, é insuficiente para representar o conhecimento a respeito de problemas e suas soluções. A Lógica de Predicados é bem mais expressiva e, conseqüentemente, bem mais complexa. Por ser mais abrangente e geral, a Lógica de Predicados é uma formalização adequada para a representação de muitas situações ou problemas para os quais não existe um tratamento conveniente por meio da Lógica Proposicional. A Lógica de Predicados é também conhecida como Lógica de Primeira Ordem ou Cálculo de Predicados.

3.1 CONCEITOS BÁSICOS

Definição 3.1 Uma *teoria de primeira ordem* τ consiste em:

1. um alfabeto;
2. uma linguagem de primeira ordem;
3. um conjunto de axiomas; e
4. um conjunto de regras de inferência.

A linguagem de primeira ordem consiste em fórmulas bem-formadas da teoria que podem ser construídas a partir dos símbolos do alfabeto. Os axiomas são um subconjunto específico de fórmulas bem-formadas. Os axiomas e regras de inferência são usados para derivar os teoremas da teoria. A seguir, são apresentadas sete classes de símbolos que definem um alfabeto de primeira ordem. Algumas convenções de notação adotadas para essas classes, bem como alguns exemplos são apresentados.

Definição 3.2 Um *alfabeto de primeira ordem* consiste em sete classes de símbolos:

(1) *Variáveis*: representadas por uma letra maiúscula seguida por uma cadeia de letras minúsculas ou maiúsculas ou dígitos. Exemplo: X, Xx, YY, Xy, Maria, Z13MN.

(2) *Constantes*: representadas por letras minúsculas ou dígitos, tais como: mar, azul, 3.

(3) *Funções n-árias*: um símbolo de função n-ária é uma letra minúscula seguida por uma cadeia de letras minúsculas ou maiúsculas ou dígitos, agregado a um conjunto de argumentos, tais como: f(Z), f(a), f2(X,y), mae(X,ana). Para enfatizar a aridade (número de argumentos) de um símbolo funcional é usada a notação símbolo/aridade, referenciada como *funtor*. Por exemplo, mae/2.

(4) *Predicados n-ários*: um símbolo de predicado n-ário é uma letra minúscula seguida por uma cadeia de letras minúsculas ou maiúsculas ou dígitos, agregado a um conjunto de

argumentos, tais como: f(Z), f(a), f2(X,y), mae(X,ana). Para enfatizar a aridade (número de argumentos) de um símbolo funcional é usada a notação símbolo/aridade, referenciada como *funtor*. Por exemplo, f2/2. Dois símbolos de predicados são considerados especiais: os símbolos verdade e falso, ambos com aridade 0. Note que funções e predicados têm a mesma simbologia.

(5) *Conectivos*: $\neg$ (negação), $\wedge$ (conjunção), $\vee$ (disjunção), $\rightarrow$ (implicação) e $\leftrightarrow$ (dupla implicação).

(6) *Quantificadores*: $\forall$ (universal) e $\exists$ (existencial).

(7) *Símbolos de pontuação*: ( ) e ,

Observação 3.1 As classes (5), (6) e (7) da Definição 3.2 são constantes para todo o alfabeto, enquanto as classes (1), (2), (3) e (4) variam de alfabeto para alfabeto. Para qualquer alfabeto, apenas as classes (2) e (3) podem ser vazias.

Observação 3.2 Embora na Definição 3.2 constantes tenham sido definidas como uma classe, isso não é necessário. Freqüentemente é conveniente abordar constantes como símbolos funcionais com aridade 0; dessa maneira, o conjunto de constantes é uma subclasse do conjunto de símbolos funcionais. Símbolos predicados de aridade 0 desempenham um papel especial: eles podem ser usados da mesma maneira que as proposições atômicas são usadas na Lógica Proposicional. Uma vez que conectivos também podem ser usados da mesma maneira que são usados na Lógica Proposicional, a Lógica de Predicados é uma generalização da Lógica Proposicional. A estrutura da Lógica Proposicional está imersa na estrutura da Lógica de Predicados.[1]

Observação 3.3 Como mencionada na caracterização sintática de predicados, nenhuma distinção (sintática) é imposta entre símbolos funcionais e símbolos predicados. Fica claro, no contexto, se um determinado símbolo é funcional ou predicado. É importante notar, também, que o conjunto de símbolos funcionais e o conjunto de símbolos predicados podem conter símbolos idênticos, com mesma aridade ou com aridade diferente. Em geral, procurar-se-á usar uma sintaxe similar àquela da linguagem de programação Prolog.

Definição 3.3 *Termos* são definidos recursivamente como:

1. uma constante é termo;
2. uma variável é termo;
3. se f é um funtor (ou seja, um símbolo funcional com aridade n) e t_1, t_2, ..., t_n são termos, então f(t_1, t_2, ..., t_n) é um termo;

---

1 Algumas das convenções adotadas para a Lógica Proposicional serão mantidas e lembradas quando for necessário.

4. Todos os termos são criados usando as regras anteriores.

Exemplo 3.1 Considere um alfabeto A cujo conjunto de conectivos, quantificadores e símbolos de pontuação é o da Definição 3.2 e que tenha conjuntos de:

- constantes: {a,b,c};
- variáveis: {X1,X2,Y};
- símbolos funcionais: {f/1,g/3}; e
- símbolos predicados: {p/2, q/1, r/2}.

A Tabela 3.1 mostra exemplos de expressões que são e que não são termos, considerando o alfabeto A definido anteriormente.

Tabela 3.1 Exemplos de termos e não-termos considerando o alfabeto A.

São termos	Comentários
a	a é uma constante do alfabeto e, conseqüentemente, é um termo.
X1	X1 é uma variável do alfabeto e, conseqüentemente, é um termo.
f(c)	f é um símbolo funcional com aridade 1, e seu argumento (constante c) é um termo.
f(f(X1))	f é um símbolo funcional com aridade 1, e seu argumento, f(X1), é um termo, uma vez que é um símbolo funcional (f) cujo argumento é um termo, dado que é a variável X1.
g(X1,a,f(f(f(a))))	g é um símbolo funcional com aridade 3 e cada um de seus três argumentos são termos: X1 é um termo porque é uma variável; a é um termo porque é uma constante; f(f(f(a))) é um termo porque f é um símbolo funcional; f(f(a)) é um termo uma vez que f é um símbolo funcional; f(a) é um termo uma vez que f é um símbolo funcional e a é uma constante.

Não são termos	Comentários
f(a,b)	f é um símbolo funcional com aridade 1, e não 2.
p(a,b)	Símbolos predicados não são usados para a construção de termos.
$a \vee b$	Conectivos não podem ser usados para a construção de termos.

Tabela 3.1 *Continuação...*

Não são termos	Comentários
f(h(a))	f é símbolo funcional com aridade 1, mas seu argumento, h(a), deveria ser um termo, para que f(h(a)) fosse considerado um termo. O símbolo h, entretanto, não foi definido como símbolo funcional do alfabeto considerado.
f(¬a)	f é símbolo funcional com aridade 1, mas ¬a não é uma constante, uma vez que envolve a negação.
g(b,c,d)	g é símbolo funcional e dois de seus argumentos são constantes do alfabeto. O símbolo d, entretanto, não é um termo neste alfabeto, dado que não é constante, variável ou símbolo funcional aplicado ao termo.

Exemplo 3.2 Considere um alfabeto A cujo conjunto de conectivos, quantificadores e símbolos de pontuação é o da Definição 3.2 e suponha que a variável X e a constante 1 sejam ambos termos em A. Se soma for um símbolo funcional de aridade 2 em A, então são termos:

- soma(X,1);
- soma(soma(X,1),1);
- soma(X,soma(X,1));
- soma(soma(X,1),soma(soma(1,X),1)), etc...

Exemplo 3.3 Como formalmente será definido na Definição 3.9, fórmulas atômicas são bem-formadas e podem ser combinadas por meio de conectivos lógicos, formando novas proposições.

(a) Considere, por exemplo, que mae/2 é um símbolo predicado da linguagem, e que ana e maria são dois símbolos constantes (ou seja, são termos). A fórmula atômica mae(ana,maria) pode ser usada para expressar que ana é mãe de maria e, então, pode ser parte de uma fórmula composta, tal como:

$$\text{mae(ana, maria)} \rightarrow \neg\text{mae(maria,ana)}$$

Se o símbolo predicado filha/2 for também definido como parte do alfabeto, outra fórmula bem-formada poderia ser:

$$\text{mae(ana, maria)} \rightarrow \text{filha(maria,ana)}$$

(b) Considere, por exemplo, que cachorro/1, tem/2, cor/2 sejam símbolos predicados da linguagem e que cariri, cauda e marrom sejam três símbolos constantes. As fórmulas atômicas cachorro(cariri), tem(cariri,cauda) e cor(cariri,marrom) podem ser combinadas na fórmula composta:

$$\text{cachorro(cariri)} \wedge \text{tem(cariri,cauda)} \wedge \text{cor(cariri,marrom)}$$

para exprimir que um cachorro identificado como cariri tem cauda e tem cor marrom.

Definição 3.4 Se p é um símbolo predicado n-ário e $t_1, t_2, ..., t_n$ são termos, então $p(t_1, t_2, ..., t_n)$ é um *átomo* ou *fórmula atômica*. Fórmulas atômicas são comumente chamadas de predicados.

Exemplo 3.4 As variáveis X, Y e a constante 1 são termos. Se soma/3 foi definido como um símbolo predicado, então soma(X,1,Y) é um átomo.

Observação 3.4 Observe que:

- constantes representam objetos em um domínio;
- variáveis são usadas para fazer abstrações sobre constantes e termos;
- termos representam objetos em um domínio;
- símbolos funcionais permitem a criação de novos termos a partir dos termos existentes;
- símbolos predicados permitem o estabelecimento de assertivas a respeito de termos, ou seja, a respeito de objetos em um domínio.

Antes de definir formalmente o conceito de fórmula, é importante distinguir entre variáveis livres e variáveis ligadas.

Definição 3.5 O *escopo de um quantificador* ocorrendo em uma fórmula α é a fórmula à qual o quantificador se aplica.

Exemplo 3.5

(a) O escopo do quantificador $\forall$ em $(\forall X\ \alpha)$ é α. O escopo do quantificador $\exists$ em $(\exists X\ \alpha)$ é α.

(b) O escopo do quantificador $\forall$ em $(\forall X\ (p(X,Y) \rightarrow q(X)))$ é a fórmula $(p(X,Y) \rightarrow q(X))$.

(c) O escopo do quantificador $\exists$ em $(\exists X(\forall Y(p(X,Y)) \wedge q(X)))$ é a fórmula $(\forall Y(p(X,Y)) \wedge q(X))$. Nessa mesma fórmula, o escopo do quantificador $\forall$ é a fórmula $(p(X,Y))$.

Definição 3.6 A ocorrência de uma variável em uma fórmula é *ligada* se e somente se a ocorrência está dentro do escopo do quantificador ou, então, é a ocorrência que segue imediatamente o quantificador. A ocorrência de uma variável em uma fórmula é *livre* se e somente se a ocorrência da variável não for ligada.

Definição 3.7 Uma variável é *livre* em uma fórmula se pelo menos uma ocorrência da variável for livre na fórmula. Uma variável é *ligada* em uma fórmula se pelo menos uma ocorrência da variável for ligada.

Definição 3.8 Uma *fórmula fechada* é uma fórmula que não tem ocorrência livre de qualquer variável.

Exemplo 3.6

(a) Na fórmula

$$(\exists \underset{\uparrow}{X}\ p(\underset{\uparrow}{X},Y)) \vee q(\underset{\uparrow}{X})$$

as duas primeiras ocorrências de X são ligadas e a terceira ocorrência é livre, uma vez que o escopo de $\exists X$ é $p(X,Y)$. A variável Y é livre na fórmula.

(b) Na fórmula

$$\exists \underset{\uparrow}{X}(p(\underset{\uparrow}{X},Y) \vee q(\underset{\uparrow}{X}))$$

as três ocorrências de X são ligadas, uma vez que o escopo de $\exists X$ é $(p(X,Y) \vee q(X))$. A variável Y é livre na fórmula.

(c) Na fórmula

$$(\forall X\ p(X,Y))$$

uma vez que ambas as ocorrências de X são ligadas, a variável X é ligada. A variável Y, entretanto, é livre, uma vez que a única ocorrência de Y é livre.

(d) Na fórmula

$$(\forall X\ p(X,\underset{\uparrow}{Y})) \wedge (\forall \underset{\uparrow}{Y}\ q(\underset{\uparrow}{Y}))$$

a variável Y é livre e ligada. A primeira ocorrência é livre e as duas outras, ligadas.

Definição 3.9 Uma *fórmula bem-formada* – wff – é definida recursivamente como:

(1) os símbolos especiais verdade, falso e símbolos predicados de aridade 0 são fórmulas;

(2) um átomo é uma fórmula;

(3) se α e β forem fórmulas, então são fórmulas:

$$(\alpha)$$

$$(\neg\alpha)$$

$$(\alpha \wedge \beta)$$

$$(\alpha \vee \beta)$$

$$(\alpha \rightarrow \beta)$$

$$(\alpha \leftrightarrow \beta)$$

(4) se α for uma fórmula e X for uma variável livre em α, então $(\forall X\ \alpha)$ e $(\exists X\ \alpha)$ são fórmulas;

(5) fórmulas são geradas apenas por meio de um número finito de aplicações de (1), (2), (3) e (4).

Exemplo 3.7 Considerando o alfabeto A do Exemplo 3.1, a Tabela 3.2 mostra exemplos de fórmulas e não-fórmulas.

Tabela 3.2 Exemplos de fórmulas e não-fórmulas considerando o alfabeto A.

São fórmulas	Comentários
p(a,a)	a é uma constante do alfabeto e, conseqüentemente, é um termo, e p é um símbolo predicado de aridade 3.
q(Y)	Y é uma variável do alfabeto e, conseqüentemente, é um termo, e q é um símbolo predicado de aridade 1.
r(b,f(c))	f é um símbolo funcional com aridade 1, e seu argumento (constante c) é um termo. A letra b é uma constante no alfabeto e, portanto, é um termo. A expressão dada, portanto, é uma fórmula válida.
(r(b,f(c)) ∧ p(a,a))	As linhas 1 e 3 dessa tabela mostram que r(b,f(c)) e p(a,a) são fórmulas válidas e, conseqüentemente, sua conjunção é também uma fórmula válida.
(¬∃X1 (q(f(X1))))	f(X1) é um termo e q é um símbolo predicado de aridade 1. Portanto, q(f(X1)) é uma fórmula válida. Como X1 ocorre livre na fórmula, ela quantificada existencialmente é também uma fórmula. Sua negação, conseqüentemente, é também uma fórmula.
q(g(X1,a,f(f(f(a)))))	Como visto na Tabela 3.1, g(X1,a,f(f(f(a)))) é um termo válido. Como q é um símbolo predicado de aridade 1, a expressão é uma fórmula válida.

Tabela 3.2 *Continuação...*

Não são fórmulas	Comentários
(∀a p(a,a))	a não é uma variável.
r(a,b,c)	a, b, c são constantes e, conseqüentemente, termos. Como r é um símbolo predicado de aridade 2, a expressão não é uma fórmula valida.
p(a,q(c))	q(c) não é um termo, e sim uma fórmula; conseqüentemente, não pode ser usado como argumento de um símbolo predicado; a expressão não é uma fórmula válida.
(q(f(b)) ∧ g(a,Y,X2))	g(a,Y,X2) não é uma fórmula, mas sim um termo; conseqüentemente, a expressão toda não é válida.

Definição 3.10 Um termo *ground* é um termo que não contém variáveis; de maneira semelhante, uma fórmula *ground* é uma fórmula que não contém variáveis.

Exemplo 3.8

(a) Considere o alfabeto A do Exemplo 3.1. Os termos g(a,b,a), g(g(a,b,a),b,c), f(a), f(g(a,b,c)) são todos *ground*. Já os termos g(X1,a,b), f(X2), g(f(X1),a,b) não são *ground*.

(b) Considerando o mesmo alfabeto, as fórmulas q(g(c,a,f(f(f(a))))), q(a), p(b,a) são *ground*; já a fórmula q(g(X1,a,f(f(f(a))))) não é *ground*.

Definição 3.11 A *linguagem de primeira ordem* λ dada por um alfabeto é o conjunto de todas as fórmulas bem-formadas que podem ser construídas a partir dos símbolos do alfabeto.

Definição 3.12 Se α for uma fórmula, então ∀(α) denota o *fechamento universal* de α, que é a fórmula fechada obtida pela adição de um quantificador universal associado a cada variável que ocorre livre em α. De maneira semelhante, ∃(α) denota o *fechamento existencial* de α, obtido pela adição de um quantificador existencial associado a cada variável que ocorre livre em α.

Exemplo 3.9 Considere a fórmula α: (∀X p(X,Y)). Como a variável Y ocorre livre em α, o fechamento universal de α é a fórmula ∀Y(∀X p(X,Y)) e o fechamento existencial de α é a fórmula ∃Y(∀X p(X,Y)).

Definição 3.13 Um *literal* é um átomo ou a negação de um átomo. Um literal positivo é um átomo. Um literal negativo é a negação do átomo.

Observação 3.5 É importante observar que os literais L e ¬L são ditos complementares um do outro e formam, em qualquer ordem, um par complementar.

Definição 3.14 Uma *cláusula* é uma fórmula na forma:

$$\forall X_1 \forall X_2 \forall X_3 .... \forall X_p (L_1 \vee L_2 \vee L_3 \vee ... \vee L_q)$$

na qual cada L_i $(1 \leq i \leq q)$ é um literal, e $X_1, X_2, ..., X_p$ são todas as variáveis que ocorrem em $L_1, L_2, ..., L_q$.

Analogamente ao que foi descrito na Seção 1.6.4, uma cláusula pode ser representada pelo conjunto finito de seus literais. Uma vez que cláusulas são bastante comuns em Programação Lógica, é conveniente adotar uma notação clausal conveniente. Reescrevendo a cláusula anterior, identificando os literais positivos dos negativos, tem-se:

$$\forall X_1 \forall X_2 \forall X_3 .... \forall X_p (A_1 \vee A_2 \vee ... \vee A_m \vee \neg B_1 \vee \neg B_2 \vee ... \vee \neg B_n)$$

na qual $A_1, A_2, ..., A_m, B_1, B_2, ..., B_n$ são átomos e $X_1, X_2, ..., X_p$ são as variáveis que ocorrem nos átomos. Mudando a notação, pode-se escrever que:

$$A_1, A_2, ..., A_m \leftarrow B_1, B_2, ..., B_n$$

com o significado de:

$$A_1 \vee A_2 \vee ... \vee A_m \leftarrow B_1 \wedge B_2 \wedge ... \wedge B_n$$

Assim, na notação clausal, todas as variáveis são consideradas universalmente quantificadas. Os seguintes casos especiais podem acontecer:

1. $m > 1$ as conclusões são indefinidas, ou seja, existe mais do que uma conclusão;
2. $m = 1$, $n \geq 0$, conhecida como cláusula definida;
3. $m = 0$, $n = 0$, conhecida como cláusula vazia e representada por *nil*.

Exemplo 3.10 Reescrevendo uma cláusula:

$$(\forall X(\forall Y(\forall Z(p(X,a,b) \vee \neg q(b,Z) \vee r(X,Y) \vee \neg s(Z) \vee t(X,Z)))))$$
$$\Downarrow$$
$$\forall X \forall Y \forall Z(p(X,a,b) \vee \neg q(b,Z) \vee r(X,Y) \vee \neg s(Z) \vee t(X,Z))$$
$$\Downarrow$$
$$\forall X \forall Y \forall Z(p(X,a,b) \vee r(X,Y) \vee t(X,Z) \vee \neg q(b,Z) \vee \neg s(Z))$$
$$\Downarrow$$
$$\forall X \forall Y \forall Z(A_1 \vee A_2 \vee A_3 \vee B_1 \vee B_2) \text{ na qual}$$
$$A_1 = p(X,a,b); A_2 = r(X,Y); A_3 = t(X,Z); B_1 = q(b,Z); B_2 = s(Z)\ (M=3\ \&\ n = 2)$$
$$\Downarrow$$

$$p(X,a,b),\ r(X,Y),\ t(X,Z) \leftarrow q(b,Z),\ s(Z)$$

significando

$$p(X,a,b) \vee r(X,Y) \vee t(X,Z) \leftarrow q(b,Z) \wedge s(Z)$$

Definição 3.15 Uma *cláusula definida de programa* é uma cláusula que contém exatamente um literal positivo. Tem a forma:

$$\underbrace{A}_{\text{cabeça}} \leftarrow \underbrace{B_1 \wedge B_2 \wedge \ldots \wedge B_n}_{\text{corpo}}$$

na qual A, $B_1, B_2, \ldots, B_n$ são átomos. O literal positivo A é chamado de *cabeça* e os átomos $B_1, B_2, \ldots, B_n$ são coletivamente chamados de *corpo* da cláusula definida de programa.

Observação 3.6 Apenas átomos são permitidos no corpo de cláusulas definidas de programa. Em Prolog, entretanto, literais da forma *not* B, tal que B é um átomo, são permitidos, e o *not* é interpretado sob a regra de negação-como-falha.

Definição 3.16 Uma *cláusula de programa* é uma cláusula na forma:

$$A \leftarrow L_1 \wedge L_2 \wedge \ldots \wedge L_n$$

na qual A é um átomo e cada um dos $L_1, L_2, \ldots, L_n$ é da forma B ou $\neg B$, e B é um átomo.

Definição 3.17 Uma *cláusula unitária* é uma cláusula na forma:

$$A \leftarrow$$

ou seja, uma cláusula de programa (ou uma cláusula definida de programa) com um corpo vazio. Em terminologia Prolog, tal cláusula é denominada *fato* ou, ainda, *cláusula definida incondicional* e é simplesmente denotada por A.

Definição 3.18 Uma *cláusula meta* é uma cláusula na forma:

$$\leftarrow B_1 \wedge B_2 \wedge \ldots \wedge B_n$$

ou seja, uma cláusula que não tem cabeça. Cada B_i $(1 \leq i \leq n)$ é chamado de submeta da cláusula meta.

Observação 3.7 Se $Y_1, \ldots, Y_r$ são as variáveis da cláusula meta $\leftarrow B_1 \wedge B_2 \wedge \ldots \wedge B_n$, essa notação clausal é uma simplificação de

$$\forall Y_1 \forall Y_2 \forall Y_3 \ldots \forall Y_r (\neg B_1 \vee \neg B_2 \vee \ldots \vee \neg B_n)$$

ou, equivalentemente,

$$\neg(\exists Y_1 \exists Y_2 \exists Y_3 ... \exists Y_r (B_1 \wedge B_2 \wedge ... \wedge B_n)).$$

Definição 3.19 Um conjunto finito de cláusulas de programa é chamado de *programa lógico*, e, conseqüentemente, um programa lógico não contém qualquer cláusula meta.

Definição 3.20 Em um programa lógico, o conjunto de todas as cláusulas de programa com o mesmo predicado p como cabeça é chamado de *definição de p*.

Definição 3.21 Uma *cláusula de Horn* é uma cláusula que ou é uma cláusula de programa ou uma cláusula meta. Isso significa que uma cláusula de Horn contém no máximo um literal positivo.

Observação 3.8 É importante lembrar que duas fórmulas α e β são equivalentes ($\alpha \equiv \beta$) se e somente se os valores-verdade de α e β são os mesmos sob toda interpretação. As equivalências básicas da Lógica Proposicional discutidas no Capítulo 1 continuam válidas em Lógica de Predicados. Além daqueles, existem pares de fórmulas equivalentes contendo quantificadores, que são apresentados a seguir.

Seja α uma fórmula expressa usando a variável X. Para enfatizar esse fato, α será notada por $\alpha[X]$. Seja β uma fórmula que não é expressa usando a variável X. Os seguintes pares de fórmulas são equivalentes, nos quais Q representa qualquer um dos quantificadores: $\forall$ ou $\exists$.

(a)	$(QX\ \alpha[X]) \vee \beta$	$\equiv$	$(QX\ (\alpha[X] \vee \beta))$
(b)	$(QX\ \alpha[X]) \wedge \beta$	$\equiv$	$(QX\ (\alpha[X] \wedge \beta))$
(c)	$\neg(\forall X\ \alpha[X])$	$\equiv$	$(\exists X\ (\neg\alpha[X]))$
(d)	$\neg(\exists X\ \alpha[X])$	$\equiv$	$(\forall X\ (\neg\alpha[X]))$

Prova de (c):

Seja I uma interpretação arbitrária sobre um domínio D.

- se $\neg(\forall X\ \alpha[X])$ é verdade em I, então $(\forall X\ \alpha[X])$ é falso em I. Isto significa que existe um elemento *e* em D tal que $\alpha[e]$ é falso, isto é, $\neg\alpha[e]$ é verdade em I. Portanto, $(\exists X\ (\neg\alpha[X]))$ é verdade em I;
- se $\neg(\forall X\ \alpha[X])$ é falso em I, então $(\forall X\ \alpha[X])$ é verdade em I. Isto significa que $\alpha[X]$ é verdade para todo elemento X em D, isto é, $\neg\alpha[X]$ é falso para todo elemento em D. Portanto, $(\exists X\ (\neg\alpha\ [X]))$ é falso em I.

Desde que $\neg(\forall X\ \alpha[X])$ e $(\exists X\ (\neg\alpha[X]))$ sempre assumam o mesmo valor-verdade para toda interpretação arbitrária, por definição: $\neg(\forall X\ \alpha[X]) \equiv (\exists X\ (\neg\alpha[X]))$.

Considere que α e β sejam duas fórmulas expressas usando a variável X e, por essa razão, ambas serão notadas como $\alpha[X]$ e $\beta[X]$, respectivamente. As seguintes equivalências são válidas:

$$\text{(e)}\quad (\forall X\, \alpha[X]) \wedge (\forall X\, \beta[X]) \;\equiv\; (\forall X\, (\alpha[X] \wedge \beta[X]))$$

$$\text{(f)}\quad (\exists X\, \alpha[X]) \vee (\exists X\, \beta[X]) \;\equiv\; (\exists X)\, (\alpha[X] \vee \beta[X])$$

Isto é, os quantificadores $\forall$ e $\exists$ podem distribuir-se sobre $\wedge$ e $\vee$, respectivamente. Entretanto, os quantificadores $\forall$ e $\exists$ não se distribuem sobre $\vee$ e $\wedge$, respectivamente, ou seja:

(g) $(\forall X\, \alpha[X]) \vee (\forall X\, \beta[X])$ não é equivalente a $(\forall X\, (\alpha[X] \vee \beta[X]))$

(h) $(\exists X\, \alpha[X]) \wedge (\exists X\, \beta[X])$ não é equivalente a $(\exists X\, (\alpha[X] \wedge \beta[X]))$

Considere um exemplo de fórmulas como em (g) e considere uma interpretação I na qual as fórmulas são avaliadas como segue:

$\alpha[1]$	$\alpha[2]$	$\beta[1]$	$\beta[2]$
v	f	f	v

Pode-se dizer, então, que $(\forall X\, \alpha[X])$ e $(\forall X\, \beta[X])$ são falsos. Portanto, $(\forall X\, \alpha[X]) \vee (\forall X\, \beta[X])$ é falso. Por outro lado, a fórmula $(\forall X\, (\alpha[X] \vee \beta[X]))$ é sempre avaliada verdade em I, pois, para:

$$X = 1 \Rightarrow \alpha[1] \vee \beta[1] = \text{verdade}$$

$$X = 2 \Rightarrow \alpha[2] \vee \beta[2] = \text{verdade}$$

Uma vez que toda variável ligada em uma fórmula pode ser considerada uma variável *dummy*, toda variável X pode ser renomeada Z e a fórmula $(\forall X\, \beta[X])$ torna-se $(\forall Z\, \beta[Z])$, isto é, $(\forall X\, \beta[X]) \equiv (\forall Z\, \beta[Z])$. Suponha que seja escolhida a variável Z que não aparece em $\alpha[X]$. Então,

$$(\forall X\, \alpha[X]) \vee (\forall X\, \beta[X]) \overset{\text{(a)}}{\equiv} (\forall X\, \alpha[X]) \vee (\forall Z)\, \beta[Z] \equiv (\forall X(\forall Z(\alpha[X] \vee \beta[Z]))).$$

De maneira similar,

$$(\exists X\, \alpha[X]) \wedge (\exists X\, \beta[X]) \overset{\text{(b)}}{\equiv} (\exists X\, \alpha[X]) \wedge (\exists Z\, \beta[Z]) \equiv (\exists X(\exists Z(\alpha[X] \wedge \beta[Z]))).$$

Em geral, tem-se:

$$(Q_1 X\, \alpha[X]) \vee (Q_2 X\, \beta[X]) \equiv (Q_1 X(Q_2 Z\, (\alpha[X] \vee \beta[Z])))$$

$$(Q_3 X\, \alpha[X]) \wedge (Q_4 X\, \beta[X]) \equiv (Q_3 X(Q_4 Z\, (\alpha[X] \wedge \beta[Z])))$$

nas quais Q_1, Q_2, Q_3, Q_4 são $\forall$ ou $\exists$ e a variável Z não comparece em $\alpha[X]$. Se $Q_1 = Q_2 = \exists$ e $Q_3 = Q_4 = \forall$, não é preciso a renomeação das variáveis Xs em $(Q_2X)\ \beta[X]$ ou $(Q_4X)\ \beta[X]$ e pode-se usar (f) e (e) diretamente.

3.2 A LÓGICA DE PREDICADOS COMO LINGUAGEM DE REPRESENTAÇÃO DE CONHECIMENTO

Esta seção inicialmente apresenta algumas sugestões e diretivas de como abordar o processo de tradução de uma expressão em língua natural (no caso, português) para a linguagem da Lógica de ~~Predicados~~. Na subseção seguinte são apresentados vários exemplos de tradução de expressões em português para a Lógica de Predicados.

3.2.1 Usando lógica de predicados para representar sentenças em língua natural

Algumas sugestões de tradução de sentenças em língua natural para a Lógica de Predicados discutidas nesta subseção estão fundamentadas nas propostas descritas em Grassmann & Tremblay (1996) e foram adaptadas à notação usada neste material. Considere as seguintes sentenças:

(1) Todos tiram férias de vez em quando.

(2) Todos os cachorros têm cauda.

(3) Algumas pessoas gostam de peixe cru.

As três sentenças indicam quão freqüentemente determinadas situações acontecem (ou não). Na Lógica de Predicados, dois quantificadores são utilizados neste contexto: o universal, que relaciona uma determinada assertiva a todos os indivíduos de um certo domínio, e o existencial, que relaciona a assertiva a alguns indivíduos do domínio.

Sentenças contendo palavras tais como 'tudo', 'cada' e 'todos' usualmente indicam quantificação universal. Tais assertivas devem ser parafraseadas usando a expressão "para todo X" que, então, é traduzida para $\forall X$.

Para a representação da sentença (1), se a situação 'tiram férias de vez em quando' for representada pelo predicado ferias, então, ferias(X) significa que X tira férias de vez em quando. A palavra 'todos' indica que a assertiva se refere a todos os indivíduos do domínio considerado, o que leva à representação ($\forall X$ ferias(X)).

Considerando a assertiva (2) e que o domínio considerado seja o domínio de animais, então, primeiramente, deve-se encontrar o escopo do quantificador universal que, no caso, é "se X for um cachorro, então X tem uma cauda". Considerando os predicados cachorro e temcauda com aridade 1, a assertiva (2) pode ser representada em Lógica de Predicados por ($\forall$X (cachorro(X) $\rightarrow$ temcauda(X))). Outra maneira de representar essa assertiva é ($\forall$Y (cachorro(Y) $\rightarrow$ temcauda(Y))), desde que Y seja também um símbolo de variável válido no alfabeto que está sendo considerado.

Já as palavras 'algum', 'alguns' e 'pelo menos um' sugerem quantificação existencial, devendo ser parafraseadas como "existe um X tal que", traduzido por $\exists$X. Portanto, considerando o domínio de pessoas e que o predicado p representa 'gostam de peixe cru', a sentença (2) pode ser expressa em Lógica de Predicados por ($\exists$X p(X)).

Os quantificadores $\forall$X e $\exists$X devem ser tratados como se fossem conectivos unários, com prioridade mais alta do que qualquer conectivo binário. Em uma situação em que p(X) representa 'X está vivo' e que q(X) representa 'X está morto', a representação correta de que tudo ou está vivo ou está morto é ($\forall$X (p(X) $\vee$ q(X))), e não ($\forall$X (p(X)) $\vee$ q(X), que significa que tudo está vivo ou que um determinado elemento do domínio está morto (em razão da atribuição à variável, parte da avaliação de uma fórmula, vista em XX).

Quantificadores podem aparecer aninhados, como na representação da assertiva "Existe alguém que conhece todo mundo". Suponha que o predicado conhece(X,Y) representa que X conhece Y. A melhor maneira de abordar a representação da assertiva é por partes. Informalmente, pode-se escrever $\exists$X(X conhece todo mundo). A sentença "X conhece todo mundo" significa que para todo Y, X conhece Y, o que pode ser representada por ($\forall$Y conhece(X,Y)). A expressão final é: ($\exists$X($\forall$Y conhece(X,Y))).

Considere agora a assertiva "Todos têm alguém por mãe". Usando o predicado mae(X,Y) para representar que X é mãe de Y, a assertiva "alguém é mãe de Y" é escrita como ($\exists$X(mãe(X,Y))). Para representar que todos têm alguém por mãe, usa-se ($\forall$Y($\exists$X(mãe(X,Y)))).

Algumas vezes, a quantificação deve ser feita sobre um subconjunto de elementos do domínio. Suponha que o domínio seja o conjunto de todos os animais e que as assertivas a serem representadas sejam "Todos os cachorros são mamíferos" e "Alguns cachorros são marrons". Uma vez que o quantificador deve se restringir a cachorros, a assertiva pode ser parafraseada como "se X for um cachorro, então X é um mamífero", que é representada por ($\forall$Y (cachorro(X) $\rightarrow$ mamifero(X))). Geralmente, a expressão lógica ($\forall$X (p(X) $\rightarrow$ q(X))) pode ser traduzida como "todos os indivíduos com a propriedade p têm também a propriedade q".

Por outro lado, a assertiva "Alguns cachorros são marrons" significa que existem alguns animais que são cachorros e que são marrons. A assertiva "X é cachorro e X é marrom" pode ser escrita como cachorro(X) $\wedge$ marrom(X) e a assertiva "Existem alguns cachorros marrons" pode ser representada como ($\exists$X (cachorro(X) $\wedge$ marrom(X))). Geralmente, a expressão lógica ($\exists$X (p(X) $\wedge$ q(X))) pode ser traduzida como "alguns indivíduos com a propriedade p têm também a propriedade q".

Note, pois, que, se o quantificador universal deve ser aplicado apenas a elementos com uma determinada propriedade, uma expressão condicional deve ser usada para restringir o domínio.

Por outro lado, se o uso do quantificador existencial deve ser restringido, uma conjunção deve ser usada.

A conversão de assertivas que contém a palavra 'apenas', tal como "Apenas cachorros latem", exige uma reescrita tal como "Late apenas se for cachorro" ou, equivalentemente, "Se late, então é cachorro", representada por (∀Z (late(Z) → cachorro(Z))).

3.2.2 Exemplos de tradução de sentenças para expressões da lógica de predicados

Cada exemplo apresentado nesta subseção traz uma sentença em português identificada por um número, a sua representação em Lógica de Predicados e comentários sobre a representação. A simbolização que segue é para um domínio fixado inicialmente.

(1) Sentença	Lógica de predicados
Ana é estudiosa.	estudiosa(ana)

Literal (átomo), no qual estudiosa é o símbolo predicado, e ana é símbolo constante.

(2) Sentença	Lógica de predicados
Pedro é advogado, e Guilherme é estudante.	advogado(pedro) ∧ estudante(guilherme)

Fórmula composta da conjunção de dois átomos. No primeiro, advogado é o símbolo predicado e pedro, o símbolo constante. No segundo, estudante é o símbolo predicado e guilherme, o símbolo constante.

(3) Sentença	Lógica de predicados
Maria vai à festa ou ao teatro.	festa(maria) ∨ teatro(maria)

Fórmula composta da disjunção de dois átomos. No primeiro, festa é o símbolo predicado e maria, o símbolo constante. No segundo, teatro é o símbolo predicado.

(4) Sentença	Lógica de predicados
Gabriel não é programador.	¬programador(gabriel)

Literal (átomo negado). O símbolo predicado é programador, e gabriel é o símbolo constante.

(5) Sentença	Lógica de predicados
Pedro e Guilherme são primos.	primos(pedro,guilherme)

Literal (átomo), no qual primos é símbolo predicado, e pedro e guilherme são símbolos constantes.

(6) SENTENÇA	LÓGICA DE PREDICADOS
Maria e Ana não são primas. Maria e Ana são vizinhas.	¬primas(maria,ana) ∧ vizinhas(maria,ana)

Fórmula composta conjuntiva. O primeiro literal tem por símbolo predicado primas, e o segundo, vizinhas. Ambos os símbolos são aplicados aos termos maria e ana, que são símbolos constantes.

(7) SENTENÇA	LÓGICA DE PREDICADOS
se a função f1 for diferenciável, ela é contínua.	diferenciavel(f1) → continua(f1)

Fórmula composta condicional, cujos predicados envolvidos são diferenciavel e continua, e o único termo empregado é a constante f1.

(8) SENTENÇA	LÓGICA DE PREDICADOS
A função f2 é contínua, mas não é diferenciável.	continua(f2) ∧ ¬diferenciavel(f2)

Fórmula composta conjuntiva, cujos predicados envolvidos são diferenciavel e continua, e o único termo empregado é a constante f2.

(9) SENTENÇA	LÓGICA DE PREDICADOS
Antonio faltou à aula, mas Ana não faltou.	faltou(antonio) ∧ ¬faltou(ana)

Fórmula composta conjuntiva, cujo único predicado envolvido é faltou, e os termos empregados são as constantes ana e antonio.

(10) SENTENÇA	LÓGICA DE PREDICADOS
Se Pedro é mais alto que Ana, e Ana é mais alta que André, então Pedro é mais alto que André.	(alto(pedro,ana) ∧ alto(ana,andre)) → alto(pedro,andre)

Fórmula composta condicional, cujo antecedente é uma conjunção. O único símbolo predicado utilizado é alto. Três símbolos constantes foram usados: pedro, ana, andre.

(11) SENTENÇA	LÓGICA DE PREDICADOS
Maria lê jornal.	le(maria,jornal).

Fórmula atômica, cujo símbolo predicado é le e com dois argumentos, que são os termos constantes maria e jornal.

(12) SENTENÇA	LÓGICA DE PREDICADOS
A mãe de Maria gosta de Maria.	gosta(mae(maria),maria)

Fórmula atômica, cujo símbolo predicado é gosta. Um dos argumentos é o símbolo funcional mae, com aridade 1. O único símbolo constante é maria, que é usado tanto como argumento do símbolo funcional mae quanto do símbolo predicado gosta.

(13) SENTENÇA	LÓGICA DE PREDICADOS
A mãe da mãe de Maria gosta da mãe de Maria e gosta de Maria.	gosta(mae(mae(maria)),mae(maria)) ∧ gosta(mae(mae(maria)),maria)

Fórmula composta conjuntiva, cujo símbolo predicado que define os dois átomos envolvidos na fórmula é gosta. No primeiro átomo, o primeiro argumento é o símbolo funcional mae, com aridade 1, cujo argumento é o símbolo funcional mae com argumento maria. O segundo argumento do primeiro átomo é o símbolo funcional mae com argumento maria. O segundo átomo tem por primeiro argumento o mesmo primeiro argumento do primeiro átomo e, como segundo argumento, o termo constante maria.

(14) SENTENÇA	LÓGICA DE PREDICADOS
Todos os homens são mortais.	(∀X mortal(X))

Fórmula quantificada universalmente, cujo único símbolo predicado usado é mortal. Está subentendido que o domínio é o conjunto dos seres humanos. Na eventualidade do domínio ser, por exemplo, seres vivos, a sentença em língua natural seria: (∀X (homem(X) → mortal(X))).

(15) SENTENÇA	LÓGICA DE PREDICADOS
Alguns homens são mortais.	(∃X mortal(X))

Fórmula quantificada existencialmente, cujo único símbolo predicado usado é mortal. Está subentendido que o domínio é o conjunto dos seres humanos. Na eventualidade do domínio ser, por exemplo, seres vivos, a sentença em língua natural seria: (∃X (homem(X) ∧ mortal(X))).

(16) SENTENÇA	LÓGICA DE PREDICADOS
Nenhum homem é mortal.	(∀X ¬mortal(X))

Fórmula quantificada existencialmente, cujo único símbolo predicado usado é mortal. Está subentendido que o domínio é o conjunto dos seres humanos.

(17) SENTENÇA	LÓGICA DE PREDICADOS
Alguns cachorros latem, outros não.	(∃X late(X)) ∧ (∃X ¬late(X))

Fórmula quantificada existencialmente, cujo único símbolo predicado usado é late. Está subentendido que o domínio é o conjunto dos cachorros. Note que a representação em Lógica estaria incorreta se fosse escrita como: (∃X late(X) ∧ ¬late(X)), uma vez que a sentença estaria afirmando que existe um objeto do domínio (um cachorro, no caso) que late e que não late.

(18) SENTENÇA	LÓGICA DE PREDICADOS
Existem pessoas bondosas, no entanto nem todas as pessoas são bondosas.	(∃X bondosa(X)) ∧ ¬(∀X bondosa(X))

Considerando o domínio dos seres humanos, essa é uma fórmula composta (conjunção) que pode ser lida como: existem pessoas que são bondosas e não é verdade que todas as pessoas são bondosas.

(19) SENTENÇA	LÓGICA DE PREDICADOS
Se Pedro não é estudioso, nenhum dos rapazes é estudioso.	¬estudioso(pedro) → (∀X ¬estudioso(X))

Fórmula composta (condicional).

(20) SENTENÇA	LÓGICA DE PREDICADOS
Se Guilherme acordar tarde, então Gabriel irá ao mercado e alguém vai gastar dinheiro.	acordartarde(guilherme) → (irmercado(gabriel) ∧ (∃X gastar(X)))

Fórmula composta (condicional).

(21) SENTENÇA	LÓGICA DE PREDICADOS
Se Guilherme acordar tarde, então Gabriel irá ao mercado e alguém vai gastar dinheiro.	acordartarde(guilherme) → (irmercado(gabriel) ∧ (∃X gastar(X)))

Fórmula composta (condicional).

(22) SENTENÇA	LÓGICA DE PREDICADOS
Se o abacate e o caqui engordam, então há alimentos que engordam.	(engorda(abacate) ∧ engorda(caqui)) → (∃X engorda(X)))

Fórmula composta (condicional), cujo antecedente é uma conjunção de dois átomos. O domínio é o conjunto de alimentos.

(23) SENTENÇA	LÓGICA DE PREDICADOS
Alguns alunos estudam, e nem todos os alunos estudam.	(∃X estuda(X)) ∧ ¬(∀X estuda(X))
Fórmula composta (conjunção), cujo domínio é o conjunto de estudantes.	

(24) SENTENÇA	LÓGICA DE PREDICADOS
Todo aluno é mais novo que alguns professores.	(∀X (aluno(X) → (∃Y(professor(Y) ∧ maisnovo(X,Y)))))
Fórmula composta (condicional), cujo conseqüente é uma expressão composta (conjunção).	

(25) SENTENÇA	LÓGICA DE PREDICADOS
Nem todos os pássaros podem voar.	¬(∀X (passaro(X) → voa(X)))
O domínio é o conjunto dos animais. Alternativamente, a sentença em Lógica de Predicados poderia ter sido escrita como ¬(∃X (passaro(X) ∧ ¬(voa(X))).	

(26) SENTENÇA	LÓGICA DE PREDICADOS
Toda criança é mais nova que a mãe da criança.	(∀X(∀Y ((crianca(X) ∧ mae(Y,X)) → maisnova(X,Y))))
Alternativamente, a sentença em Lógica de Predicados poderia ter sido escrita como (∀X (crianca(X) → maisnova(X,mae(X)))). Note que nesta expressão mae/1 é usado como símbolo funcional; na expressão anterior, mae/2 é um símbolo predicado.	

(27) SENTENÇA	LÓGICA DE PREDICADOS
Nenhum número natural é negativo.	¬(∃X negativo(X))
Pré-condição: o domínio é o conjunto dos números naturais.	

Nenhum número natural é negativo.	¬(∃X (natural(X) ∧ negativo(X)))
Pré-condição: o domínio é o conjunto dos números inteiros.	

Nenhum número natural é negativo.	¬(∃X (natural(X) ∧ negativo(X)))
Pré-condição: o domínio é o conjunto dos números reais.	

(28) SENTENÇA	LÓGICA DE PREDICADOS
Alguns números primos são pares.	$(\exists X\ (primo(X) \wedge par(X)))$

Pré-condição: o domínio é o conjunto dos números naturais. Os dois predicados usados são: primo e par, ambos com aridade 1. O único termo que comparece na expressão é a variável X.

Todos os números pares são maiores que 1.	$(\forall X\ (par(X) \rightarrow maior(X,1)))$

Pré-condição: o domínio é o conjunto dos números naturais. Os dois predicados usados são: par e maior, com aridade 1 e 2, respectivamente. Os dois termos que comparecem na expressão são: variável X e constante 1.

Número pares são primos apenas se forem menores que 3.	$(\forall X\ ((par(X) \wedge primo(X)) \rightarrow menor(X,3)))$

Pré-condição: o domínio é o conjunto dos números naturais. A fórmula usa três predicados: par, primo e menor. Os dois primeiros com aridade 1 e o segundo com aridade 2. Os dois termos usados são: a variável X e a constante 3.

Não existe número primo menor que 3.	$\neg(\exists X\ (primo(X) \wedge menor(X,3)))$

Pré-condição: o domínio é o conjunto dos números naturais. A fórmula usa dois predicados: primo e menor. O primeiro com aridade 1 e o segundo com aridade 2. Os dois termos usados são: a variável X e a constante 3.

3.3 INTERPRETAÇÕES E MODELOS

Como visto no Capítulo 1, fórmulas bem-formadas têm significado apenas quando uma interpretação é dada aos símbolos. Na Lógica Proposicional, uma interpretação consiste em uma atribuição de valores-verdade a átomos. Na Lógica de Predicados, uma vez que existem variáveis envolvidas, para a definição de uma interpretação para uma linguagem λ devem ser especificados:

(1) o domínio;

(2) atribuições a constantes, símbolos funcionais e símbolos predicados que ocorrem na linguagem. Devem ser associados a cada:

- *símbolo constante*, uma entidade no domínio;
- *símbolo funcional*, uma função no domínio;
- *símbolo predicado*, uma correspondente relação no domínio.

Definição 3.22 Uma *interpretação* I de uma *linguagem de primeira ordem*[2] λ consiste em:

1. um conjunto não-vazio D, chamado de *domínio da interpretação*, no qual variáveis irão assumir valores;
2. atribuição a cada:
   - símbolo constante de λ, de um elemento de D;
   - símbolo funcional n-ário de λ, de uma função de $D^n \to D$;
   - símbolo predicado n-ário de λ, de uma função de $D^n \to \{v, f\}$.

Observação 3.9 Cada interpretação especifica um significado para cada símbolo da linguagem. Essas atribuições definem a semântica da linguagem de primeira ordem.

Observação 3.10 Para uma dada interpretação, uma fórmula bem-formada sem variáveis livres representa uma proposição que é verdadeira ou falsa, enquanto uma fórmula bem-formada com variáveis livres representa uma relação no domínio de interpretação que pode ser verdadeira para alguns valores no domínio da variável livre e falsa para outros.

Definição 3.23 Seja I uma interpretação de uma linguagem de primeira ordem λ. Uma *atribuição à variável* (com relação a I) é a atribuição a cada variável em λ de um elemento do domínio de I.

Definição 3.24 Seja I uma interpretação de uma linguagem de primeira ordem λ, com domínio D. Seja A uma atribuição à variável com relação a I. Uma *atribuição a termos* (com relação a I e a A), para os termos de λ é definida da seguinte maneira:

- a cada variável é dada uma atribuição de acordo com A;
- a cada constante é dada uma atribuição de acordo com I;
- se $t_1', \ldots, t_n'$ são as atribuições de termos dos termos $t_1,\ldots,t_n$ e se f' é a atribuição a f, então $f'(t_1',\ldots,t_n') \in D$ é a atribuição de termo de $f(t_1,\ldots,t_n)$.

Definição 3.25 Seja I uma interpretação de uma linguagem de primeira ordem λ, com domínio D. Seja A uma atribuição à variável com relação a I. Uma fórmula em λ pode ser avaliada, isto é, a ela pode ser atribuído o valor-verdade v ou f (com relação a I e a A):

1. Se a fórmula é um átomo $p(t_1,\ldots,t_n)$, então o valor-verdade é obtido calculando o valor de $p'(t_1',\ldots,t_n')$, no qual p' é a função associada a p por I e $t_1',\ldots, t_n'$ são as atribuições de termo dos termos $t_1, \ldots, t_n$, com relação a I e A.

2 Algumas vezes, a definição de uma interpretação pode ser dada com relação a uma fórmula α, em vez de uma linguagem de primeira ordem λ – a definição é a mesma, exceto que λ deve ser substituída por α.

2. Se a fórmula tem a forma $\neg\alpha$, $\alpha \wedge \beta$, $\alpha \vee \beta$, $\alpha \rightarrow \beta$, $\alpha \leftrightarrow \beta$, então seu valor-verdade é dado pela Tabela 3.3.

Tabela 3.3 Valores-verdade de fórmulas compostas usando conectivos lógicos.

α	β	$\neg\alpha$	$\alpha \wedge \beta$	$\alpha \vee \beta$	$\alpha \rightarrow \beta$	$\alpha \leftrightarrow \beta$
v	v	f	v	v	v	v
v	f	f	f	v	f	f
f	v	v	f	v	v	f
f	f	v	f	f	v	v

3. Se a fórmula é da forma $(\exists X\ \alpha)$, seu valor-verdade é v, se existe $d \in D$ tal que α tem valor-verdade v com relação a I e a A(X/d) (A(X/d) é A exceto que a X é atribuído o valor d); caso contrário, seu valor-verdade é f.
4. Se a fórmula é da forma $(\forall X\ \alpha)$, seu valor-verdade é v se, para todo $d \in D$, α tem valor-verdade v com relação a I e a A(X/d) (A(X/d) é A exceto que a X é atribuído o valor d); caso contrário, seu valor-verdade é f.

Observação 3.11 O valor-verdade de uma fórmula fechada não depende de atribuição de variável; conseqüentemente, a atribuição de variável não deve ser levada em consideração no cálculo do valor-verdade de uma fórmula fechada – o valor-verdade de uma fórmula fechada é sempre considerado apenas com relação a uma interpretação.

Definição 3.26 Seja I uma interpretação de uma linguagem de primeira ordem λ e seja α uma fórmula fechada de λ. I é um modelo de α se o valor-verdade de α com relação a I for v.

Exemplo 3.11 Considere uma linguagem λ com os seguintes símbolos:

Símbolo funcional	f/1	g/1	
Símbolo predicado	p/1	q/2	r/1
Símbolo constante	a	b	c
Símbolo variável	X	Y	Z

Seja I a seguinte interpretação para λ:

Domínio: {1,2}

Atribuição a constantes:

a	b	c
1	2	1

Atribuição a variáveis:

X	Y	Z
1	1	2

Atribuição a símbolos funcionais:

f(1)	f(2)	g(1)	g(2)
2	1	1	2

Atribuição a símbolos predicados:

p(1)	p(2)	q(1,1)	q(1,2)	q(2,1)	q(2,2)	r(1)	r(2)
f	v	v	v	v	v	v	f

(a) Usando a interpretação dada, a seguir é determinado o valor-verdade da fórmula:

$$\alpha: (\forall X\ (p(X) \rightarrow q(f(X),a)))$$

Para X = 1

p(X)	→	q(f(X),a)	
p(1)	→	q(f(1),a)	
p(1)	→	q(2,1)	
f	→	v	(avaliada v)

Para X = 2

p(X)	→	q(f(X),a)	
p(2)	→	q(f(2),a)	
p(2)	→	q(1,1)	
v	→	v	(avaliada v)

Uma vez que a fórmula $(p(X) \rightarrow q(f(X),a))$ é v para todos os elementos X do domínio D, α: $(\forall X\ (p(X) \rightarrow q(f(X),a)))$ é v na interpretação I e, conseqüentemente, I é um modelo.

(b) Usando a interpretação dada anteriormente, a seguir é determinado o valor-verdade da fórmula β, que tem Z como variável livre:

$$\beta: (\forall X\ (p(X) \rightarrow (q(f(X),a) \wedge p(Z))))$$

Para X = 1

p(X)	$\rightarrow$	$q(f(X),a) \wedge p(Z)$	
p(1)	$\rightarrow$	$q(f(1),a) \wedge p(2)$	
p(1)	$\rightarrow$	$q(2,1) \wedge v$	
f	$\rightarrow$	$v \wedge v$	
f	$\rightarrow$	v	(avaliada v)

Para X = 2

p(X)	$\rightarrow$	$q(f(X),a) \wedge p(Z)$	
p(2)	$\rightarrow$	$q(f(2),a) \wedge p(2)$	
p(2)	$\rightarrow$	$q(1,1) \wedge v$	
v	$\rightarrow$	$v \wedge v$	
v	$\rightarrow$	v	(avaliada v)

Como a fórmula é avaliada v para todo possível valor de X do domínio, o valor-verdade de β de acordo com a interpretação dada é v. Note que se a atribuição a variáveis considerada fosse tal que:

X	Y	Z
1 ou 2 (tanto faz, dado que X comparece ligada na fórmula β)	1 ou 2 (tanto faz, dado que Y nem comparece na fórmula β)	1

o valor-verdade da fórmula β seria f.

(c) Usando a interpretação dada anteriormente, a seguir é determinado o valor-verdade da fórmula γ.

$$\gamma: (\forall X(\exists Z((p(X) \rightarrow q(Z,X)) \wedge \neg r(Z))))$$

Para X = 1

Z=1	$(p(1) \rightarrow q(1,1)) \wedge \neg r(1)$	(avaliada f)
Z=2	$(p(2) \rightarrow q(2,1)) \wedge \neg r(2)$	(avaliada v) $\Leftarrow$

Para X = 2

Z=1	$(p(2) \rightarrow q(1,2)) \wedge \neg r(1)$	(avaliada f)
Z=2	$(p(2) \rightarrow q(2,2)) \wedge \neg r(2)$	(avaliada v) $\Leftarrow$

A fórmula γ é interpretada como v, uma vez que, para qualquer valor de X, existe um valor de Z que a torna v.

(d) Usando a interpretação dada anteriormente, a seguir é determinado o valor-verdade da fórmula δ.

$$\delta: (\exists Z(\forall X ((p(X) \rightarrow q(Z,X)) \wedge \neg r(Z))))$$

Para Z = 1

X=1	$(p(1) \rightarrow q(1,1)) \wedge \neg r(1)$	(avaliada f)
Se existe um Z que torna a fórmula v, ele não é 1		

Para Z = 2

X=1	$(p(1) \rightarrow q(2,1)) \wedge \neg r(2)$	(avaliada v)
X=2	$(p(2) \rightarrow q(2,2)) \wedge \neg r(2)$	(avaliada v)

A fórmula δ é v na interpretação dada, uma vez que existe Z = 2, tal que para todo valor de X do domínio δ é avaliada v.

(e) Usando a interpretação dada anteriormente, a seguir é determinado o valor-verdade da fórmula ψ.

$$\psi: (\exists Y(\exists X ((p(X) \wedge q(Z,X) \wedge \neg r(Z)))))$$

Para Y = 1

X=1	$p(1) \wedge q(1,1) \wedge \neg r(2)$	(avaliada f)
X=2	$p(2) \wedge q(2,1) \wedge \neg r(2)$	(avaliada v)

A fórmula ψ é v na interpretação dada, uma vez que existe Y (Y=1) e existe X (X=2), que tornam a fórmula v na interpretação dada.

(f) Usando a interpretação dada anteriormente, a seguir é determinado o valor-verdade da fórmula σ.

$$\sigma: (\forall X \forall Y ((q(X,Y) \vee r(X))) \wedge (\exists X ((q(X,a) \rightarrow p(Y)))$$

Considerando que a fórmula σ está no padrão $\alpha \wedge \beta$, tal que:

$$\alpha: (\forall X(\forall Y ((q(X,Y) \vee r(X)))) \text{ e}$$

$$\beta: (\exists X ((q(X,a) \rightarrow p(Y)))$$

a avaliação de σ depende da avaliação de α e de β.

(1) Avaliando α.

Para X = 1

Y=1	$q(1,1) \vee r(1)$	(avaliada v)
Y=2	$q(1,2) \vee r(1)$	(avaliada v)

Para X = 2

Y=1	$q(2,1) \vee r(2)$	(avaliada v)
Y=2	$q(2,2) \vee r(2)$	(avaliada v)

e, portanto, a fórmula α é avaliada v na interpretação dada.

(2) Avaliando β.

X=1	$q(1,1) \rightarrow p(1)$	(avaliada f)
X=2	$q(2,1) \rightarrow p(1)$	(avaliada f)

e, portanto, a fórmula β é avaliada f na interpretação I, uma vez que não existe valor de X que torna a fórmula v. Dado que α e β são avaliadas v e f, respectivamente, a fórmula σ é avaliada f em I.

Exemplo 3.12 Considere uma linguagem λ que só tem o símbolo funcional p/2 e por variáveis, os símbolos: {X,Y,Z}. Os símbolos constante e funcional não comparecem na especificação de linguagem. Considere, ainda, uma interpretação que tem por domínio D = {a,b,c} e associa ao predicado p uma função (também nomeada p: D $\rightarrow$ {v,f}), como segue:

p(a,a)	p(a,b)	p(a,c)	p(b,a)	p(b,b)	p(b,c)	p(c,a)	p(c,b)	p(c,c)
v	f	v	f	v	v	f	v	v

(a) Determinar o valor-verdade da fórmula $(\forall X(\exists Y\ p(X,Y)))$.

X = a	Y = a	p(a,a) = v	Para X=a, existe Y=a que torna a fórmula v.
X = b	Y = a	p(b,a) = f	• Para X=b, Y=a torna a fórmula f e, portanto, uma outra instanciação para a variável Y é tentada.
	Y = b	p(b,b) = v	• Para X=b e Y=b a fórmula p(b,b) = v.
X = c	Y = a	p(c,a) = f	• Para X=c, Y=a torna a fórmula f e, portanto, uma outra instanciação para a variável Y é tentada.
	Y = b	p(c,b) = v	• Para X=c e Y=b a fórmula p(c,b) = v.

Como a fórmula $\exists Y\ p(a,Y)$ é avaliada v, a fórmula $\exists Y\ p(b,Y)$ é também avaliada v e a $\exists Y\ p(c,Y)$ também v, a fórmula $(\forall X(\exists Y\ p(X,Y)))$ é avaliada v na interpretação dada.

(b) Determinar o valor-verdade da fórmula ($\forall$Y p(Y,b)).

Y = a	p(a,b) = f	Para Y=a, p(a,b) é f.

($\forall$Y p(Y,b)) é f uma vez que p(a.b) = f.

(c) Determinar o valor-verdade da fórmula ($\forall$Y p(Y,Y)).

Y = a	p(a,a) = v
Y = b	p(b,b)= v
Y = c	p(c,c) = v

Como p(a,a), p(b,b) e p(c,c) são interpretadas v, a fórmula ($\forall$Y p(Y,Y))) é interpretada v.

(d) Determinar o valor-verdade da fórmula ($\exists$X($\exists$Y p(X,Y)).

X = a	Y = a	p(a,a) = v	Qualquer outra entre as seis possíveis combinações que tornam a fórmula v poderia ser considerada também.

Como p(a,a) = v, a fórmula ($\exists$X($\exists$Y p(X,Y)) é avaliada v.

(e) Determinar o valor-verdade da fórmula ($\forall$Y p(Y,Y)) $\wedge$ ($\exists$X($\forall$Y p(X,Y))).

Note que ($\forall$Y p(Y,Y)) foi avaliada v na parte (b) deste exemplo. Resta, pois, avaliar a subfórmula ($\exists$X($\forall$Y p(X,Y))).

X = a	Y = a Y = b	p(a,a) = v p(a,b) = f	Se existir uma instanciação da variável X que torna a fórmula v, certamente essa instanciação não é com a constante a, dado que, para X instanciado com a, a fórmula ($\forall$Y p(a,Y)) é f, uma vez que p(a,b) = f.
X = b	Y = a	p(b,a)= f	Vale o mesmo comentário anterior, dado que p(b,a)=f.
X = c	Y = a	p(c,a) = f	Vale o mesmo comentário anterior, dado que p(c,a)=f.

Como ($\forall$Y p(a,Y)) = f, ($\forall$Y p(a,Y)) = f e ($\forall$Y p(a,Y)) = f, a fórmula ($\exists$X($\forall$Y p(X,Y))) = , o que faz com que a avaliação da fórmula ($\forall$Y p(Y,Y)) $\wedge$ ($\exists$X($\forall$Y p(X,Y))) seja f.

Observação 3.12 Os axiomas de uma teoria de primeira ordem são um subconjunto de fórmulas fechadas da linguagem da teoria.

Definição 3.27 Seja τ uma teoria de primeira ordem e seja λ a linguagem de τ. Um modelo para τ é uma interpretação para λ, a qual é um modelo para cada axioma de τ.

Definição 3.28 Seja S um conjunto de fórmulas fechadas de uma linguagem de primeira ordem λ e seja I uma interpretação de λ. I é um modelo para S, se I for um modelo para cada fórmula de S.

Observação 3.13 Note que se $S = \{F_1,...,F_n\}$ é um conjunto finito de fórmulas fechadas, então I é um modelo para S se e somente se I for um modelo para $F_1 \wedge ... \wedge F_n$.

Definição 3.29 Seja S um conjunto de fórmulas fechadas de uma linguagem de primeira ordem λ:

- S é satisfatível em λ se λ tem uma interpretação que é um modelo para S;
- S é válida se toda interpretação de λ for um modelo para S;
- S é insatisfatível se não tem modelos (ou seja, não existe interpretação de λ que seja um modelo para S).

Definição 3.30 Seja S um conjunto de fórmulas fechadas de uma linguagem de primeira ordem λ. Seja F uma fórmula fechada de λ. Diz-se que *F é conseqüência lógica de S* se, para toda interpretação I de λ, I for um modelo para S, I é também um modelo para F. Se $S = \{F_1,...,F_n\}$ for o conjunto de fórmulas fechadas, então a fórmula fechada F é conseqüência lógica de S se e somente se a fórmula

$$F_1 \wedge ... \wedge F_n \rightarrow F \text{ for válida.}$$

Teorema 3.1 Seja S um conjunto de fórmulas fechadas de uma linguagem de primeira ordem λ. Seja α uma fórmula fechada de λ. Então

F é conseqüência lógica de S se e somente se $S \cup \{\neg\alpha\}$ for insatisfatível.

Prova:

- Suponha que α seja conseqüência lógica de S. Seja I uma interpretação de λ e suponha que I seja um modelo para S. Então, I é também um modelo para F. Portanto, I não é um modelo para $S \cup \{\neg\alpha\}$. Assim, $S \cup \{\neg\alpha\}$ é insatisfatível.
- Suponha agora que $S \cup \{\neg\alpha\}$ seja insatisfatível. Seja I uma interpretação de λ. Suponha que I seja um modelo para S. Desde que $S \cup \{\neg\alpha\}$ seja insatisfatível, I não pode ser um modelo para $\neg\alpha$, o que torna I um modelo para α, ou seja, α é conseqüência lógica de S.

Exemplo 3.13 Seja $S = \{p(a), (\forall X(p(X) \rightarrow q(X)))\}$ e seja α a fórmula q(a). Seja I um modelo para S. Assim, p(a) é verdade com relação a I. Segue disso e da verdade de $(\forall X(p(X) \rightarrow q(X)))$ em I que q(a) deve ser v com relação a I.

Observação 3.14 Como discutido, a conseqüência lógica pode ser reduzida ao teste de insatisfatibilidade de uma fórmula na forma conjuntiva. Em Lógica Proposicional é possível enumerar todas as atribuições de valores-verdade para testar insatisfatibilidade. Em Lógica de Predicados, entretanto, será necessário enumerar todas as estruturas para usar essa abordagem, o que a torna inexeqüível, uma vez que existem infinitas dessas estruturas. Felizmente, entretanto, existe uma maneira de decidir sobre insatisfatibilidade por meio do exame de apenas uma estrutura. Para cada fórmula existe um domínio especial – chamado de universo de Herbrand – e uma função especial, com base na interpretação de Herbrand, tal que apenas estruturas usando esse domínio e essa função devem ser consideradas. Isso é discutido em detalhe no Capítulo 5.

4. SUBSTITUIÇÃO, UNIFICAÇÃO E FORMA NORMAL

Freqüentemente, quando se lida com conceitos lógicos e particularmente na prova de teoremas que envolvem fórmulas quantificadas, é necessário fazer o "casamento" de certas subexpressões. Encontrar substituições de variáveis por termos de maneira a tornar as subexpressões idênticas é um processo extremamente importante ~~em Inteligência~~ Artificial ~~e fundamenta~~ qualquer raciocínio representado por expressões lógicas de primeira ordem. Esse processo é denominado *unificação* e fundamenta-se no conceito de substituição, que troca variáveis por termos e que é definido na Definição 4.2.

4.1 O PROCESSO DE SUBSTITUIÇÃO

Definição 4.1 Uma *expressão* E é um termo, ou um literal ou uma conjunção ou disjunção de literais. Uma expressão simples é um termo ou um literal.

Definição 4.2 Uma *substituição* é um conjunto finito da forma:

$$\theta = \{t_1/v_1, t_2/v_2, \ldots, t_n/v_n\}$$

no qual cada v_i ($1 \leq i \leq n$) é uma variável, cada t_i ($1 \leq i \leq n$) é um termo diferente de v_i e não existem dois elementos no conjunto θ que tenham a mesma variável após a barra (*propriedade da funcionalidade*).

Observação 4.1

1. quando $t_1,\ldots, t_n$ são termos *ground*, a substituição é chamada de substituição *ground*;
2. uma substituição que consiste em nenhum elemento é chamada de substituição vazia (ou substituição identidade) e é notada por ε;
3. geralmente, substituições são representadas por letras gregas;
4. a restrição de θ ao conjunto de variáveis V é a substituição $\{t/v \in \theta \mid v \in V\}$.

Observação 4.2 Uma substituição $\theta = \{t_1/v_1, t_2/v_2, \ldots, t_n/v_n\}$ pode ser abordada como uma função do conjunto de variáveis de uma linguagem, no conjunto de termos da linguagem, da seguinte maneira:

$$\theta(x) = \begin{cases} t_i & \text{se } x = v_i \\ x & \text{caso contrário} \end{cases}$$

Definição 4.3 Seja $\theta = \{t_1/v_1, t_2/v_2, \ldots, t_n/v_n\}$ uma substituição. O conjunto $\{v_1, v_2, \ldots, v_n\}$ é chamado de *domínio* de θ (dom(θ)) e o conjunto $\{t_1, t_2, \ldots, t_n\}$ é chamado de *contradomínio* de θ.

Definição 4.4 Seja $\theta = \{t_1/v_1, t_2/v_2,\ldots,t_n/v_n\}$ uma substituição e seja E uma expressão. Então, $E\theta$ é uma expressão obtida a partir de E substituindo simultaneamente cada ocorrência da variável v_i ($1 \leq i \leq n$) em E pelo termo t_i. O resultado da operação $E\theta$ é chamado de instanciação de E por θ.

Exemplo 4.1 Uma substituição troca variáveis por termos. Por exemplo, na cláusula $p(X) \vee q(X)$, a variável X pode ser substituída pelo termo f(a), e uma nova cláusula $p(f(a)) \vee q(f(a))$ é obtida. Considerando que cláusulas são universalmente quantificadas, pode-se dizer que esta substituição torna a cláusula "menos geral". Note, entretanto, que a segunda cláusula é uma conseqüência lógica da primeira, ou seja:

$$p(X) \vee q(X) \models p(f(a)) \vee q(f(a)).$$

Exemplo 4.2 Considere $E = p(Y,f(X))$ e a substituição $\theta = \{a/X, g(g(X))/Y\}$. A instanciação de E por θ é dada por $E\theta = p(g(g(X)),f(a))$. Note que X e Y são simultaneamente substituídas. Considerando agora a substituição $\sigma = \{f(a)/X,b/Y\}$, tem-se $E\sigma = p(b,f(f(a)))$.

Exemplo 4.3 A Tabela 4.1 mostra quatro instanciações do literal p(X,f(Y),b) obtidas usando quatro substituições diferentes.

Tabela 4.1 Quatro instanciações do literal p(X,f(Y),b).

Substituição	Instanciação resultante
{Z/X,W/Y}	p(Z,f(W),b)
{a/Y}	p(X,f(a),b)
{g(Z)/X,a/Y}	p(g(Z),f(a),b)
{c/X,a/Y}	p(c,f(a),b)

Observação 4.3 É importante que qualquer substituição θ seja *idempotente*. Isso significa que, para qualquer expressão E, θ deve ser tal que $E\theta = (E\theta)\theta$. Isso, por sua vez, exige que para qualquer substituição $\{t_1/v_1, t_2/v_2,\ldots,t_n/v_n\}$ nenhuma das variáveis v_i deve ocorrer em qualquer dos termos t_j.

Exemplo 4.4 Considere a expressão $E = p(X,Y)$ e a substituição $\sigma_1 = \{Z/X, a/Y\}$. Então, $E\sigma_1 = p(X,Y)\{Z/X, a/Y\} = p(Z,a)$. Aplicando novamente a substituição ao resultado obtido, tem-se $(E\sigma_1)\sigma_1 = p(Z,a)\sigma_1 = p(Z,a)\{Z/X, a/Y\} = p(Z,a) = E\sigma_1$. Portanto, a substituição σ_1 é idempotente.

Exemplo 4.5 Considere a mesma expressão anterior $E = p(X,Y)$ e a substituição $\sigma_2 = \{Y/X, a/Y\}$. Então, $E\sigma_2 = p(X,Y)\{Y/X, a/Y\} = p(Y,a)$. Aplicando novamente a substituição ao resultado obtido, tem-se $(E\sigma_2)\sigma_2 = p(Y,a)\sigma_2 = p(Y,a)\{Y/X, a/Y\} = p(a,a)$. A substituição σ_2 não é idempotente uma vez que $E\sigma_2 \neq (E\sigma_2)\sigma_2$.

Definição 4.5 Seja $\theta_1 = \{t_1/x_1, t_2/x_2,\ldots,t_n/x_n\}$ e $\theta = \{u_1/y_1, u_2/y_2,\ldots,u_m/y_m\}$. A *composição* de θ_1 e θ é a substituição notada por $\theta_1\theta$, que é obtida a partir do conjunto $\{t_1\theta/x_1, t_2\theta/x_2,\ldots,t_n\theta/x_n,u_1/y_1, u_2/y_2,\ldots,u_m/y_m\}$, por meio da eliminação de qualquer elemento $t_j\theta/x_j$ para o qual $t_j\theta = x_j$, e qualquer elemento u_i/y_i, tal que y_i está em $\{x_1,x_2,\ldots,x_n\}$.

Exemplo 4.6 Considere as substituições $\theta_1 = \{p(X,Y)/Z\}$ e $\theta = \{a/X, b/Y, c/W, d/Z\}$. Tem-se, então, $\theta_1\theta = \{p(a,b)/Z,a/X,b/Y,c/W\}$. A Tabela 4.2 mostra mais algumas composições de substituições.

Tabela 4.2 Composição de substituições.

Substituição θ	Substituição θ_1	Substituição composta $\theta\theta_1$
$\{f(Z)/Y\}$	$\{b/X, a/Y, a/Z\}$	$\{f(Z)\theta_1/Y,b/X, a/Y, a/Z\} = \{f(a)/Y, b/X, a/Z\}$
$\{f(Y)/X,U/Z\}$	$\{b/Y, Z/U\}$	$\{f(Y)\theta_1/X,U\theta_1/Z, b/Y, Z/U\} =$ $\{f(b)/X, Z/Z, b/Y, Z/U\} = \{f(b)/X,b/Y, Z/U\}$
$\{Y/X\}$	$\{a/X,a/Y\}$	$\{Y\theta_1/X,a/X,a/Y\} = \{a/X,a/Y\}$
$\{p(X,Y)/Z\}$	$\{a/X,b/Y,c/W,d/Z\}$	$\{p(X,Y)\theta_1/Z,a/X,b/Y,c/W,d/Z\} =$ $\{p(a,b)/Z,a/X,b/Y,c/W\}$

Exemplo 4.7 Considere a expressão $W = p(U1,g(V1),V2)$ e as substituições $\theta_1 = \{f(V1)/U1\}$ e $\theta_2 = \{a/V1, b/V2, c/U1\}$.

(a) $W\theta_1 = p(U1,g(V1),V2)\theta_1 = p(f(V1),g(V1),V2)$

$(W\theta_1)\theta_2 = p(f(V1),g(V1),V2)\theta_2 = p(f(a),g(a),b)$

(b) $W\theta_2 = p(U1,g(V1),V2)\ \theta_2 = p(c,g(a),b)$

$(W\theta_2)\theta_1 = p(c,g(a),b)\theta_1 = p(c,g(a),b)$

(c) $\theta_1\theta_2 = \{f(V1)\theta_2/U1, a/V1, b/V2, c/U1\} = \{f(a)/U1,a/V1,b/V2\}$

(d) $W(\theta_1\theta_2) = p(U1,g(V1),V2)\{f(a)/U1,a/V1,b/V2\} = p(f(a),g(a),b) = (W\theta_1)\theta_2$ (ver (a)).

(e) $\theta_2\theta_1 = \{a\theta_1/V1, b\theta_1/V2, c\theta_1/U1,f(V1)/U1\} = \{a/V1, b/V2, c/U1\}$

(f) $W(\theta_2\theta_1) = p(U1,g(V1),V2)\{a/V1, b/V2, c/U1\} = p(c,g(a),b) = (W\theta_2)\theta_1$ (ver (b)).

Observação 4.4 Pode ser mostrado que aplicando θ_1 e θ sucessivamente a uma expressão E é o mesmo que aplicar a composição $\theta_1\theta$ a E, ou seja,

$$(E\theta_1)\theta = E(\theta_1\theta)$$

O Exemplo 4.7 (a) e (d), bem como (b) e (e), mostram isso.

Observação 4.5 Pode ser mostrado que a composição de substituições é associativa, ou seja:

$$(\theta_1\theta)\theta_2 = \theta_1(\theta\theta_2).$$

No procedimento de prova de teoremas por resolução é comum ter de unificar duas ou mais expressões. A próxima definição contempla essa situação.

Observação 4.6 A composição de duas substituições que satisfazem a propriedade de funcionalidade satisfará a propriedade de funcionalidade também.

Definição 4.6 Uma substituição θ é chamada de *unificadora* do conjunto de expressões {E1, E2,..., Ek} se e somente se:

$$E_1\theta = E_2\theta = ...= E_k\theta.$$

O conjunto $\{E_1, E_2, ..., E_k\}$ é dito *unificável* se existir uma substituição unificadora para ele.

Exemplo 4.8 Considere as expressões $E_1 = p(X,f(Y),b)$ e $E_2 = p(X,f(b),b)$. A substituição $\theta = \{a/X,b/Y\}$ é unificadora do conjunto $\{E_1, E_2\}$, uma vez que:

$$E_1\theta = E_2\theta = p(a,f(b),b).$$

Exemplo 4.9 Considere as expressões $E_1 = p(a,Y)$ e $E_2 = p(X,f(b))$. O conjunto $\{E_1, E_2\}$ é unificável. A substituição $\theta = \{a/X,f(b)/Y\}$ é uma unificadora do conjunto, uma vez que: $E_1\theta = E_2\theta = p(a,f(b))$, ou seja, $\{E_1,E_2\}\theta = \{p(a,f(b))\}$.

Observação 4.7 No Exemplo 4.5, embora $\theta = \{a/X,b/Y\}$ seja um unificador do conjunto $\{p(X,f(Y),b), p(X,f(b),b)\}$, não é o unificador mais geral. A variável X, por exemplo, não precisava ser substituída pelo termo constante a para tornar as duas expressões unificáveis. A próxima definição estabelece o critério que caracteriza a substituição chamada de unificadora mais geral.

Definição 4.7 Uma substituição unificadora δ do conjunto de expressões $\{E_1, E_2, ..., E_k\}$ é a *unificadora mais geral* – mgu[1] – se e somente se para cada unificadora θ do conjunto existir uma substituição γ, tal que θ = δγ. Duas ou mais expressões são unificáveis se elas têm a mesma mgu.

Observação 4.8 Parafraseando a Definição 4.6, a mgu δ de $\{E_i\}$ tem a propriedade que se θ é qualquer unificadora de $\{E_i\}$ produzindo $\{E_i\}\theta$, então existe uma substituição γ, tal que $\{E_i\}\theta = \{E_i\}\delta\gamma$.

Exemplo 4.10 Considere novamente o conjunto de expressões do Exemplo 4.8, $\{p(X,f(Y),b), p(X,f(b),b)\}$.

- A mgu do conjunto de expressões é $\delta = \{b/Y\}$ e $\{p(X,f(Y),b), p(X,f(b),b)\}\delta = \{p(X,f(b),b), p(X,f(b),b)\} = \{p(X,f(b),b)\}$.
- Considere uma substituição unificadora $\theta = \{a/X, b/Y\}$. Tem-se: $\{p(X,f(Y),b), p(X,f(b),b)\}$

1 Decidiu-se por manter a identificação mgu (*most general unifier*) para referenciar a substituição que é a unificadora mais geral de um conjunto de expressões.

{a/X, b/Y} = {p(a,f(b),b), p(a,f(b),b)}} = {p(a,f(b),b)}.

- Para a substituição θ, $\exists\gamma$ = {a/X} tal que a composição $\delta\gamma$ = {b/Y,a/X} e, portanto, {p(X,f(Y),b), p(X,f(b),b)}$\delta\gamma$ = {p(X,f(Y),b), p(X,f(b),b)} {b/Y,a/X} = {p(a,f(b),b), p(a,f(b),b)}} = {p(a,f(b),b)}.

Definição 4.8 Sejam C_1 e C_2 duas fórmulas bem-formadas e seja θ a substituição $\{t_1/v_1, ..., t_n/v_n\}$, na qual, para $i \neq j$, $t_i \neq t_j$. Diz-se que $C_1\theta$ e $C_2\theta$ são padronizadas à parte sempre que não existir variável que ocorra em ambas $C_1\theta$ e $C_2\theta$.

4.2 O PROCESSO DE UNIFICAÇÃO

Existem muitos algoritmos que podem ser usados para unificar um conjunto finito de expressões unificáveis e que sinalizam uma falha quando o conjunto não pode ser unificado. O procedimento recursivo descrito no Procedimento 1, encontrado em Nilsson (1980), é útil para o estabelecimento da idéia geral de como unificar duas expressões estruturadas em lista, que é o formato de entrada esperado pelo algoritmo (e.g., o literal p(X,f(a,Y)) é representado como (p X (f a Y)).

Procedimento 4.1 Procedimento para a unificação de duas expressões E1 e E2. Se as expressões são unificáveis, o procedimento retorna a *mgu* das duas expressões; se não forem unificáveis, o procedimento falha.

```
recursive procedure unify(E1,E2)
if either E1 or E2 is an atom (that is, a predicate symbol, a function symbol, a
constant symbol, a negation symbol or a variable), interchange the arguments E1
and E2 (if necessary) so that E1 is an atom, and do:
begin
 if E1 and E2 are identical
      then return nil
      else
        if E1 is a variable
          then
               if E1 occurs in E2
                 then return fail
                 else return {E2/E1}
          else
               if E2 is a variable
                    then return {E1/E2}
                    else return fail
 F1 ← the first element of E1; T1 ← the rest of E1
 F2 ← the first element of E2; T2 ← the rest of E2
 Z1 ← unify(F1,F2)
 if Z1 = fail
      then return fail
      else
        begin
             G1 ← result of applying Z1 to T1
             G2 ← result of applying Z1 to T2
             Z2 ← unify(G1,G2)
             if Z2 = fail
                    then return fail
                    else return the composition of Z1 and Z2
        end
end
```

Observação 4.9 Na descrição do Procedimento 4.1 optou-se por manter o pseudocódigo original, como descrito em Nilsson (1972). Note que o teste de ocorrência de variável é apenas feito com relação a E1 e E2, e não vice-versa, e, por essa razão, se E2 for variável e E1 foi um termo, é preciso que os dois argumentos troquem entre si suas posições. O Exemplo 4.14 trata dessa situação.

4.2.1 Exemplos do processo de unificação – o procedimento UNIFY

Exemplo 4.11 Considere a interrogação Prolog e a respectiva instanciação de variáveis subseqüente, provocada pela prova:

?- f(X,b,Y) = f(a,b,c).
X = a
Y = c

Segue o "*trace*" do algoritmo descrito no Procedimento 4.1, que faz a unificação de dois termos, retornando a substituição unificadora (mgu) entre eles (caso a unificação seja possível, o que, no exemplo em questão, já se sabe que é, em razão de a interrogação Prolog anterior ter sido bem-sucedida).

E1 E2
unify((f X b Y), (f a b c)) → retorna {a/X,b/Y} (***substituição final***)
F1 T1 F2 T2

Z1 ← unify (f,f) ⇒ procedimento bem-sucedido e retorna nil → nil{a/X,c/Y} = {a/X,c/Y}

E1 E2
Z2 ← unify ((X b Y), (a b c))
F1 T1 F2 T2

composição das duas substituições

Z1 ← unify (X,a) ⇒ procedimento bem-sucedido e retorna {a/X} → {a/X}{c/Y} = {a/X,c/Y}
G1 ← (b Y){a/X} ⇒ (b Y)
G2 ← (b c){a/X} ⇒ (b c)

composição das duas substituições

E1 E2
Z2 ← unify ((b Y), (b c))
F1T1 F2T2

Z1 ← unify (b , b) ⇒ procedimento bem-sucedido e retorna nil → nil{c/Y} = {c/Y}

G1 ← (Y)nil ⇒ (Y)
G2 ← (c)nil ⇒ (c)

composição das duas substituições

Z2 ← unify ((Y) , (c)) procedimento bem-sucedido e retorna substituição {c/Y}

Z1 ← unify(Y , c) ⇒ procedimento bem-sucedido e retorna {c/Y}
Z2 ← unify(( ), ( ))⇒ procedimento bem-sucedido e retorna nil →

composição das duas substituições

Exemplo 4.12 Considere a interrogação Prolog e a respectiva instanciação subseqüente de variáveis, provocada pela prova:

?- p(M,a,h(X,b)) = p(N,Y,h(Z,R)).

M = _0084

X = _0098

N = _0084

Y = a

Z = _0098

R = b

Segue o "*trace*" do algoritmo descrito em Procedimento 4.1, que faz a unificação das duas expressões, retornando a substituição unificadora (*mgu*) entre elas (dado que a unificação é bem-sucedida, como visto na interrogação Prolog anterior). Com o objetivo de manter o "*trace*" legível, os vários níveis da recursão foram enumerados.

$$\text{unify}(\overbrace{(\underset{\substack{\uparrow \\ F1}}{p}\ \underbrace{M\ a\ (h\ X\ b)}_{T1})}^{E1},\ \overbrace{(\underset{\substack{\uparrow \\ F2}}{p}\ \underbrace{N\ Y\ (h\ Z\ R)}_{T2})}^{E2})$$

(a.1)

Z1 ← unify (p,p) ⇒ nil

(a.2)

$$Z2 \leftarrow \text{unify}\ (\overbrace{(\underset{\substack{\uparrow \\ F1}}{M}\ \underbrace{a\ (h\ X\ b)}_{T1})}^{E1},\ \overbrace{(\underset{\substack{\uparrow \\ F2}}{N}\ \underbrace{Y\ (h\ Z\ R)}_{T2})}^{E2})$$

(a.2.1)

Z1 ← unify (M,N) ⇒ {N/M}

G1 ← (a (h X b)){N/M} ⇒ (a (h X b))

G2 ← (Y (h Z R)){N/M} ⇒ (Y (h Z R))

(a.2.2)

$$Z2 \leftarrow \text{unify}(\overbrace{(\underset{\substack{\uparrow \\ F1}}{a}\ \underbrace{(h\ X\ b)}_{T1})}^{E1},\ \overbrace{(\underset{\substack{\uparrow \\ F2}}{Y}\ \underbrace{(h\ Z\ R)}_{T2})}^{E2})$$

(a.2.2.1)

$Z1 \leftarrow \text{unify}\ (a\ ,Y) \Rightarrow \{a/Y\}$

$G1 \leftarrow (\ (h\ X\ b))\{a/Y\} \Rightarrow (\ (h\ X\ b)\ )$

$G2 \leftarrow (\ (h\ Z\ R)\ )\{a/Y\} \Rightarrow (\ (h\ Z\ R)\ )$

(a.2.2.2)

$Z2 \leftarrow \text{unify}\ (\ \underbrace{(h\ X\ b)}_{F1}\ ),\ (\ \underbrace{(h\ Z\ R)}_{F2}\ )$ (ambos, T1 e T2, são listas vazias, i.e., ( ))

(a.2.2.2.1)

$Z1 \leftarrow \text{unify}\ (\ (\underset{F1}{\underset{\uparrow}{h}}\ \underbrace{X\ b}_{T1})\ ,(\underset{F2}{\underset{\uparrow}{h}}\ \underbrace{Z\ R}_{T2})\ )$

(a.2.2.2.1.1)

$Z1 \leftarrow \text{unify}\ (h,h) \Rightarrow nil$

(a.2.2.2.1.2)

$Z2 \leftarrow \text{unify}\ ((\underset{F1}{\underset{\uparrow}{X}}\ \underset{T1}{\underset{\uparrow}{b}})\ ,(\underset{F2}{\underset{\uparrow}{Z}}\ \underset{T2}{\underset{\uparrow}{R}})\ )$

(a.2.2.2.1.2.1)

$Z1 \leftarrow \text{unify}\ (X\ ,Z) \Rightarrow \{Z/X\}$

(a.2.2.2.1.2.2)

$Z2 \leftarrow \text{unify}\ (\ (\underset{F1}{\underset{\uparrow}{b}})\ ,(\underset{F2}{\underset{\uparrow}{R}})\ )$ (ambos, T1 e T2, são listas vazias, i.e., ( ))

(a.2.2.2.1.2.2.1)

$Z1 \leftarrow \text{unify}(b,R) \Rightarrow \{b/R\}$

(a.2.2.2.1.2.2.1)

$Z2 \leftarrow \text{unify}\ (\ (\ ),(\ )\ ) \Rightarrow nil$

(a.2.2.2.2)

$Z2 \leftarrow \text{unify}\ (\ (\ ),(\ )\ ) \Rightarrow nil$

Compondo as substituições para a obtenção da substituição *mgu* final:

1. (a.2.2.2.1.2.2.1) com (a.2.2.2.1.2.2.1) $\Rightarrow$ {b/R}nil = {b/R}, que é a substituição que retorna da unificação (a.2.2.2.1.2.2);

2. (a.2.2.2.1.2.1) com (a.2.2.2.1.2.2) ⇒ {Z/X}{b/R} = {Z/X,b/R}, que é a substituição que retorna da unificação (a.2.2.2.1.2);
3. (a.2.2.2.1.1) com (a.2.2.2.1.2) ⇒ nil{Z/X,b/R} = {Z/X,b/R}, que é a substituição que retorna da unificação (a.2.2.2.1);
4. (a.2.2.2.1) com (a.2.2.2.2) ⇒ {Z/X,b/R}nil = {Z/X,b/R}, que é a substituição que retorna da unificação (a.2.2.2);
5. (a.2.2.1) com (a.2.2.2) ⇒ {a/Y}{z/X,b/R} = {a/Y,z/X,b/R}, que é a substituição que retorna da unificação (a.2.2);
6. (a.2.1) com (a.2.2) ⇒ {N/M}{a/Y,z/X,b/R} = {N/M,a/Y,z/X,b/R}, que é a substituição que retorna da unificação (a.2);
7. (a.1) com (a.2) ⇒ nil{N/M,a/Y,z/X,b/R} = {N/M,a/Y,z/X,b/R}, que é a substituição final do processo de unificação das expressões dadas. Compare essa substituição com os resultados exibidos pela interrogação Prolog mostrada anteriormente.

No "*trace*" anterior, as expressões associadas às variáveis G1 e G2 foram, muitas vezes, omitidas para simplificação.

Exemplo 4.13 Considere a interrogação Prolog e a respectiva instanciação subseqüente de variáveis, provocada pela prova:

?- p(M,K,c) = p(Z,f(a),M).

M = c

K = f(a)

Z = c

Segue o "*trace*" do algoritmo descrito em Procedimento 4.1, que faz a unificação das duas expressões, retornando a substituição unificadora (*mgu*) entre elas (dado que a unificação é bem-sucedida, como visto na interrogação Prolog anterior).

$$\text{unify}(\ \overbrace{(\underset{\substack{\uparrow\\ \text{F1}}}{p}\ \underbrace{Y\ M\ c}_{\text{T1}})}^{\text{E1}},\ \overbrace{(\underset{\substack{\uparrow\\ \text{F2}}}{p}\ \underbrace{Z\ (f\ a)\ M}_{\text{T2}})}^{\text{E2}}\ )$$

(a.1)

Z1 ← unify (p,p) ⇒ nil

(a.2)

$$Z2 \leftarrow \text{unify}\ (\ \overbrace{(\underset{\substack{\uparrow\\ F1}}{M}\ \underbrace{K\ c}_{T1})}^{E1},\ \overbrace{(\underset{\substack{\uparrow\\ F2}}{Z}\ \underbrace{(f\ a)\ M}_{T2})}^{E2}\ )$$

(a.2.1)

$Z1 \leftarrow \text{unify}\ (M,Z) \Rightarrow \{Z/M\}$

$G1 \leftarrow (K\ c)\{Z/M\} \Rightarrow (K\ c)$

$G2 \leftarrow (\ (f\ a)\ M)\{Z/M\} \Rightarrow (\ (f\ a)\ Z)$

(a.2.2)

$$Z2 \leftarrow \text{unify}(\ \overbrace{(\underset{\substack{\uparrow\\ F1}}{K}\ \underset{\substack{\uparrow\\ T1}}{c})}^{E1},\ \overbrace{(\ \underset{\substack{\uparrow\\ F2}}{(f\ a)}\ \underset{\substack{\uparrow\\ T2}}{Z})}^{E2}\ )$$

(a.2.2.1)

$Z1 \leftarrow \text{unify}\ (K\ ,\ (f\ a)) \Rightarrow \{(f\ a)/K\}$

$G1 \leftarrow (c)\{(f\ a)/K\} \Rightarrow (c)$

$G2 \leftarrow (Z)\{(f\ a)/K\} \Rightarrow (Z)$

(a.2.2.2)

$Z2 \leftarrow \text{unify}\ (\ (\underset{\substack{\uparrow\\ F1}}{c})\ ,\ (\underset{\substack{\uparrow\\ F2}}{Z})\ )$ (ambos, T1 e T2, são listas vazias, i.e., ( ))

(a.2.2.2.1)

$Z1 \leftarrow \text{unify}\ (c\ ,\ Z) \Rightarrow \{c/Z\}$

(a.2.2.2.2)

$Z2 \leftarrow \text{unify}\ (\ (\ )\ ,\ (\ )\ ) \Rightarrow \text{nil}$

Compondo as substituições para a obtenção da *mgu* final:

1. (a.2.2.2.1) com (a.2.2.2.2) $\Rightarrow$ {c/Z}nil = {c/Z}, que é a substituição que retorna da unificação (a.2.2.2);
2. (a.2.2.1) com (a.2.2.2) $\Rightarrow$ {(f a)/K}{c/Z} = {(f a)/K, c/Z}, que é a substituição que retorna da unificação (a.2.2);
3. (a.2.1) com (a.2.2) $\Rightarrow$ {Z/M}{(f a)/K, c/Z} = {c/M,(f a)/K,c/Z}, que é a substituição que retorna da unificação (a.2). Note que o resultado da composição de duas substituições é a

substituição que resulta da aplicação da segunda substituição aos termos da primeira, seguindo o estabelecido na Definição 4.5 deste *Apontamento*;

4. (a.1) com (a.2) ⇒ nil{c/M,(f a)/K,c/Z} = {c/M,(f a)/K,c/Z}, que é a substituição mgu final.

Exemplo 4.14 Considere a interrogação Prolog e a respectiva instanciação subseqüente de variáveis, provocada pela prova:

?- p(g(a)) = p(X).

X = g(a)

Segue o "*trace*" do algoritmo descrito no Procedimento 4.1, que faz a unificação das duas expressões, retornando a substituição unificadora (*mgu*) entre elas (dado que a unificação é bem-sucedida, como visto na interrogação Prolog anterior).

$$\text{unify}(\overbrace{(\underset{\substack{\uparrow\\ F1}}{p}\ \underbrace{(g\ a)}_{T1})}^{E1},\ \overbrace{(\underset{\substack{\uparrow\\ F2}}{p}\ \underset{\substack{\uparrow\\ T2}}{X})}^{E2})$$

(a.1)

Z1 ← unify (p,p) ⇒ nil

(a.2)

$$Z2 \leftarrow \text{unify}\ (\overbrace{(\underbrace{(g\ a)}_{F1})}^{E1},\ \overbrace{(\underset{\substack{\uparrow\\ F2}}{X})}^{E2})$$ (ambos, T1 e T2, são listas vazias, i.e., ( ))

(a.2.1)

Z1 ← unify ((g a), X) ⇒ unify (X , (g a)) ⇒ {(g a)/X}

(a.2.2)

Z2 ← unify(( ) , ( )) ⇒ nil

Compondo as substituições para a obtenção da *mgu* final:

1. (a.2.1) com (a.2.2) ⇒ {(g a)/X}*nil* = {(g a)/X}, que é a substituição que retorna da unificação (a.2);
2. (a.1) com (a.2) ⇒ nil{(g a)/X} = {(g a)/X}, que é a substituição mgu final.

4.2.2 Implementação Prolog do algoritmo UNIFY

A seguir, é apresentado um metaprograma escrito em Prolog que implementa o processo de unificação realizado pela linguagem Prolog (por meio do operador =).

```
% unificando duas variaveis
unify(X,Y):-
var(X),
var(Y),
X = Y.

% unificando uma variavel com uma nao-variavel
% a unificação apenas pode ser feita se a variável em questão
% não comparece na não-variavel
unify(X,Y):-
var(X),
nonvar(Y),
((atomic(Y),!); not_appear(X,Y)),
X = Y.

% mesmo que anterior
unify(X,Y):-
var(Y),
nonvar(X),
((atomic(X),!); not_appear(Y,X)),
X = Y.

% argumentos atomicos
unify(X,Y):-
nonvar(X),
nonvar(Y),
atomic(X),
atomic(Y), !,
X = Y.

% argumentos sao estruturas
unify(X,Y):-
nonvar(X),
nonvar(Y),
```

```
X =.. [F,X1|Xs],
Y =.. [F,Y1|Ys],
unify_args([X1|Xs],[Y1|Ys]).

unify_args([],[]).
unify_args([X|Xs],[Y|Ys]):-
unify(X,Y),
unify_args(Xs,Ys).

not_appear(X,Y):-
Y =.. L,
not_appear1(X,L).

not_appear1(X,[]).

not_appear1(X,[L|Ls]):-
atomic(L),!,
not_appear1(X,Ls).

not_appear1(X,[L|Ls]):-
var(L),!,
not ( X == L),
not_appear1(X,Ls).

not_appear1(X,[L|Ls]):-
not_appear(X,L),
not_appear1(X,Ls),!.
```

Exemplos de interrogações com o programa Prolog anterior carregado em memória:

```
?- unify(not p(X,a), not p(a,Y)).

X = a
Y = a
yes

?- unify(not(p(X,a)), not(p(a,Y))).

X = a
Y = a
yes
```

```
?- unify(p(X,q), p(N,M)).

X = _0084
Y = _0084
M = q ->;
no

?- unify(p(M,g(X)),p(a,X)).

no

?- unify([a],[M]).

M = a
yes

?- unify([a],[M|N]).

M = a
N = []
yes

?- unify(f(g(Y)),f(Z)).

Y = _0084
Z = g(_0084) ->;
no
```

Observação 4.10 O programa Prolog que implementa a operação de unificação, descrito anteriormente, usa alguns predicados predefinidos do Prolog (no caso, Arity Prolog):

- var/1: predicado que é bem-sucedido se seu argumento for uma variável não-instanciada;
- símbolo '= ': é o predicado de unificação do Prolog;
- nonvar/1: predicado que é bem-sucedido quando seu argumento estiver instanciado;
- atomic/1: bem-sucedido quando seu argumento for um tipo atômico de dado (átomos, inteiros, reais, cadeias);
- símbolo '== ': predicado que compara dois termos e é bem-sucedido se os termos são equivalentes.

4.3 O PROCESSO DE CONVERSÃO PARA A FORMA CLAUSAL

Como visto na Seção 1.9 do Capítulo 1, a FNC de wffs é de extrema importância para viabilizar o uso do método de resolução, uma vez que para tal método ser aplicado a uma fórmula ela deve estar representada na FNC. Para fórmulas da Lógica de Predicados a transformação exige alguns cuidados adicionais que os exigidos em Lógica Proposicional, fundamentando-se também no conceito de forma Prenex.

Definição 4.9 Uma fórmula α da Lógica de Predicados está na *Forma Normal Prenex* (FNP) se e somente se α estiver na forma:

$$(Q_1X_1)(Q_2X_2)\ldots(Q_nX_n)\ (M)$$

em que cada (Q_iX_i), $i = 1,\ldots,n$ é $(\forall X_i)$ ou $(\exists X_i)$, e M é uma fórmula que não contém quantificadores. A parte $(Q_1X_1)(Q_2X_2)\ldots(Q_nX_n)$ é chamada de *prefixo* de α e M é chamada de matriz de α.

Diz-se que α está na FNC se e somente se estiver na FNP e sua matriz for uma conjunção de disjunções de fórmulas atômicas, negadas ou não.

As regras, a seguir, devem ser seguidas para transformar uma fórmula da Lógica de Predicados em sua representação em forma normal conjuntiva.

1. *Eliminar variáveis livres*

   Se a fórmula α tiver uma variável livre, X, substituir α por $(\exists X\ \alpha)$. Isto deve ser repetido até que a fórmula não contenha mais variável livre.

2. *Eliminar quantificadores redundantes*

   Eliminar todo quantificador $\forall X$ ou $\exists X$ que não contenha nenhuma ocorrência livre de X em seu escopo – isto é, eliminar todo quantificador 'desnecessário'.

3. *Renomear variáveis quantificadas mais do que uma vez*

   Se uma mesma variável é governada por dois quantificadores, substituir a variável de um deles e todas as suas ocorrências livres no escopo do quantificador por uma nova variável que não ocorra na fórmula. Esse passo deve ser repetido até que todos os quantificadores governem variáveis diferentes. Assim, em vez de:

$$(\forall X\ p(X)) \rightarrow (\exists X\ q(X)),$$

escreve-se:

$$(\forall X\ p(X)) \rightarrow (\exists Y\ q(Y))$$

4. *Remover equivalências e implicações*

   $\alpha \leftrightarrow \beta$ é substituída por sua equivalente $(\neg\alpha \vee \beta) \wedge (\neg\beta \vee \alpha)$ e $\alpha \rightarrow \beta$ por $(\neg\alpha \vee \beta)$.

5. *Mover a negação para o interior da fórmula*

   Os sinais de negação são movidos de fora dos parênteses para a frente dos átomos, usando as generalizações de De Morgan e o fato de $\neg(\neg\alpha) \equiv \alpha$. Então, até que a ocorrência de $\neg$ preceda imediatamente uma fórmula atômica, substitui-se:

$$\neg(\forall X)(\alpha) \text{ por } (\exists X)(\neg\alpha)$$

$$\neg(\exists X)(\alpha) \text{ por } (\forall X)(\neg\alpha)$$

$$\neg(\alpha \wedge \beta) \text{ por } (\neg\alpha \vee \neg\beta)$$

$$\neg(\alpha \vee \beta) \text{ por } (\neg\alpha \wedge \neg\beta)$$

$$\neg\neg\alpha \text{ por } \alpha$$

   Dada a fórmula: $(\forall X\ (\neg p(X) \vee (\neg(\forall Y\ (\neg q(X,Y) \vee p(Y))))))$, após o Passo 5, obtém-se:

$$(\forall X\ (\neg p(X) \vee (\neg(\forall Y\ (\neg q(X,Y) \vee p(Y))))))$$

$$(\forall X\ (\neg p(X) \vee (\exists Y\ (q(X,Y) \wedge \neg p(Y))))).$$

6. *Eliminar os quantificadores existenciais – Skolemização*

   Considere a wff:

$$(\forall Y(\exists X\ p(X,Y)))$$

   que pode ser lida como para todo Y, existe um X, tal que p(X,Y). Pelo fato de o quantificador existencial estar dentro do escopo do quantificador universal, está aberta a possibilidade de o X que existe depender do valor Y. Esta dependência pode ser evidenciada por meio de uma função *g(Y)*, a qual associa cada valor de Y ao X que existe. Essa função é chamada de *função de Skolem*, e, se for utilizada no lugar do X que existe, o quantificador existencial pode ser eliminado e a fórmula anterior é reescrita da seguinte maneira:

$$(\forall Y)(p(g(Y),Y)).$$

   A regra geral para eliminar um quantificador existencial de uma wff é substituir cada ocorrência da variável quantificada existencialmente por uma função de Skolem. Tal função tem como argumentos as variáveis quantificadas universalmente, desde que o escopo do quantificador universal inclua o escopo do quantificador existencial sendo eliminado. Por exemplo, ao se skolemizar a variável Z da fórmula a seguir:

$$(\forall W\ q(W)) \wedge (\forall X(\forall Y(\exists Z\ (\ p(X,Y,Z) \vee (\forall U\ r(X,U,Y,Z))))))$$

obtém-se a fórmula:

$$(\forall W\ q(W)) \wedge (\forall X(\forall Y(p(X,Y,g(X,Y)) \vee (\forall U\ r(X,U,Y,g(X,Y)))))).$$

Se o quantificador existencial que está sendo eliminado não estiver dentro do escopo de nenhum quantificador universal, então é usada a função de Skolem sem argumentos, a qual é simplesmente uma constante. Assim, $\exists X\ p(X)$ torna-se $p(a)$, em que o símbolo a é usado como referência a um objeto do qual se sabe a existência. O símbolo a deve ser um novo símbolo constante, e não um símbolo já usado em referência a objetos conhecidos. Para eliminar todas as variáveis existencialmente quantificadas de uma wff usa-se, repetidamente, o procedimento descrito.

7. *Obter a forma normal Prenex e remover os quantificadores universais*

   Neste ponto não existem mais quantificadores existenciais, e cada quantificador universal tem sua própria variável. Pode-se, pois, mover todos os quantificadores universais para a frente da wff e deixar que o escopo de cada um deles inclua a wff inteira; esse procedimento deixará a fórmula da FNP.

   Desde que todas as variáveis nas wffs estejam ligadas, é certo que todas as variáveis que comparecem na fórmula, neste ponto, estão quantificadas universalmente. Pode-se, então, eliminar as ocorrências explícitas dos quantificadores universais e assumir, por convenção, que todas as variáveis da matriz são universalmente quantificadas.

8. *Colocar a matriz da FNP na forma conjuntiva*

   Qualquer matriz pode ser escrita como uma conjunção de um conjunto finito de disjunções de literais – forma conhecida como forma normal conjuntiva. Então, até que a matriz da fórmula seja uma conjunção de disjunções, substituir:

$$(\alpha_1 \wedge \alpha_2) \vee \alpha_3 \Rightarrow (\alpha_1 \vee \alpha_3) \wedge (\alpha_2 \vee \alpha_3)$$

$$\alpha_1 \vee (\alpha_2 \wedge \alpha_3) \Rightarrow (\alpha_1 \vee \alpha_2) \wedge (\alpha_1 \vee \alpha_3).$$

Exemplo 4.15 Considere a obtenção da FNC associada a cada uma das wffs a seguir, usando o processo anterior.

(a)

$$(\forall X(\exists Y(\forall Z(irma(Z,Y) \leftrightarrow (irma(Z,X) \wedge irma(X,Y))))))$$

$$\Downarrow$$

$$(\forall X(\exists Y(\forall Z((irma(Z,Y) \rightarrow ((irma(Z,X) \wedge irma(X,Y))) \wedge ((irma(Z,X) \wedge irm\tilde{a}(X,Y))) \rightarrow (irma(Z,Y))))))$$

$$\Downarrow$$

$$(\forall X(\exists Y(\forall Z((\neg irma(Z,Y) \vee (irma(Z,X) \wedge irma(X,Y))) \wedge ((\neg irma(Z,X) \vee \neg irm\tilde{a}(X,Y)) \vee (irma(Z,Y))))))$$

$$\Downarrow$$

$$(\forall X(\exists Y(\forall Z((\neg irma(Z,Y) \vee (irma(Z,X)) \wedge (\neg irma(Z,Y) \vee irma(X,Y)) \wedge ((\neg irma(Z,X) \vee \neg irma(X,Y)) \vee (irma(Z,Y))))))$$

$$\Downarrow$$

$$(\forall X(\forall Z((\neg irma(Z,f(X)) \vee (irma(Z,X)) \wedge (\neg irma(Z,f(X)) \vee irma(X,f(X))) \wedge ((\neg irma(Z,X) \vee \neg irma(X,f(X)) \vee (irma(Z,f(X))))))$$

C_1: $\neg irma(Z,f(X)) \vee (irma(Z,X)$

C_2: $\neg irma(Z,f(X)) \vee irma(X,f(X))$

C_3: $\neg irma(Z,X) \vee \neg irma(X,f(X)) \vee irma(Z,f(X))$

(b)

$$(\forall X(\forall Y((\exists Z(p(X,Z) \wedge p(Y,Z))) \rightarrow (\exists U\ q(X,Y,U)))))$$

$$\Downarrow$$

$$(\forall X(\forall Y(\neg(\exists Z\ (p(X,Z) \wedge p(Y,Z))) \rightarrow ((\exists U\ q(X,Y,U)))))$$

$$\Downarrow$$

$$(\forall X(\forall Y((\forall Z\ \neg((p(X,Z) \wedge p(Y,Z))) \vee ((\exists U\ q(X,Y,U)))))$$

$$\Downarrow$$

$$(\forall X(\forall Y((\forall Z\ (\neg p(X,Z) \vee \neg p(Y,Z))) \vee ((\exists U\ q(X,Y,U)))))$$

$$\Downarrow$$

$$(\forall X(\forall Y((\forall Z\ (\neg p(X,Z) \vee \neg p(Y,Z))) \vee (q(X,Y,f(X,Y))))))$$

$$\Downarrow$$

$$(\forall X(\forall Y(\forall Z\ (\neg p(X,Z) \vee \neg p(Y,Z) \vee q(X,Y,f(X,Y))))))$$

C: $\neg p(X,Z) \vee \neg p(Y,Z) \vee q(X,Y,f(X,Y))$

(c)

$$(\forall X(\forall Y(\, s(X,Y) \leftrightarrow (\forall U(\, p(U,X) \rightarrow p(U,Y))))))$$

$$\Downarrow$$

$$(\forall X(\forall Y((s(X,Y) \rightarrow (\forall U(\, p(U,X)\ \ p(U,Y)))) \wedge ((\forall U(\, p(U,X) \rightarrow p(U,Y))) \rightarrow s(X,Y)))))$$

$$\Downarrow$$

$$(\forall X(\forall Y((\neg s(X,Y) \vee (\forall U(\neg\, p(U,X) \vee p(U,Y)))) \wedge (\neg(\forall U(\, p(U,X) \rightarrow p(U,Y))) \vee s(X,Y)))))$$

$$\Downarrow$$

$$(\forall X(\forall Y((\neg s(X,Y) \vee (\forall U(\neg\, p(U,X) \vee p(U,Y)))) \wedge ((\exists U\, \neg(\neg p(U,X) \vee p(U,Y))) \vee s(X,Y))))$$

$$\Downarrow$$

$$(\forall X(\forall Y((\neg s(X,Y) \vee (\forall W(\neg\, p(W,X) \vee p(W,Y)))) \wedge (\exists U\, (\, p(U,X) \wedge \neg p(U,Y)) \vee s(X,Y))))$$

$$\Downarrow$$

$$(\forall X(\forall Y((\neg s(X,Y) \vee (\forall W(\neg\, p(W,X) \vee p(W,Y)))) \wedge (((p(f(X,Y),X) \wedge \neg p(f(X,Y),Y)) \vee s(X,Y))$$

$$\Downarrow$$

$$(\forall X(\forall Y((\neg s(X,Y) \vee (\forall W(\neg p(W,X) \vee p(W,Y)))) \wedge ((p(f(X,Y),X) \vee s(X,Y)) \wedge (\neg p(f(X,Y),Y)) \vee s(X,Y))))$$

$$\Downarrow$$

$$(\forall X(\forall Y(\forall W\, ((\neg s(X,Y) \vee \neg p(WX) \vee p(W,Y)) \wedge (p(f(X,Y),X) \vee s(X,Y)) \wedge (\neg p(f(X,Y),Y)) \vee s(X,Y)))))$$

C_1: $\neg s(X,Y) \vee \neg p(W,X) \vee p(W,Y)$

C_2: $p(f(X,Y),X) \vee s(X,Y)$

C_3: $\neg p(f(X,Y),Y)) \vee s(X,Y)$

(d)

$$(\forall X(p(X)) \rightarrow (\exists X\, q(X))$$

$$\Downarrow$$

$$(\forall X(p(X)) \vee (\exists Y\, q(Y))$$

$$\Downarrow$$

$$(\exists X\, \neg p(X)) \vee (\exists Y\, q(Y))$$

C: $\neg p(a) \vee q(b)$

5. RESOLUÇÃO EM LÓGICA DE PREDICADOS

5.1 CONSIDERAÇÕES INICIAIS

Como visto na Seção 1.9 do Capítulo 1, sistemas fundamentados em resolução são projetados para a produção de provas por contradição ou refutação. A regra de inferência da resolução pode ser aplicada a certas wffs denominadas cláusulas, que são disjunções de literais. O processo de unificação com a obtenção de uma substituição, tratado no Capítulo 4, é fundamental para o uso de resolução em Lógica de Predicados, em razão das variáveis e dos termos que comparecem em literais. Para o uso de resolução em cláusulas contendo variáveis, é preciso encontrar uma substituição (mgu) que possa ser aplicada a cláusulas pais de maneira que essas cláusulas passem a ter (quando for possível) literais complementares, precondição que permite o uso de resolução. Este capítulo discute o processo em detalhes, com base nos conceitos apresentados e discutidos no Capítulo 4.

Considere S um conjunto de wffs a partir do qual se deseja provar alguma meta G (wff). Em um processo de resolução por refutação, primeiro se nega a wff meta e, então, adiciona-se a negação ao conjunto S. Esse conjunto expandido é então convertido em um conjunto de cláusulas e a resolução é usada para derivar a contradição, representada pela cláusula nil.

Se G for uma wff que segue logicamente de um conjunto S de wffs, toda interpretação que satisfaz S satisfaz G também, ou seja, nenhuma interpretação que satisfaz S irá satisfazer $\neg$G. Como visto anteriormente, um conjunto de wffs que não pode ser satisfeito por qualquer interpretação é denominado insatisfatível. Assim, se G segue logicamente de S, o conjunto $S \cup \{\neg G\}$ deve ser insatisfatível. Pode ser mostrado que se a resolução for repetidamente aplicada a um conjunto de cláusulas insatisfatíveis, eventualmente a cláusula vazia, nil, será produzida. Assim, se G segue logicamente de S, o processo de resolução deverá, eventualmente, produzir a cláusula vazia, a partir da representação clausal de $S \cup \{\neg G\}$. De maneira análoga, pode ser mostrado que, se a cláusula vazia é produzida a partir da representação clausal de $S \cup \{\neg G\}$, então G segue logicamente de S.

5.2 UM EXEMPLO

Considere as três expressões da Lógica de Predicados, a seguir, que representam três premissas:

1. α: $(\forall X\ (p(X) \rightarrow q(X)))$
2. β: $(\forall X\ (r(X) \rightarrow \neg q(X)))$
3. γ: $(\exists X\ (r(X) \land s(X)))$

Suponha que se queira provar, usando resolução, que a conclusão:

4. δ: $(\exists X\ (s(X) \land \neg p(X)))$

segue logicamente das três premissas. Como visto anteriormente, o uso de resolução é condicionado à representação das fórmulas envolvidas em FNC. Assim, primeiramente a representação de cada uma das premissas na FNC deve ser obtida como a seguir (lembre-se de que na representação em FNC as variáveis estão implicitamente quantificadas universalmente).

$$\text{FNC}(\alpha)\text{: } \neg p(X) \vee q(X)$$

$$\text{FNC}(\beta)\text{:} \neg r(X) \vee \neg q(X)$$

$$\text{FNC}(\gamma)\text{: } r(a) \wedge s(a)$$

Na FNC(γ), a letra a representa uma constante de Skolem. A negação da conclusão e sua subseqüente conversão para a FNC produzem: $\neg(\exists X\ (s(X) \wedge \neg p(X))) \equiv (\forall X\ \neg(s(X) \wedge \neg p(X))) \equiv (\forall X\ (\neg s(X) \vee \neg(\neg p(X)))) \equiv (\forall X\ (\neg s(X) \vee p(X)))$. Portanto,

$$\text{FNC}(\delta)\text{:} \neg s(X) \vee p(X)$$

Identificando cada uma das cláusulas das FNCs obtidas, tem-se:

$$C_1\text{: } \neg p(X) \vee q(X)$$

$$C_2\text{: } \neg r(X) \vee \neg q(X)$$

$$C_3\text{: } r(a)$$

$$C_4\text{: } s(a)$$

$$C_5\text{: } \neg s(X) \vee p(X)$$

Padronizando as variáveis à parte, de maneira que cláusulas não compartilhem variáveis, tem-se:

$$C_1\text{: } \neg p(X) \vee q(X)$$

$$C_2\text{: } \neg r(Y) \vee \neg q(Y)$$

$$C_3\text{: } r(a)$$

$$C_4\text{: } s(a)$$

$$C_5\text{: } \neg s(Z) \vee p(Z)$$

A prova que a conclusão segue das premissas usando resolução por refutação envolve a geração de resolventes a partir do conjunto de cláusulas, a adição dos resolventes obtidos ao conjunto e a repetição desse processo até que a cláusula vazia seja produzida.

A Figura 5.1 mostra a árvore de refutação para o exemplo. Via de regra existem várias cláusulas e vários pares de literais complementares candidatos à aplicação do processo de resolução. Isso, entretanto, não previne a aplicação da resolução apropriada, mesmo após a aplicação de algumas irrelevantes.

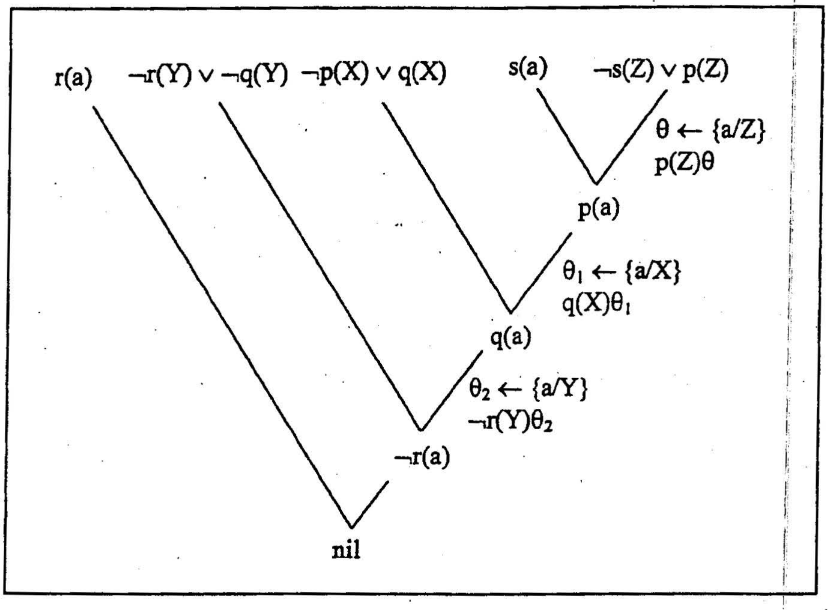


Figura 5.1 Árvore de refutação relativa à prova, usando resolução para o exemplo considerado.

5.3 USANDO RESOLUÇÃO

Definição 5.1 Sejam duas cláusulas candidatas a serem cláusulas pais, que não têm variáveis comuns, representadas como conjuntos de literais C:$\{L_i\}$ e C': $\{M_i\}$, com a disjunção implícita entre eles. Seja $\{l_i\} \subseteq \{L_i\}$ e $\{m_i\} \subseteq \{M_i\}$, tal que exista a substituição $\theta = mgu(\{l_i\} \cup \neg\{m_i\})$. Diz-se, então, que as duas cláusulas podem ser resolvidas e que a nova cláusula:

$$(\{L_i\} - \{l_i\})\theta \cup \subseteq (\{M_i\} - \{m_i\})\theta$$

é o *resolvente* das duas cláusulas C e C'.

Observação 5.1 Se duas cláusulas resolvem, então pode existir mais do que um resolvente, uma vez que podem existir diferentes escolhas para o subconjunto de literais $\{l_i\}$ e o subconjunto de literais $\{m_i\}$ da Definição 5.1. Em qualquer caso, o número máximo de resolventes existentes é finito. O Procedimento 5.1 descreve um pseudocódigo geral do processo de resolução, que recebe como entrada um conjunto de cláusulas, cujas variáveis já foram padronizadas à parte.

Procedimento 5.1 Procedimento de resolução.

```
procedure resolution(S)
begin
  C ← S
  repeat
    selecionar({Ci,Cj},C)
    while not (resolvível(Ci,Cj)) selecionar({Ci,Cj},C)
    rij ← resolvente(Ci,Cj)
    C ← C ∪ {rij}
  until nil ∈ C do
end
```

Exemplo 5.1 Considere as duas cláusulas e as várias possibilidades de escolhas dos subconjuntos de literais, como mostra a Tabela 5.1.

$$C_1: p(X,f(a)) \vee p(X,f(Y)) \vee q(Y)$$
$$C_2: \neg p(Z,f(a)) \vee \neg q(Z)$$

Tabela 5.1 Diferentes escolhas dos subconjuntos de literais, quando aplicando resolução.

$\{l_i\}$	$\{m_i\}$	resolvente
$\{p(X,f(a))\}$	$\{\neg p(Z,f(a))\}$	$p(Z,f(Y)) \vee q(Y) \vee \neg q(Z)$
$\{p(X,f(a)),p(X,f(Y))\}$	$\{\neg p(Z,f(a))\}$	$q(a) \vee \neg q(Z)$
$\{q(Y)\}$	$\{\neg q(Z)\}$	$p(X,f(a)) \vee p(X,f(Z)) \vee \neg p(Z,f(a))$
$\{p(X,f(Y))\}$	$\{\neg p(Z,f(a))\}$	$p(Z,f(a)) \vee q(a) \neg q(Z)$

Exemplo 5.2 Considere a sentença lógica α e a FNC(α) dadas a seguir:

$$\alpha: (\forall X(\forall Y\, p(X,Y) \rightarrow q(X,Y)) \wedge (\forall Y\, p(Y,a))$$
$$\text{FNC}(\alpha): (\neg p(X,Y) \vee q(X,Y)) \wedge p(Y,a)$$

$$C_1: (\neg p(X,Y) \vee q(X,Y))$$
$$C_2: p(Y,a)$$

Como pode ser visto na Figura 5.2, a fórmula logicamente implicada por α foi muito especializada, como conseqüência da variável Y, comum às duas cláusulas.

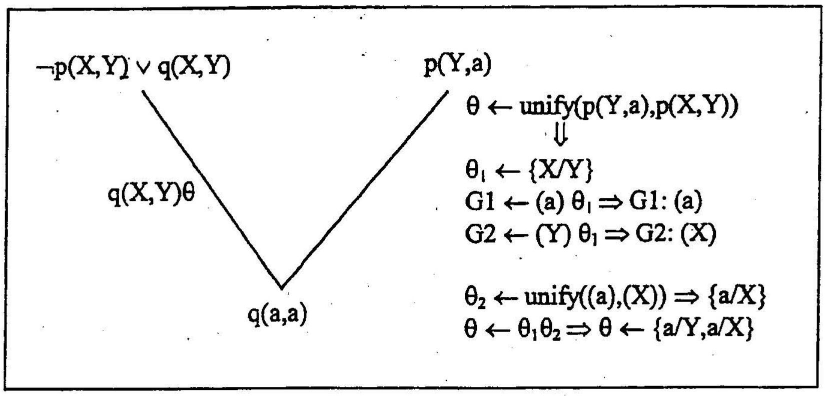


Figura 5.2 Efeito indesejável causado por compartilhamento de variáveis entre cláusulas, quando do uso de resolução.

Como mostra a Figura 5.2, usando a regra de inferência da resolução, chega-se apenas a q(a,a). Se a fórmula original for analisada cuidadosamente, entretanto, é fácil verificar que a fórmula α: $(\forall X(\forall Y\ p(X,Y) \rightarrow q(X,Y)) \wedge (\forall Y\ p(Y,a))$ logicamente implica a fórmula $(\forall Y\ q(Y,a))$; no entanto, no processo descrito, foi obtido apenas q(a,a) – note que houve um excesso de especialização desnecessária, quando da unificação. A razão disso se deve ao fato de ambas as cláusulas compartilharem a variável Y. Considere novamente todo o processo, agora renomeando a variável da segunda cláusula. Como pode ser visto na árvore de derivação mostrada na Figura 5.3, q(X,a) foi obtida e, como todas as variáveis são consideradas quantificadas universalmente, tem-se $(\forall X\ p(X,a))$.

α_1: $(\forall X(\forall Y\ p(X,Y) \rightarrow q(X,Y)) \wedge (\forall Z\ p(Z,a))$
FNC(α_1): $(\neg p(X,Y) \vee q(X,Y)) \wedge p(Z,a)$

C_1: $(\neg p(X,Y) \vee q(X,Y))$
C_2: $p(Z,a)$

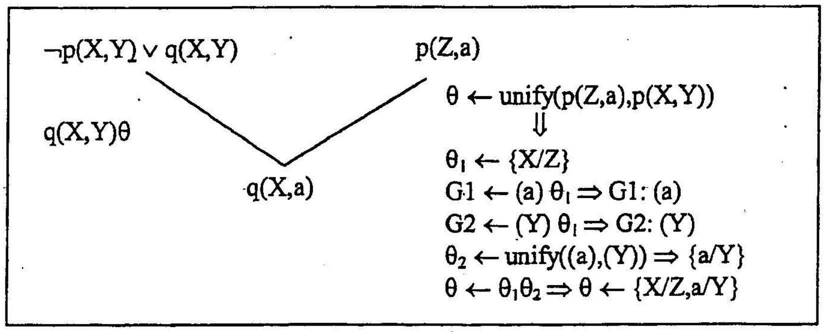


Figura 5.3 Processo de resolução em cláusulas que não compartilham variáveis.

Exemplo 5.3 Os possíveis resolventes das cláusulas C_1 e C_2 dadas a seguir estão mostrados na Tabela 5.2.

C_1: p(X,f(Y)) ∨ q(g(Y)) ∨ r(X,Z,b)
C_2: ¬p(X,f(b)) ∨ ¬q(Z) ∨ r(X,a,b)

Como ambas as cláusulas têm variáveis comuns, o primeiro passo é reescrevê-las de maneira que não compartilhem variáveis. Reescrevendo C_2 como $C_{2'}$:

C_1: p(X,f(Y)) ∨ q(g(Y)) ∨ r(X,Z,b)
$C_{2'}$: ¬p(U,f(b)) ∨ ¬q(V) ∨ r(U,a,b)

Tabela 5.2 Possíveis resolventes de C_1 e $C_{2'}$.

$literal_1$	$literal_2$	mgu	resolvente
p(X,f(Y))	{¬p(U,f(b))}	{U/X,b/Y}	q(g(b) ∨ r(U,Z,b) ∨ ¬q(V) ∨ r(U,a,b)
q(g(Y))	{¬q(V)}	{g(Y)/V}	p(X,f(Y)) ∨ r(X,Z,b) ∨ ¬p(U,f(b)) ∨ r(U,a,b)

Teorema 5.1 (Princípio de Resolução) Sejam α e β duas wffs da Lógica de Predicados que têm como resolvente a cláusula C. Então, {α, β} |= C.

Exemplo 5.4 A Figura 5.4 exemplifica o uso do princípio de resolução para evidenciar a insatisfatibilidade de A = {{¬p(X),q(X)}, {p(a)}, {r(b,a)}, {¬q(Y),¬r(b,Y)}}. Note que as cláusulas agora estão representadas como conjuntos de literais com a disjunção implicitamente assumida entre eles.

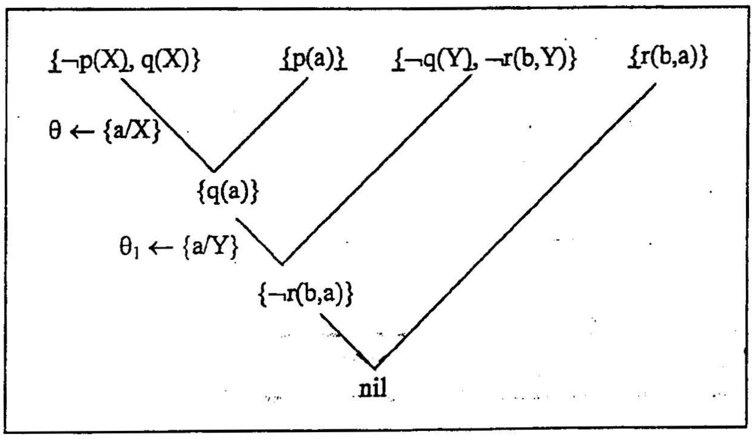


Figura 5.4 Resolução usada para evidenciar insatisfatibilidade.

Exemplo 5.5 A Figura 5.5 exemplifica o uso do princípio de resolução para evidenciar que o conjunto de cláusulas A = {{p(X), q(X,Y)}, {¬p(Z)}, {¬q(a,b),p(a),p(b)} é insatisfatível. Nesta figura, a segunda cláusula foi repetida três vezes para facilitar a visualização.

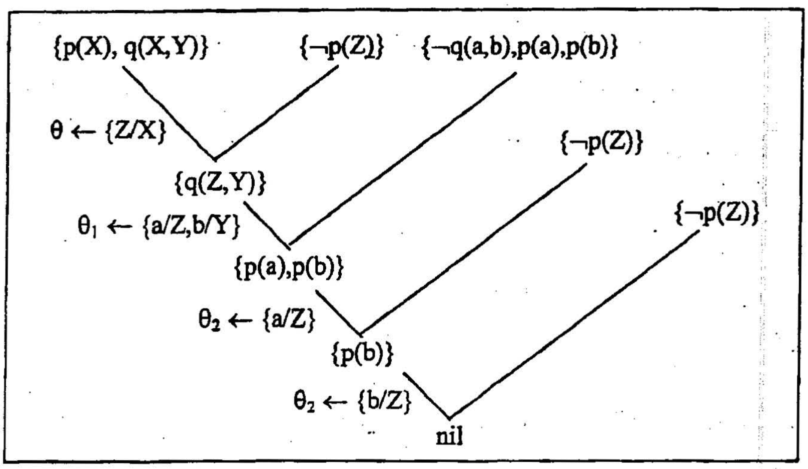


Figura 5.5 Árvore de refutação para um conjunto de cláusulas.

Exemplo 5.6 Considere o seguinte argumento representado em Lógica de Predicados pelas duas premissas e pela conclusão:

Premissa_1	α: $(\forall X(\forall Y((pai(X,Y) \wedge pai(Y,Z)) \rightarrow avo(X,Z))))$
Premissa_2	β: $(\forall X(\exists Y\, pai(X,Y)))$
Conclusão	δ: $(\exists X(\exists Y\, avo(X,Y)))$

A Tabela 5.3 mostra o processo de obtenção da forma normal conjuntiva de ambas as premissas α e β, bem como a conclusão negada, de maneira a construir um conjunto de cláusulas para o uso de resolução por refutação, como mostrado na Figura 5.6.

Tabela 5.3 Obtenção da FNC de duas premissas e da conclusão negada.

α: $(\forall X(\forall Y((pai(X,Y) \wedge pai(Y,Z)) \rightarrow avo(X,Z)))) \equiv$ $(\forall X(\forall Y(\neg(pai(X,Y) \wedge pai(Y,Z)) \vee avo(X,Z)))) \equiv$ $(\forall X(\forall Y(\neg pai(X,Y) \vee \neg pai(Y,Z) \vee avo(X,Z)))$ FNC(α):$(\neg pai(X,Y) \vee \neg pai(Y,Z) \vee avo(X,Z))$ (uma única cláusula com três literais)
β: $(\forall X(\exists Y\, pai(X,Y)))$ FNC(β): $pai(f(Y),Y)$ (uma única cláusula com um único literal)

Tabela 5.3 *Continuação...*

δ: Conclusão: $(\exists X(\exists Y\ avo(X,Y)))$ Negar a conclusão: $\neg(\exists X(\exists Y\ avo(X,Y)))$ FNC($\neg\delta$): $\neg(\exists X(\exists Y\ avo(X,Y))) \equiv$ $(\forall X \neg(\exists Y\ avo(X,Y))) \equiv$ $(\forall X(\forall Y \neg avo(X,Y)))$ FNC($\neg\delta$): $\neg avo(X,Y)$ (uma única cláusula com um único literal)
Reescrevendo o conjunto de cláusulas com as respectivas variáveis padronizadas à parte, tem-se: C_1: $(\neg pai(X,Y) \lor \neg pai(Y,Z) \lor avo(X,Z))$ C_2: $pai(f(K),K)$ C_3: $\neg avo(U,V)$

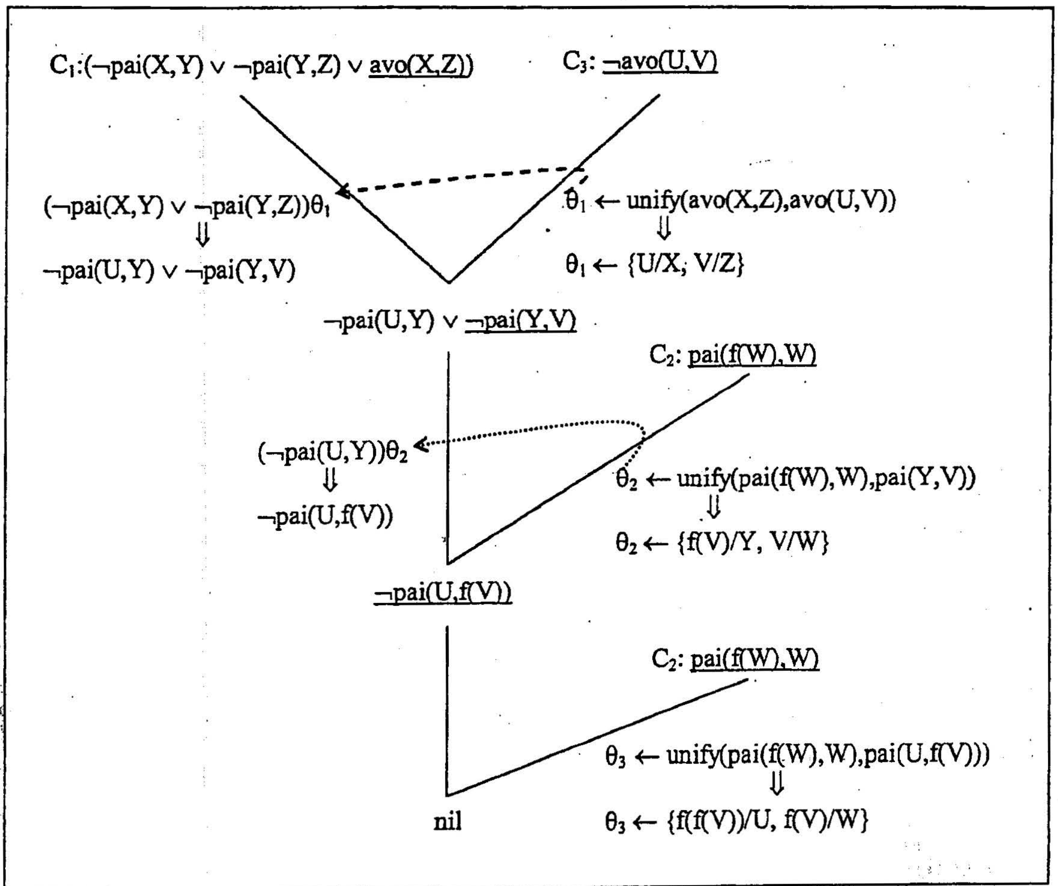


Figura 5.6 Árvore de refutação de um argumento usando resolução.

Para promover o entendimento, na Figura 5.6 estão mostrados os resultados dos processos de unificação entre literais, bem como as cláusulas estão representadas com os operadores $\vee$, e não como conjuntos de literais, como supostamente deveriam estar para a aplicação do processo de unificação, como formalmente definido anteriormente.

Nesta figura, a cláusula C_2 é mostrada duas vezes para uma melhor visualização do processo, dado que é uma das cláusulas pais em duas vezes que a regra de resolução é usada. As três substituições que possibilitaram o processo de resolução estão mostradas na Figura 5.7.

$\theta_1 \leftarrow$ unify((avo X Z), (avo U V)} $\Rightarrow \theta_1 \leftarrow$ {U/X, V/Z}

$\theta_2 \leftarrow$ unify((pai (f W) W), (pai Y V)) $\Rightarrow$ {(f V)/Y, V/W}
$\Downarrow$
unify(((f W)), (Y)) $\Rightarrow$ {(f W)/Y} {(f W)/Y} {V/W} = {(f V)/Y, V/W}
(W) {(f W)/Y} $\Rightarrow$ (W)
(V) {(f W)/Y}$\Rightarrow$ (V)
$\Downarrow$
unify((W) , (V)) $\Rightarrow$ {V/W}

$\theta_3 \leftarrow$ unify((pai (f W) W), (pai U (f V)) $\Rightarrow$ {(f V)/Y, V/W}
$\Downarrow$
unify(((f W)), (Y)) $\Rightarrow$ {(f W)/Y} {(f W)/Y} {V/W} = {(f V)/Y, V/W}
(W) {(f W)/Y} $\Rightarrow$ (W)
(V) {(f W)/Y}$\Rightarrow$ (V)
$\Downarrow$
unify((W) , (V)) $\Rightarrow$ {V/W}

Figura 5.7 Principais passos das três substituições usadas no processo de resolução mostrado na Figura 5.6.

Exemplo 5.7 Considere a seguinte descrição em língua natural:

1. Todo número primo outro que 2 é ímpar.
2. O quadrado de um número ímpar é ímpar.
3. O número 7 é primo.
4. O número 7 é diferente de 2.
5. O quadrado de 7 é ímpar.

Usando resolução, verificar se a assertiva 5 segue das premissas 1, 2, 3 e 4. Para a representação das sentenças em Lógica de Predicados os seguintes predicados serão usados: primo/1, ímpar/1, igual/2 e o símbolo funcional quadrado/1. Cada uma das premissas e a conclusão são traduzidas para:

1. $(\forall X\ ((\text{primo}(X) \wedge \neg\text{igual}(X,2)) \rightarrow \text{impar}(X)))$
2. $(\forall X\ (\text{impar}(X) \rightarrow \text{impar}(\text{quadrado}(X))))$

3. primo(7)
4. $\neg$igual(7,2)
5. impar(quadrado(7)).

Obtendo a FNC de cada premissa e da conclusão negada tem-se:

1. $(\forall X\ ((primo(X) \wedge \neg igual(X,2)) \rightarrow impar(X))) \equiv (\forall X\ (\neg(primo(X) \wedge \neg igual(X,2)) \vee impar(X))) \equiv (\forall X\ (\neg primo(X) \vee igual(X,2) \vee impar(X)))$.
2. $(\forall X\ (\cancel{impar}(X) \rightarrow impar(quadrado(X)))) \equiv (\forall X\ (\neg impar(X) \vee impar(quadrado(X))))$.
3. primo(7).
4. $\neg$igual(7,2).
5. $\neg$impar(quadrado(7)).

A Figura 5.8 mostra a árvore de refutação referente ao processo de resolução usado, na qual a cláusula referente à premissa 2 teve sua variável renomeada, uma vez que ela já comparecia na cláusula referente à premissa 1.

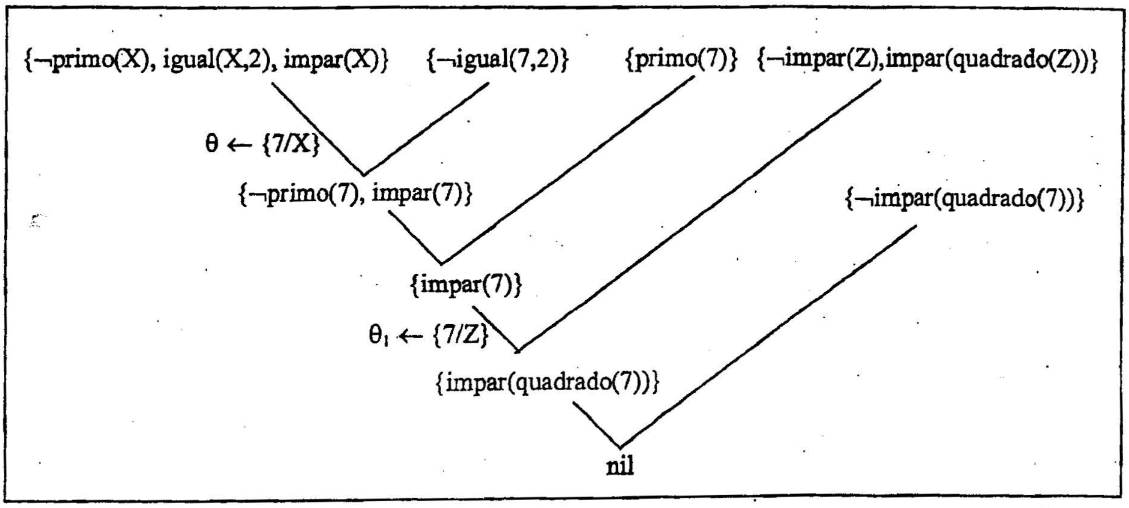


Figura 5.8 Árvore de refutação para um conjunto de cláusulas.

5.4 ESTRATÉGIAS DE CONTROLE PARA MÉTODOS DE RESOLUÇÃO

A decisão de quais cláusulas considerar para o uso de resolução, no Procedimento 5.1, é feita via estratégia de controle. Existem diferentes estratégias para controlar esse processo. A seguir, são apresentadas brevemente quatro estratégias, cujas descrições estão fundamentadas na discussão apresentada em Nilsson (1972).

Como mostrado anteriormente, uma resolução por refutação pode ser representada como uma árvore de refutação (dentro do grafo de derivação), tendo um nó raiz rotulado *nil.* A estratégia de controle busca a refutação, por meio do crescimento do grafo de derivação, até que a árvore seja produzida com o nó raiz rotulado pela cláusula nil.

Uma estratégia de controle para um sistema de refutação é completa se seu uso resultar em um procedimento que irá encontrar uma contradição (eventualmente) sempre onde existir uma. Para muitas aplicações é mais interessante ter uma estratégia eficiente do que completa.

5.4.1 Estratégia em largura

Na estratégia em largura, todos os resolventes do primeiro nível são calculados primeiro, então os do segundo nível e assim por diante. Um resolvente do primeiro nível é obtido entre duas cláusulas do conjunto base; um resolvente no i-ésimo nível é aquele cujo pai em nível mais profundo é um resolvente do nível (i–1). A estratégia em largura é completa mas altamente ineficiente. A Figura 5.9 mostra o uso da estratégia no exemplo discutido na Seção 5.2.

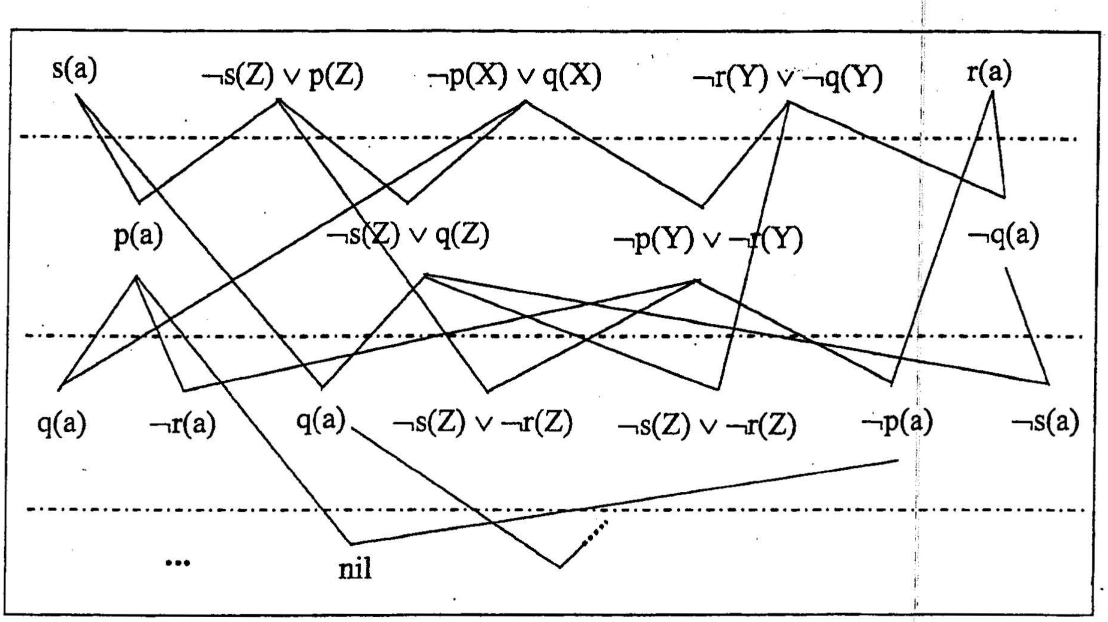


Figura 5.9 Estratégia para uso de resolução em largura.

5.4.2 Estratégia do *input* linear

A estratégia do *input* linear garante que cada resolvente tenha pelo menos um dos pais pertencente ao conjunto básico (conjunto inicial de cláusulas). A Figura 5.10 mostra o uso da estratégia no exemplo discutido na Seção 5.2.

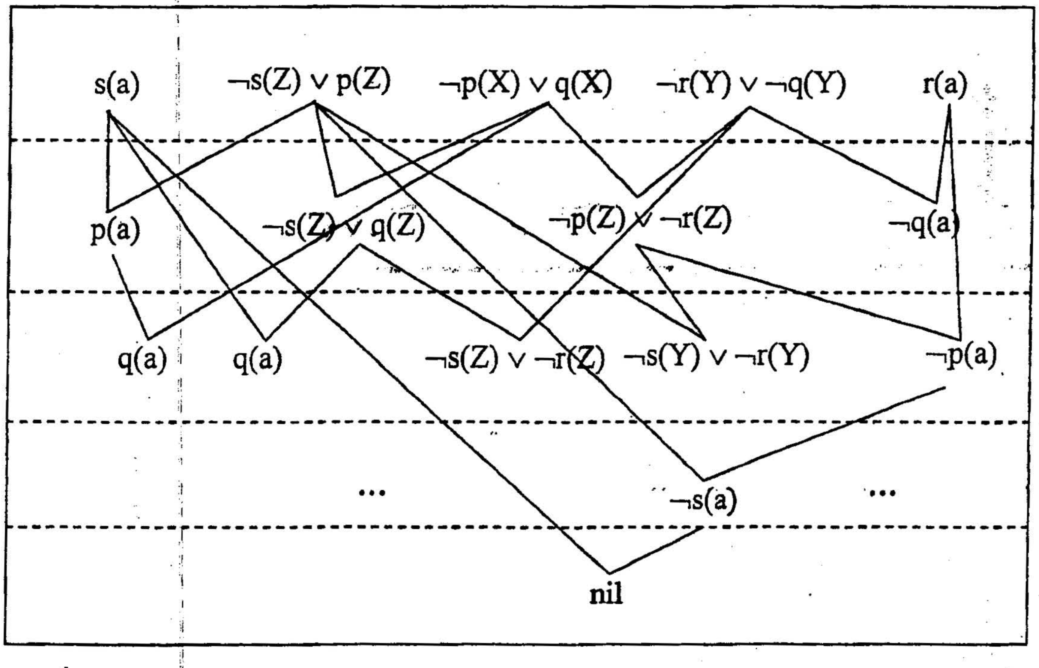


Figura 5.10 Árvore de refutação gerada pela estratégia do *input* linear.

A estratégia do *input* linear não é completa. Existem casos em que a refutação existe e, no entanto, a estratégia não a deriva. Apesar da falta de completude, a estratégia do *input* linear é freqüentemente usada por causa de sua simplicidade e eficiência.

5.4.3 Estratégia do conjunto suporte

O controle do processo de resolução via estratégia do conjunto suporte garante que pelo menos um pai de cada resolvente seja selecionado entre as cláusulas que resultam da negação da conclusão ou de seus descendentes (*o conjunto suporte*). Pode ser provado que sempre que existir uma refutação, a estratégia do conjunto suporte a identifica. A Figura 5.11 mostra a árvore de refutação para o exemplo em questão. Comparando a Figura 5.9 com a Figura 5.11, fica evidente que a estratégia do conjunto suporte produz menos cláusulas a cada nível que a estratégia em largura.

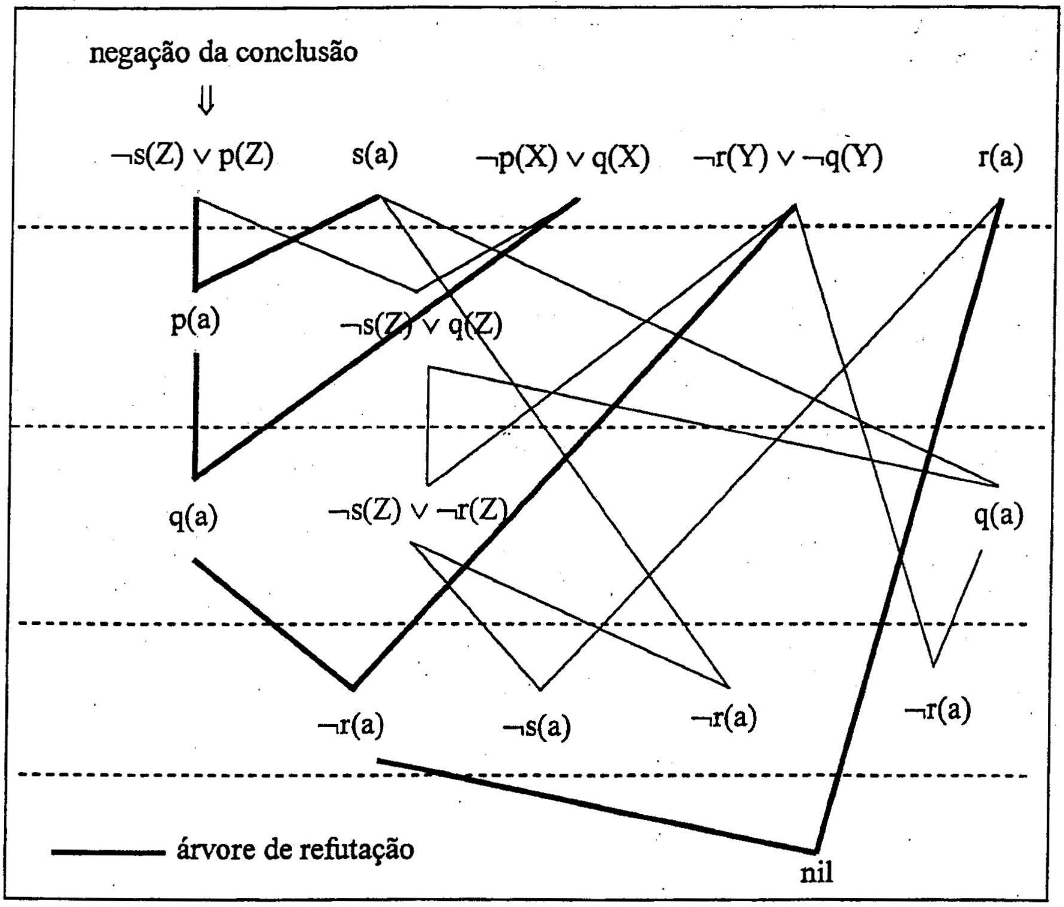


Figura 5.11 Árvore de refutação gerada pela estratégia do conjunto suporte.

5.4.4 Estratégia da preferência por cláusulas unitárias

Esta é uma modificação da estratégia do conjunto suporte, em que, em vez de preencher cada nível em largura, a estratégia busca selecionar como um dos pais do resolvente uma cláusula que tenha um único literal (cláusula unitária). Toda vez que cláusulas unitárias são usadas em resolução, os resolventes têm menos literais que seus pais. Esse processo ajuda a focalizar a busca na direção da produção de uma cláusula vazia e, assim, aumenta a eficiência. A árvore de refutação da Figura 5.12 é uma que pode ter sido produzida por uma estratégia de preferência por cláusulas unitárias, no exemplo discutido na Seção 5.2.

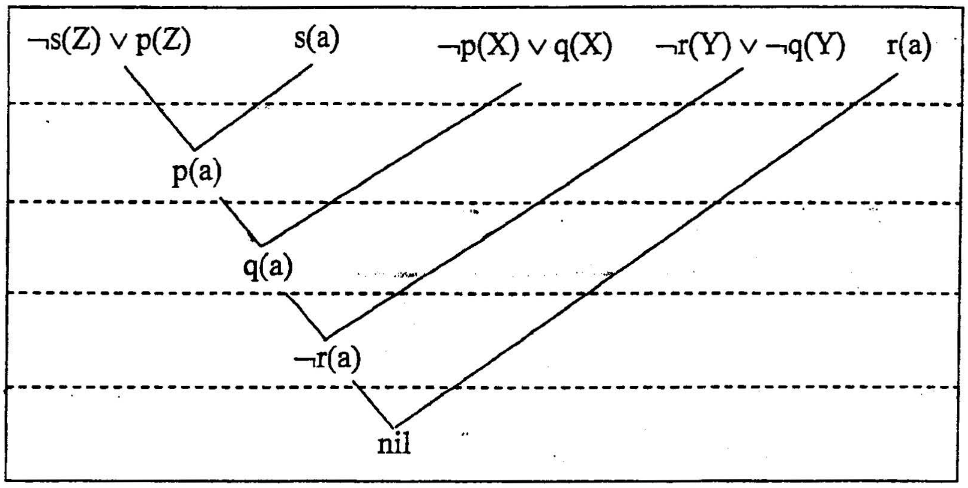


Figura 5.12 Árvore de refutação usando a estratégia com base em cláusulas unitárias.

A estratégia de preferência por cláusulas unitárias é um exemplo de estratégia de ordenação. Outras estratégias fundamentadas no número de literais em uma cláusula e na complexidade de seus termos podem também ser consideradas. A ordem, na qual resoluções são realizadas, é crucial à eficiência de sistemas de resolução.

6. UNIVERSO, INTERPRETAÇÃO E MODELO DE HERBRAND

6.1 CONSIDERAÇÕES INICIAIS

Como discutido anteriormente, a implicação lógica pode ser reduzida ao teste de insatisfatibilidade de uma fórmula na forma normal conjuntiva. Em Lógica Proposicional é possível enumerar todas as atribuições de valores-verdade para testar insatisfatibilidade. Em Lógica de Predicados, entretanto, seria necessário enumerar todas as estruturas para usar essa abordagem, o que a torna não-factível, uma vez que existem infinitas de tais estruturas. Felizmente, entretanto, existe uma maneira de decidir sobre insatisfatibilidade por meio do exame de apenas uma estrutura. Para cada fórmula existe um domínio especial – chamado de universo de Herbrand – e uma função especial – com base na interpretação de Herbrand, tal que apenas estruturas usando esse domínio e função devem ser consideradas. Esses conceitos são discutidos neste capítulo.

Como apresentado na Seção 5.1 do Capítulo 5, para uma fórmula G ser conseqüência lógica de um conjunto de fórmulas S, toda interpretação que for modelo de S deve também ser modelo de G. Assim, nenhum dos modelos de S pode ser um modelo de $\neg G$ e, portanto, nenhum modelo pode satisfazer $S \wedge \{\neg G\}$. Portanto, $S \wedge \{\neg G\}$ deve ser insatisfatível, o que permite dizer que, se M for conseqüência lógica de S, $S \wedge \{\neg G\}$ é insatisfatível.

Sabe-se que um conjunto de cláusulas é insatisfatível (ou inconsistente) se e somente se for falso sob todas as interpretações, em todos os domínios. Dado que não é possível considerar todas as interpretações em todos os domínios, um domínio particular H será definido, tal que S será insatisfatível se e somente se S for falso sob todas as interpretações em H. Este domínio particular é chamado de *universo de Herbrand de S*, e as interpretações são chamadas de *Interpretações de Herbrand.*

6.2 O UNIVERSO E A BASE DE HERBRAND

Definição 6.1 Seja S um conjunto de cláusulas e H_0 o conjunto de constantes que aparecem em S.

- Se nenhuma constante aparece em S, então seja $H_0 = \{a\}$.
- Para i = 0,1,...., seja

$$H_{i+1} = H_i \cup T$$

tal que T é o conjunto de todos os termos da forma $f^n(t_1,t_2,\ldots,t_n)$ para todos os símbolos funcionais f^n ocorrendo em S, tal que t_j para $j = 1,\ldots,n$ são elementos de H_i.

Cada H_i é chamado de conjunto constante de nível i de S e H_∞ (ou simplesmente H) é chamado de *universo de Herbrand de S.* Assim, o universo de Herbrand coincide com o conjunto de termos *ground* que são gerados a partir dos símbolos de constantes e dos símbolos funcionais que aparecem em S. Três situações podem ocorrer:

1. se S contém ambos, constantes e funções (símbolos funcionais), então o universo de Herbrand será sempre contavelmente infinito;
2. se S contém apenas constantes, o universo de Herbrand será o conjunto finito daquelas constantes;
3. se S não contém constantes, uma constante, por exemplo, a, é escolhida arbitrariamente. Dependendo da presença ou ausência de símbolos funcionais em S, o universo de Herbrand será infinito ou não, respectivamente.

Exemplo 6.1 Construir o universo de Herbrand para o conjunto de cláusulas $S = \{\neg p(X,Y) \vee q(X,f(X)), \neg p(X,Y) \vee q(X,g(Y))\}$.

Desde que não exista constante em S, seja então $H_0 = \{a\}$. Assim,

$H_1 = H_0 \cup \{f(a),g(a)\}$.
$H_2 = H_1 \cup \{f(f(a)),f(g(a)),g(f(a)),g(g(a))\}$.
$H_3 = H_2 \cup \{f(f(f(a))), f(f(g(a))), f(g(f(a))), f(g(g(a))), g(f(f(a))), g(f(g(a))), g(g(f(a))), g(g(g(a)))\}$.
......
$H_\infty = \{a, f(a), g(a), f(f(a)), f(g(a)), g(f(a)), g(g(a)), f(f(f(a))), f(f(g(a))), f(g(f(a))), f(g(g(a))), g(f(f(a))), g(f(g(a))), g(g(f(a))), g(g(g(a))), \ldots\}$.

Exemplo 6.2 Construir o universo de Herbrand para o conjunto de cláusulas $S = \{p(X) \vee q(X), r(Z), \neg s(Y) \vee t(W)\}$.

Desde que não exista constante em S, seja então $H_0 = \{a\}$.
Desde que não exista símbolo funcional em S, $H_0 = H_1 = H_2 = \ldots = \{a\}$.

Exemplo 6.3 Construir o universo de Herbrand para o conjunto de cláusulas $S= \{\neg p(f(X),g(X),h(Y)) \vee q(X),r(Y)\}$.

Desde que não exista constante em S, seja então $H_0 = \{a\}$. Assim,

$H_1 = H_0 \cup \{f(a),g(a),h(a)\}$
$H_2 = H_1 \cup \{f(f(a)), f(g(a)), f(h(a)), g(f(a)), g(g(a)), g(h(a)), h(f(a)), h(g(a)), h(h(a))\}$
$H_3 = H_2 \cup \{ f(f(f(a))), f(f(g(a))), f(f(h(a))), f(g(f(a))), f(g(g(a))), f(g(h(a))), f(h(f(a))), f(h(g(a))), f(h(h(a))), g(f(f(a))), g(f(g(a))), g(f(h(a))), g(g(f(a))), g(g(g(a))), g(g(h(a))),$

g(h(f(a))), g(h(g(a))), g(h(h(a))), h(f(f(a))), h(f(g(a))), h(f(h(a))), h(g(f(a))), h(g(g(a))), h(g(h(a))), h(h(f(a))), h(h(g(a))), h(h(h(a)))}

.....

H_∞ = {a, f(a), g(a), h(a), f(f(a)), f(g(a)), f(h(a)), g(f(a)), g(g(a)), g(h(a)), h(f(a)), h(g(a)), h(h(a)), f(f(f(a))), f(f(g(a))), f(f(h(a))), f(g(f(a))), f(g(g(a))), f(g(h(a))), f(h(f(a))), f(h(g(a))), f(h(h(a))), g(f(f(a))), g(f(g(a))), g(f(h(a))), g(g(f(a))), g(g(g(a))), g(g(h(a))), g(h(f(a))), g(h(g(a))), g(h(h(a))), h(f(f(a))), h(f(g(a))), h(f(h(a))), h(g(f(a))), h(g(g(a))), h(g(h(a))), h(h(f(a))), h(h(g(a))), h(h(h(a))), ...}

Exemplo 6.4 Construir o universo de Herbrand para o conjunto de cláusulas S = {p(a,b,f(X),g(Y),c)}.

Uma vez que existam três constantes – a,b e c – em S, então:

H_0 = {a, b, c}.
$H_1 = H_0 \cup$ {f(a), g(a), f(b), g(b), f(c), g(c)}.
$H_2 = H_1 \cup$ { f(f(a)), f(g(a)), f(f(b)), f(g(b)), f(f(c)), f(g(c)), g(f(a)), g(g(a)), g(f(b)), g(g(b)), g(f(c)), g(g(c))}.

....

H_∞ = {a, b, c, f(a), g(a), f(b), g(b), f(c), g(c), f(f(a)), ...}.

Definição 6.2 Seja S um conjunto de cláusulas e seja H_0 o universo de Herbrand de S. O conjunto dos átomos *ground* da forma $p^n(t_1,t_2,...,t_n)$ para todos os predicados p^n ocorrendo em S, em que t_1, t_2, ..., t_n são elementos do universo de Herbrand de S, é chamado de *base de Herbrand* ou *conjunto atômico* de S e é notado por B_S.

Exemplo 6.5 Seja S= {p(X) ∨ q(X), r(f(X))}.

Então, B_S = {p(a), q(a), r(a), p(f(a)), q(f(a)), r(f(a)),...}.

Exemplo 6.6 Seja S= {p(X), ¬p(X) ∨ q(f(X)), ¬q(f(a))}.

Então, B_S = {p(a), q(a), p(f(a)), q(f(a)), ...}.

Exemplo 6.7 Seja S= {¬p(X,a,Y) ∨ ¬p(b,X,Z) ∨ q(Y,Z)}.

Desde que S não tenha símbolo funcional, o universo de Herbrand de S é H_∞ = {a,b}. Então,

B_S = {p(a,a,a), p(a,a,b), p(a,b,a), p(a,b,b), p(b,a,a), p(b,a,b), p(b,b,a), p(b,b,b), q(a,a), q(a,), q(b,a), q(b,b)}.

Definição 6.3 Uma instância *ground* de uma cláusula C de um conjunto S de cláusulas é uma cláusula obtida pela substituição das variáveis em C por elementos do universo de Herbrand de S.

Exemplo 6.8 Seja

$$S= \{p(X), q(f(Y)) \vee r(Y)\}$$

$$H_\infty = \{a, f(a), f(f(a)), \ldots\}.$$

A Tabela 6.1 mostra algumas das instâncias *ground* das duas cláusulas C_1 e C_2 de S.

Tabela 6.1 Instâncias *ground* de cláusulas.

C_1: p(X)	C_2: q(f(Y)) ∨ r(Y)
p(a)	q(f(a)) ∨ r(a)
p(f(a))	q(f(f(a))) ∨ r(f(a))
p(f(f(a)))	q(f(f(f(a)))) ∨ r(f(f(a)))
....	....

Definição 6.4 Seja S um conjunto de cláusulas. Uma *instanciação ground* de S é o conjunto de todas as instâncias *ground* de todas suas cláusulas e é denotada por G_S.

Exemplo 6.9 Seja P o programa Prolog:

```
f(a,X) :- f(X,b).
f(c,b).
```

Note que P pode ser reescrito como o conjunto de cláusulas $S = \{\neg f(X,b) \vee f(a,X), f(c,b)\})$. Considerando que H = {a,b,c}, a instanciação *ground* G_P de P é o programa *ground*:

```
f(a,a):- f(a,b).
f(a,b):- f(b,b).
f(a,c):- f(c,b).
f(c,b).
```

Note que o programa citado pode ser reescrito como o conjunto $G_S = \{\neg f(a,b) \vee f(a,a), \neg f(b,b) \vee f(a,b), \neg f(c,b) \vee f(a,c), f(c,b)\}$. G_P é essencialmente proposicional.

6.3 INTERPRETAÇÃO E MODELO DE HERBRAND

A seguir, será definida uma interpretação especial sobre o universo de Herbrand de S chamada de *interpretação de Herbrand* de S.

Definição 6.5 Seja

S – um conjunto de cláusulas.

H – o universo de Herbrand de S.

I – uma interpretação de S sobre H.

I é uma *interpretação de Herbrand* de S se satisfaz às seguintes condições:

1. I associa cada uma das constantes de S a ela própria;
2. seja f um símbolo funcional n-ário e sejam $h_1, h_2, .., h_n$ elementos de H. Em I, a f é atribuída uma função que associa $(h_1, h_2, .., h_n)$ – um elemento de H^n – a $f(h_1,h_2,..,h_n)$ – um elemento de H.

Não existe restrição com relação à atribuição a cada símbolo predicado em S, de maneira que diferentes interpretações de Herbrand podem existir, dependendo de tais diferentes atribuições. Desde que para as interpretações de Herbrand a atribuição a constantes e símbolos funcionais seja fixa, é possível identificar uma interpretação de Herbrand com um subconjunto da base de Herbrand. Assim, daqui para frente:

Uma interpretação de Herbrand I de um conjunto de cláusulas S é qualquer subconjunto de B_S, a base de Herbrand,

entendendo que, sob a interpretação I, a qualquer átomo q em B_S é atribuído o valor-verdade v se $q \in I$; caso contrário, é atribuído f. Com o objetivo de avaliar um dado conjunto de cláusulas com relação a uma dada interpretação de Herbrand (I), as seguintes regras devem ser aplicadas:

1. a fórmula atômica *ground* A é v em I se e somente se $A \in I$;
2. uma literal negada *ground* $\neg A$ é v em I se e somente se $A \notin I$;
3. uma cláusula *ground* $L_1 \vee L_2 \vee \ldots \vee L_n$ é v em I se e somente se pelo menos um literal L_i for v em I;
4. em geral, uma cláusula C é v em I se e somente se toda instância *ground* $C\sigma$ de C for v em I ($C\sigma$ é obtida substituindo toda ocorrência de uma variável em C por um termo de H. Ocorrências diferentes da mesma variável são substituídas pelo mesmo termo);
5. um conjunto de cláusulas S é v em I se e somente se cada cláusula de S for v em I.

Um literal, cláusula ou conjunto de cláusulas é f em I se e somente se não for v. Uma *interpretação de Herbrand* é, pois, uma livre atribuição de valores-verdade (v ou f) a todos os átomos da B_S.

Tal interpretação deve, portanto, atribuir também v ou f a todos os átomos em G_P; assim, os valores-verdade desses átomos determinam os valores-verdade de todas as cláusulas em G_P.

Definição 6.6 Seja

S – um conjunto de cláusulas.

H – o universo de Herbrand de S.

I – uma interpretação de Herbrand de S sobre H.

Se S for v em I, então I é um *modelo de Herbrand* de S.

Exemplo 6.10 Seja S = {p(X) ∨ ¬q(X)} um conjunto de cláusulas.

Desde que S tenha B_S = {p(a),q(a)}, as possíveis $2^{|B_S|} = 4$ interpretações de Herbrand de S são:

- I_0: ∅;
- I_1: {p(a)};
- I_2: {q(a};
- I_3:{p(a),q(a)}.

De acordo com as regras de avaliação mostradas anteriormente, as interpretações I_0, I_1 e I_3 são *modelos de Herbrand* de S e I_2 é simplesmente uma *interpretação de Herbrand* de S.

Exemplo 6.11 Seja S = {p(X) ∨ ¬q(X), m(Y) ∨ ¬g(a), q(Z)} um conjunto de cláusulas. Desde que S tenha B_S = {p(a),q(a),m(a),g(a)}, as possíveis $2^{|B_S|} = 16$ interpretações de Herbrand de S estão mostradas na Tabela 6.2.

Tabela 6.2 As 16 possíveis interpretações de Herbrand.

∅	{g(a)}	{q(a),m(a)}	{p(a),q(a),g(a)}
{p(a)}	**{p(a),q(a)}**	{q(a),g(a)}	{p(a),m(a),g(a)}
{q(a)}	{p(a),m(a)}	{m(a),g(a)}	{q(a),m(a),g(a)}
{m(a)}	{p(a),g(a)}	**{p(a),q(a),m(a)}**	**{p(a),q(a),m(a),g(a)}**

Na Tabela 6.2, as interpretações em negrito são modelos.

Exemplo 6.12 [Hooger, 1991] Considere o conjunto de cláusulas P:

likes(chris,X) if likes(X,logic)

likes(bob, logic)

O universo de Herbrand de P é o conjunto {chris, bob, logic} e a instanciação *ground* de P, ie, G_P é:

likes(chris,chris) if likes(chris,logic)

likes(chris,bob) if likes(bob,logic)

likes(chris,logic) if likes(logic,logic)

likes(bob, logic).

que pode ser escrita, para maior facilidade de leitura, como:

CC if CL

CB if BL

CL if LL

BL

A *base de Herbrand* é B_P = {CC,CL,CB,LC,LL,LB,BC,BL,BB}. O conjunto {BL,CB} $\subset B_P$ é, assim, uma interpretação de Herbrand. Para essa interpretação, G_P é reproduzido, a seguir, com os valores-verdade associados escritos em lugar de seus átomos:

f if f

v if v

f if f

v

Com essa atribuição, todas as cláusulas de G_P são v. Esta interpretação é, portanto, um modelo de Herbrand para G_P e, portanto, um modelo de Herbrand de P.

Proposição 6.1 Seja S um conjunto de cláusulas e suponha que S tenha um modelo. Então, S tem um modelo de Herbrand.

Prova: Seja I uma interpretação de S. A interpretação de Herbrand I' de S pode ser definida como:

$$I' = \{p(t_1,\ldots,t_n)\ B_S \mid p(t_1,\ldots,t_n) \text{ é v com relação a I}\}.$$

É direto mostrar que, se I for um modelo, I' também será um modelo. Os dois exemplos seguintes mostram a construção de modelos de Herbrand correspondentes a modelos.

Exemplo 6.13 Seja S = $\{\neg p(X) \vee q(f(X),a)\}$ a forma clausal da fórmula α: $(\forall X\ (p(X) \rightarrow q(f(X),a)))$ e seja I o seguinte modelo:

a) domínio: {1,2}

b) atribuição a constante: a é atribuída a 1

c) atribuições a símbolos funcionais:

f(1)	f(2)
2	1

d) atribuição a símbolos predicados:

p(1)	p(2)	q(1,1)	q(1,2)	q(2,1)	q(2,2)
f	v	v	v	f	v

Note que esta interpretação é um modelo, desde que $\{\neg p(1) \vee q(f(1),1)\}$ e $\{\neg p(2) \vee q(f(2),1)\}$ sejam, ambos, verdade. Para a definição do correspondente modelo de Herbrand, é necessário:

- encontrar a base de Herbrand B_S de S e
- avaliar cada elemento de B_S, usando a dada interpretação I,

a fim de selecionar como elementos do modelo de Herbrand aqueles elementos de B_S que são avaliados como v pelo modelo I. Para o conjunto S considerado,

$$B_S = \{p(a), q(a,a), p(f(a)), q(a,f(a)), q(f(a),a), q(f(a),f(a)),\ldots\}$$

e a avaliação de cada elemento de BS usando o dado modelo I é:

p(a)	=	p(1)	=	f
q(a,a)	=	q(1,1)	=	v
p(f(a))	=	p(2)	=	v
q(a,f(a))	=	q(1,2)	=	v
q(a,f(a))	=	q(2,1)	=	f
....	..	....		....

Portanto, o modelo de Herbrand I'correspondente a I é:

$$I' = \{q(a,a), f(p(a)), q(a,f(a)),\ldots\}$$

No caso de não existirem constantes em S, a constante arbitrariamente escolhida a, que é usada para iniciar o universo de Herbrand de S, pode ser associada a qualquer elemento do domínio D. Neste caso, quando existir mais que um elemento em D, existirá mais que um modelo de Herbrand correspondente a I, como mostra o Exemplo 6.14.

Exemplo 6.14 Seja $S = \{\neg p(X) \vee q(f(X),Y)\}$ com a interpretação I, mostrada no exemplo anterior, exceto que não existe atribuição à constante. A base de Herbrand de S é a mesma do exemplo anterior, e a avaliação de cada elemento de B_S usando o dado modelo I é:

Para a = 1

p(a)	=	p(1)	=	f
q(a,a)	=	q(1,1)	=	v
p(f(a))	=	p(2)	=	v
....	..	....		....

Portanto, I' = {q(a,a),p(f(a)),...}.

Para a = 2

p(a)	=	p(2)	=	v
q(a,a)	=	q(2,2)	=	v
p(f(a))	=	p(1)	=	f
q(a,f(a))	=	q(2,1)	=	f
q(f(a),a)	=	q(1,2)	=	v
....	..	....		....

Portanto, I'' = {p(a),q(a,a),q(f(a),a),...}.

Para avaliar se um conjunto S de cláusulas é insatisfatível, é apenas necessário considerar interpretações sobre o universo de Herbrand de S. Se S for f sob todas as interpretações sob o universo de Herbrand de S, então S é insatisfatível.

Desde que, usualmente, exista um número infinito dessas interpretações, elas devem ser organizadas de maneira sistemática; isso pode ser feito usando uma árvore semântica. Existem muitas maneiras de expressar e de especializar o Teorema de Herbrand, talvez a especialização do Teorema 6.1 seja a mais adequada ao contexto de programação lógica.

Teorema 6.1 Uma contradição pode ser derivada de um conjunto de cláusulas $S \cup \{\neg G\}$ se e somente se puder ser derivada a partir de algum subconjunto finito de instanciações *ground* $G_{(S \cup \{\neg G\})}$.

A palavra *finito* garante a existência de um procedimento efetivo (algoritmo), por meio do qual a contradição pode ser demonstrada e, portanto, indiretamente, mostrado que G é uma conseqüência lógica de S. Em outros trabalhos, este teorema foi reescrito em termos de insatisfatibilidade em vez de derivabilidade e se torna:

Teorema 6.2 Um conjunto de cláusulas $S \cup \{\neg G\}$ é insatisfatível se e somente se algum subconjunto finito de $G_{(S \cup \{\neg G\})}$ for insatisfatível.

Exemplo 6.15 Seja $S = \{p(X), \neg p(f(a))\}$.

S é insatisfatível. Portanto, pelo Teorema de Herbrand, existe um conjunto insatisfatível S' de instâncias *ground* de cláusulas em S. Pode ser verificado que um desses conjuntos é: $S' = \{p(f(a)), \neg p(f(a))\}$.

Exemplo 6.16 Seja $S = \{\neg p(X) \lor q(f(X),X), p(g(b)), \neg q(Y,Z)\}$.

S é insatisfatível. Um dos conjuntos insatisfatíveis de instâncias *ground* de cláusulas em S é:

$$S' = \{\neg p(g(b)) \lor q(f(g(b)),g(b)), p(g(b)), \neg q(f(g(b)),g(b))\}.$$

Um dos aspectos mais importantes da interpretação de Herbrand é que não há mais a necessidade de considerar o universo de todas possíveis interpretações e usar os objetos de cada universo de toda a forma possível, dado que as interpretações de Herbrand representam todas as outras. Todas as interpretações de Herbrand compartilham o mesmo universo – o de Herbrand. Isso significa que é apenas necessário considerar instâncias obtidas, substituindo variáveis por elementos do universo de Herbrand.

Com um pouco de cuidado é possível garantir que nenhuma das possíveis conjunções de instâncias de cláusulas seja omitida no teste. Neste caso, o teste eventualmente irá descobrir se uma meta G é conseqüência lógica de um conjunto de cláusulas; entretanto, a busca por todas as possíveis instâncias pode continuar para sempre. Os seguintes casos podem acontecer:

1. G é uma conseqüência lógica de S. Neste caso, a busca irá eventualmente terminar com sucesso, ou seja, uma contradição será encontrada.
2. G não é uma conseqüência lógica de S. Existem dois possíveis subcasos a serem considerados:

   2.1 todas as possíveis instâncias foram geradas sem que tenha sido encontrada uma contradição. Neste caso, G não é uma conseqüência lógica de S;

   2.2 instâncias são geradas continuamente e uma contradição nunca é encontrada. Neste caso, não existe certeza com relação a G ser ou não uma conseqüência lógica de S.

1ª LISTA DE EXERCÍCIOS

1) Considere uma interpretação I e as fórmulas α, β, γ e λ como definidas a seguir:

$$\alpha: (p \rightarrow q);\ \beta: (p \leftrightarrow q);\ \gamma: (\neg p \vee p) \text{ e } \lambda: ((\neg p \vee p) \rightarrow (\neg q \vee p))$$

a) se $I[\alpha] = v$, o que se pode concluir a respeito de $I[p]$ e $I[q]$?
b) se $I[\beta] = v$, o que se pode concluir a respeito de $I[p]$ e $I[q]$?
c) se $I[\gamma] = v$, o que se pode concluir a respeito de $I[p]$ e $I[q]$?
d) se $I[\alpha] = f$, o que se pode concluir a respeito de $I[p]$ e $I[q]$?
e) se $I[\gamma] = f$, o que se pode concluir a respeito de $I[p]$ e $I[q]$?
f) se $I[\beta] = f$, o que se pode concluir a respeito de $I[p]$ e $I[q]$?
g) se $I[\lambda] = v$, o que se pode concluir a respeito de $I[p]$ e $I[q]$?
h) se $I[\lambda] = f$, o que se pode concluir a respeito de $I[p]$ e $I[q]$?
i) se $I[q] = v$, o que se pode concluir a respeito de $I[\alpha]$, $I[\beta]$, $I[\gamma]$ e $I[\lambda]$?
j) se $I[p] = v$, o que se pode concluir a respeito de $I[\alpha]$, $I[\beta]$, $I[\gamma]$ e $I[\lambda]$?

2) Construa a tabela-verdade associada a cada fórmula dada a seguir:
a) $((\neg p \rightarrow q) \vee (r \wedge \neg q)) \leftrightarrow p$
b) $(\neg(p \vee q)) \leftrightarrow (\neg r \rightarrow \neg q)$
c) $(p \wedge q \wedge r) \rightarrow (\neg r \leftrightarrow \neg q)$
d) $(\text{falso} \rightarrow q)$
e) $(\text{falso} \rightarrow p)$
f) $(p \rightarrow \text{falso})$
g) $(p \rightarrow \text{verdade})$
h) $((p \leftrightarrow \neg q) \rightarrow \neg r) \rightarrow (\neg(r \wedge \neg q) \vee p)$
i) $(\text{falso} \rightarrow q) \leftrightarrow (\neg r)$
j) $((p \vee \neg q) \rightarrow r) \rightarrow (s \leftrightarrow \neg q)$
k) $(q \vee r) \rightarrow ((q \vee s) \rightarrow (p \vee s))$
l) $(p \rightarrow r) \rightarrow q$
m) $(p \rightarrow r) \vee q$
n) $((p \rightarrow q) \rightarrow r) \leftrightarrow (p \rightarrow (q \rightarrow r))$
o) $(p \rightarrow (q \rightarrow r)) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r))$
p) $(p \rightarrow q) \rightarrow ((p \wedge (q \wedge r)) \rightarrow (p \wedge (p \vee r)))$

3) Seja I uma interpretação tal que $I[p \rightarrow q] = f$, e J uma interpretação tal que $J[p \rightarrow q] = v$. O que se pode dizer a respeito dos resultados das interpretações a seguir?
a) $I[(p \vee r) \rightarrow (\neg q \vee r)]$

b) $I[(p \wedge r) \rightarrow (q \wedge r)]$

c) $I[(\neg p \vee p) \rightarrow (q \vee p)]$

d) $J[(p \vee r) \rightarrow (q \vee \neg r)]$

e) $J[(p \wedge r) \rightarrow (q \wedge r)]$

f) $J[(\neg p \vee p) \rightarrow (q \vee p)]$

g) $J[(p \wedge q) \leftrightarrow (\neg q)]$

4) As sentenças a seguir estão em língua natural. Escreva cada uma delas usando a linguagem da Lógica Proposicional.

a) Se chove, então as ruas ficam molhadas.

b) João é magro ou Maria não é brasileira.

c) Se Maria estuda bastante, então Maria vai ao cinema.

d) Antonio vai ao cinema se e somente se o filme for uma comédia.

e) Ou Maria irá ao cinema e João não, ou Maria não irá e João irá.

f) Maria tem 10 anos ou se Maria é estudiosa, então é boa aluna.

g) Uma condição necessária para que uma seqüência s convirja é que seja limitada.

h) Se $x > 0$, $x^2 > 0$.

i) Se Carlos é materialista, Carlos é ateu. Se Carlos é ateu, então Carlos é materialista.

j) Se Pedro está no teatro, então Guilherme está no teatro também.

k) Se ele tiver tempo, ele virá.

l) Se você não sair, eu chamo a polícia.

m) Duas crianças têm o mesmo tio se e somente se elas têm a mesma mãe e têm o mesmo pai.

n) Se $i > j$, então $(i-1) > j$ senão $j = 3$.

5) Verifique se a informação dada é suficiente para determinar:

a) $I[(p \rightarrow s) \rightarrow r]$, sabendo que $I[r] = v$.

b) $I[(p \vee r) \vee (s \rightarrow q)]$, sabendo que $I[q] = f$ e $I[r] = v$.

c) $I[((p \vee q) \leftrightarrow (q \wedge p)) \rightarrow ((r \wedge p) \vee q)]$, sabendo que $I[q] = v$.

6) Determine $I[p]$ e $I[q]$, sabendo que:

a) $I[(p \rightarrow q)] = v$ e $I[(p \wedge q)] = f$.

b) $I[(p \leftrightarrow q)] = f$ e $I[(\neg p \vee q)] = v$.

7)Verifique quais das fórmulas a seguir são tautologias, quais são contradições e quais são contingentes. Justifique sua resposta.

a) $((p \wedge q) \wedge r) \leftrightarrow (p \wedge (q \wedge r))$

b) $((p \vee q) \vee r) \leftrightarrow (p \vee (q \vee r))$

c) $(p \wedge p) \leftrightarrow p$

d) $(p \vee p) \leftrightarrow p$

e) $(p \wedge (q \vee r)) \leftrightarrow ((p \wedge q) \vee (p \wedge r))$

f) $(p \vee (q \wedge r)) \leftrightarrow ((p \vee q) \wedge (p \vee r))$

g) $(p \wedge (p \vee q)) \leftrightarrow p$

h) $(p \vee (p \wedge q)) \leftrightarrow p$

i) $p \rightarrow p \vee r$

8) Seja I uma interpretação tal que $I[p] = I[q] = v$ e $I[r] = I[s] = f$. Encontre:

(a) $I[(\neg(p \wedge q) \vee \neg r) \vee ((p \leftrightarrow \neg q) \rightarrow (r \vee \neg s))]$

(b) $I[(p \leftrightarrow r) \wedge (\neg q \rightarrow s)]$

(c) $I[(p \vee (q \rightarrow (r \wedge \neg p))) \leftrightarrow (q \vee \neg s)]$

9) Considere a seguinte afirmativa como verdadeira:

Se Maria for à escola, então Gabriel ou Paula irão, e se Maria não for à escola, então Paula e Guilherme irão.

É possível chegar a alguma conclusão de quem com certeza irá à escola?

10) Verifique se as fórmulas a seguir estão bem-formadas, de acordo com a definição de fórmulas bem-formadas, e mostre todos os passos da verificação e a conclusão final a que chegou.

(a) $(p \vee \neg(q \wedge \text{verdade})) \leftrightarrow \text{falso}$

(b) $((\text{verdade} \vee q) \leftrightarrow \neg(q \wedge p)) \rightarrow (r \wedge \text{falso})$

(c) $(\neg(\text{verdade} \leftrightarrow \neg(q \wedge p)) \rightarrow (\neg\neg p \vee \neg(q \wedge \text{verdade})) \leftrightarrow \text{falso}$

2ª LISTA DE EXERCÍCIOS

1) Verifique, usando a definição de conseqüência lógica, se pode ser escrito:

a) $((\neg p \rightarrow q), (r \wedge \neg q)) \models p \rightarrow r$

b) $((\neg p \rightarrow q) \vee (r \wedge \neg q)) \models p \rightarrow \neg r$

c) $((p \rightarrow q), (r \wedge \neg q)) \models p \rightarrow r$

d) $(\neg(p \vee q)) \leftrightarrow (\neg r \rightarrow \neg q), \neg q \models ((p \wedge \neg q) \vee r)$

e) $(p \wedge q \wedge r) \rightarrow (\neg r \leftrightarrow \neg q), \neg r \models p \rightarrow r$

f) $p \rightarrow (q \vee r), p \models p \wedge q$

g) $\neg p \rightarrow \neg\neg q, \neg\neg\neg p \models q$

h) $(p \wedge q) \rightarrow (r \wedge s), \neg\neg p, q \models s$

i) $p \models (p \vee q) \wedge (p \vee r)$

j) $p, \neg\neg(p \rightarrow q) \models q \vee \neg q$

k) $p \leftrightarrow (q \vee r), q \models p$

l) $p, (p \wedge q) \rightarrow \neg r, \neg r \rightarrow \neg s \models q \rightarrow \neg s$

m) $\neg p \rightarrow p \models p$

n) $p \rightarrow \neg\neg q \models \neg q \rightarrow \neg p$

o) $\neg p \leftrightarrow (\neg q \vee \neg r), r \wedge p \models p$

2) Repita o exercício 1), mas agora usando o resultado estabelecido pelo Teorema 1.1.

3) Considere que:

Se o universo é finito, então a vida é curta.

Se a vida vale a pena, então a vida é complexa.

Se a vida é curta ou complexa, então a vida tem sentido.

A vida não tem sentido.

Verifique, usando regras de inferência e equivalências lógicas:

a) Se o universo é finito e a vida vale a pena, então a vida tem sentido.

b) A vida não é curta.

c) A vida não é complexa ou o universo não é finito.

d) A vida vale a pena se e somente se a vida tem sentido.

4) Dado que:

Eu não como muito ou eu engordo.

Se chove, então a temperatura cai.

Se eu engordo ou a temperatura cai, então assisto TV.

Não assisto TV.

Verifique, usando regras de inferência, conseqüências e equivalências lógicas, se as assertivas a seguir são válidas:

a) Se eu não como muito e chove, então assisto TV.

b) Se a temperatura cai ou eu engordo, então eu não como muito.

5) Identifique os átomos, construa o argumento e verifique a validade para as situações:

a) Se Deus existe, então a vida tem significado.
Deus existe.
Portanto,
A vida tem significado.

b) Deus não existe.
Se Deus existisse, a vida teria significado.
Portanto, a vida não tem significado.

c) Como hoje não é quinta-feira, deve ser sexta-feira.
Logo, hoje é quinta-feira ou sexta-feira.

d) Se hoje for quinta-feira, então amanhã será sexta-feira.
Se amanhã for sexta-feira, então depois de amanhã será sábado.
Conseqüentemente, se hoje for quinta-feira, então depois de amanhã será sábado.

e) Hoje é um fim de semana se e somente se hoje for sábado ou domingo.
Portanto, hoje é um fim de semana, desde que hoje seja sábado.

f) Hoje é um fim de semana se e somente se hoje for sábado ou domingo.
Hoje não é sábado. Hoje não é domingo.
Portanto, hoje não é um fim de semana.

g) Ela não está em casa ou não está atendendo ao telefone.
Mas se ela não está em casa, então ela foi seqüestrada. Se ela não está atendendo ao telefone, ela está correndo algum outro perigo.
Portanto, ou ela foi seqüestrada ou ela está correndo um outro perigo.

6) Prove, usando as regras de inferência e o princípio da substituição, que os argumentos a seguir são válidos.

a) $\alpha \vdash \beta \rightarrow \alpha$

b) $\neg\alpha \rightarrow (\beta \rightarrow \delta), \neg\alpha, \beta \vdash \delta$

c) $\neg\alpha \rightarrow \neg\neg\beta, \neg\neg\neg\alpha \vdash \beta$

d) $\alpha \vdash \alpha \vee \alpha$

e) $\alpha \vdash (\alpha \vee \beta) \wedge (\alpha \vee \delta)$

f) $\alpha \rightarrow \beta, (\alpha \rightarrow \beta) \rightarrow (\beta \rightarrow \alpha) \vdash (\alpha \leftrightarrow \beta)$

7) Encontre uma fórmula equivalente à fórmula $\neg(p \wedge q) \vee r$, onde só ocorra $\rightarrow$.

8) Verifique, justificando quais dos enunciados a seguir são verdadeiros.

a) $\neg(p \wedge q) \vee q$ é uma contradição.

b) $(p \leftrightarrow q \wedge \neg p) \leftrightarrow \neg q$ é uma contradição.

c) Se p for avaliado f então $q \equiv \neg p \vee q$.

d) $p \wedge p$ é insatisfatível.

e) $p \rightarrow p$ é satisfatível.

f) $p \rightarrow p$ é válida

g) $p \vee \neg p$ é válida.

9) Usando tanto o método da tabela-verdade quanto o de manipulação algébrica de fórmulas via equivalência lógica, determine a FNC equivalente a:

a) $p \rightarrow \neg q$

b) $\neg(p \wedge q)$

c) $(p \wedge q) \vee q$

d) $p \wedge \neg(q \vee r)$

e) $\neg(p \wedge (q \vee r))$

f) $p \vee (\neg p \wedge q \wedge r)$

g) $\neg(p \rightarrow q) \vee (p \vee q)$

h) $\neg(p \rightarrow \neg q) \wedge (p \wedge q)$

10) Usando tanto o método da tabela-verdade quanto o de manipulação algébrica de fórmulas via equivalência lógica, determine a FND equivalente a:

a) $\neg p \rightarrow (q \wedge r)$

b) $\neg q \wedge (q \rightarrow r)$

c) $(p \rightarrow q) \vee \neg p$

d) $(\neg p \wedge q) \vee q$

e) $\neg(p \wedge (q \vee r))$

f) $p \vee (q \rightarrow r)) \rightarrow s$

g) $\neg(p \vee q) \wedge (s \rightarrow t)$

h) $\neg(p \wedge q) \wedge (p \vee q)$

3ª LISTA DE EXERCÍCIOS

1) Usando o algoritmo de Wang, prove (passo a passo) se cada uma das expressões lógicas a seguir é (ou não) um teorema do cálculo proposicional (note que as expressões já estão usando o separador do algoritmo de Wang ($\Rightarrow$)).

a) $((\neg p \vee q), (r \wedge \neg q)) \Rightarrow p \rightarrow r$

b) $((\neg p \rightarrow q) \vee (r \wedge \neg q)) \Rightarrow p \rightarrow \neg r, q$

c) $((p \rightarrow q), (r \wedge \neg q)) \Rightarrow p \rightarrow r$

d) $(\neg(p \vee q)) \leftrightarrow (\neg r \rightarrow \neg q), \neg q \Rightarrow ((p \wedge \neg q) \vee r)$

e) $(p \wedge q \wedge r) \rightarrow (\neg r \leftrightarrow \neg q), \neg r \Rightarrow p \rightarrow r$

f) $p \rightarrow (q \vee r), p \Rightarrow p \wedge q$

g) $\neg p \rightarrow \neg\neg q, \neg\neg\neg p \Rightarrow q$

h) $(p \wedge q) \rightarrow (r \wedge s), \neg\neg p, q \Rightarrow (s \vee q) \wedge r$

i) $p \Rightarrow (p \vee q) \wedge (p \vee r)$

j) $p, \neg(p \rightarrow q) \Rightarrow q, \neg q$

k) $p \leftrightarrow (q \vee r), q \models p$

l) $r, (\neg p \wedge q) \leftrightarrow \neg r, \neg r \vee q \rightarrow \neg s \Rightarrow r \rightarrow \neg s, p$

m) $\neg p \rightarrow p \Rightarrow p$

n) $\neg p \rightarrow \neg\neg q, s \wedge \neg p \Rightarrow \neg q \rightarrow \neg p$

o) $p \leftrightarrow (q \vee \neg r), \neg r \wedge p \Rightarrow p, \neg r$

2) Considere as seguintes premissas:

Se o universo é finito, então a vida é curta.
Se a vida vale a pena, então a vida é complexa.
Se a vida é curta ou complexa, então a vida tem sentido.
A vida não tem sentido.

2.1) Prove usando o princípio de resolução (negando a conclusão) se:

a) Se o universo é finito e a vida vale a pena, então a vida tem sentido.
b) A vida não é curta.
c) A vida não é complexa ou o universo não é finito.
d) A vida vale a pena se e somente se a vida tem sentido.

2.2) Repita o exercício 2.1) usando a negação de toda expressão lógica.

3) Considere as seguintes premissas:

Eu não como muito ou eu engordo.
Se chove, então a temperatura cai.

Se eu engordo e a temperatura cai, então assisto TV.
Assisto TV.

3.1) Prove usando o princípio de resolução (negando a conclusão) se:

a) Se eu não como muito e chove, então assisto TV.

b) Se a temperatura cai ou eu engordo, então eu não como muito.

3.2) Repita o exercício 3.1) usando a negação de toda expressão lógica.

4ª LISTA DE EXERCÍCIOS

1) Verifique se as fórmulas booleanas α e β, dadas em cada um dos cinco itens a seguir, são equivalentes, usando o estabelecido na Observação 2.6, que diz: "uma conseqüência óbvia da unicidade da forma normal é que duas formas booleanas são equivalentes se elas têm a mesma forma normal. Suas funções associadas são, conseqüentemente, iguais."

a) $\alpha = x_1' \otimes (x_2 \oplus x_1')$
$\beta = x_1' \otimes x_2$

b) $\alpha = (x_1 \otimes x_2) \otimes (x_3 \oplus (x_1 \otimes x_2)')$
$\beta = x_1 \otimes x_2 \otimes x_3$

c) $\alpha = (x_1 \otimes x_2 \otimes x_3) \oplus x_1$
$\beta = x_1$

d) $\alpha = (x_1 \otimes x_2 \otimes x_3) \oplus (x_3 \otimes x_2)$
$\beta = (x_2 \otimes x_3)$

e) $\alpha = x_1 \oplus (x_1 \otimes x_2) \oplus (x_1 \otimes x_3)$
$\beta = x_1$

2) Considere o conjunto das funções booleanas totais definidas de $\{0,1\}^3 \to \{0,1\}$.

a) Escreva as funções: $f_1, f_{10}, f_{16}, f_{38}, f_{56}, f_{57}, f_{94}, f_{137}, f_{255}$.

b) Qual a função resultante das seguintes operações (considerando as definições de igualdade funcional)?

$f_1 \oplus f_{10}$
$f_{88} \otimes f_{38}$
$(f_{63} \otimes f_{123})'$
$(f_{59} \otimes f_{57}) \oplus (f_{94} \oplus f_{94})$
$(f_{137} \otimes f_{255}) \oplus f_{119}$
$((f_{167} \otimes f_{52}) \oplus f_{19})'$

3) Considere o conjunto das funções booleanas totais definidas de $\{0,1\}^4 \to \{0,1\}$. Escreva as funções: $f_1, f_{10}, f_{16}, f_{38}, f_{56}, f_{57}, f_{94}, f_{137}, f_{255}$.

4) Minimize a forma normal definida pelo conjunto de minterms:

a) $I = \{min_i^4 \mid i \in I\}$, tal que $I = \{0, 1, 5, 6, 10, 12, 14\}$.

b) $I = \{min_i^4 \mid i \in I\}$, tal que $I = \{0, 1, 5, 6, 7, 10, 12, 14\}$.

5) Use o teorema $\alpha = \alpha x \oplus \alpha x'$ para converter a fórmula $x_1x_2 \oplus x_1x_3 \oplus x_2'x_3$ à sua equivalente forma normal disjuntiva. Minimize a forma normal.

6) Minimize a forma normal definida pelo conjunto de minterms:

a) $I = \{min_i^5 \mid i \in I\}$, tal que $I = \{3, 4, 6, 9, 10, 11, 16, 17, 20, 21, 22, 23, 25, 30, 31\}$.

b) $I = \{min_i^5 \mid i \in I\}$, tal que $I = \{0, 1, 3, 4, 7, 16, 22, 23, 24, 25, 30\}$.

7) Usando o algoritmo de Quine-McCluskey, minimize as expressões:

a) $x_1x_2'x_3 \oplus x_1'x_2x_3' \oplus x_1x_2x_3' \oplus x_1x_2x_3$

b) $x_1x_2x_3x_4 \oplus x_1x_2x_3'x_4 \oplus x_1x_2x_3'x_4' \oplus x_1x_2'x_3x_4 \oplus x_1'x_2x_3x_4 \oplus x_1'x_2x_3x_4' \oplus x_1'x_2x_3'x_4 \oplus x_1'x_2'x_3'x_4$

c) $x_1x_2x_3x_4 \oplus x_1x_2'x_3x_4 \oplus x_1'x_2x_3x_4' \oplus x_1x_2'x_3'x_4' \oplus x_1'x_2x_3'x_4' \oplus x_1x_2x_3x_4' \oplus x_1x_2'x_3x_4' \oplus x_1x_2'x_3'x_4$
$\oplus x_1'x_2x_3x_4 \oplus x_1'x_2x_3'x_4 \oplus x_1'x_2'x_3'x_4'$

d) $x_1x_2x_3x_4 \oplus x_1x_2x_3'x_4 \oplus x_1'x_2x_3x_4 \oplus x_1x_2x_3'x_4' \oplus x_1x_2'x_3'x_4 \oplus x_1'x_2x_3x_4' \oplus x_1'x_2x_3'x_4 \oplus x_1x_2'x_3'x_4'$
$\oplus x_1'x_2'x_3x_4' \oplus x_1'x_2'x_3'x_4'$

e) $x_1x_2x_3'x_4 \oplus x_1x_2'x_3'x_4' \oplus x_1'x_2x_3x_4 \oplus x_1'x_2x_3x_4' \oplus x_1'x_2'x_3x_4'$

f) $x_1x_2x_3x_4x_5 \oplus x_1x_2'x_3x_4'x_5 \oplus x_1x_2x_3'x_4x_5 \oplus x_1x_2'x_3x_4x_5 \oplus x_1'x_2x_3x_4'x_5 \oplus x_1x_2x_3x_4'x_5' \oplus x_1'x_2x_3'x_4x_5$
$\oplus x_1'x_2'x_3'x_4x_5 \oplus x_1'x_2'x_3'x_4'x_5 \oplus x_1'x_2'x_3x_4'x_5$

8) Usando mapa de Karnaugh, minimize as expressões de a), b), c), d) e) do exercício 7) desta lista.

5ª LISTA DE EXERCÍCIOS

1) Considere um alfabeto A definido por:

Constantes: {a,b,c}

Variáveis: {X,Y,Z}

Símbolos funcionais: {f/3, g/2, h/1}

Símbolos predicados: {p/3, q/2}

Quantificadores: $\forall$, $\exists$

Conectivos: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$

Símbolos de pontuação: ( ) ,

Verifique quais das expressões lógicas a seguir são fórmulas bem-formadas. Justifique sua resposta com base na definição de fórmula bem-formada (Definição 3.9).

a) $(\exists X(\forall X\ p(X,a,b)))$

b) $(\forall Y(\forall X\ p(X,Y,b)))$

c) $(\forall Y(\exists X\ p(X,Y,b)))$

d) $(\forall X(\exists Y\ p(X,a,Y) \rightarrow q(f(X),g(Y)))))$

e) $p(X,g(a,b),Y) \vee q(h(h(Z)),d) \wedge \neg q(c,d))$

f) $p(q(X,a),b,Y) \rightarrow q(c,d)$

g) $(\forall X(\exists Z(\forall Y(\ p(a,b,X) \rightarrow q(g(X,Z),f(a,b,Y))))))$

h) $\neg h(p(a,b,c)) \vee f(d,d)$

i) $(\exists Y((\exists Z(p(Z,Z,Z) \vee \neg q(Y,Y))) \vee (\forall X\ q(X,X))))$

j) $(\forall X(\exists Y(p(X,Y,f(a,b,Y)) \rightarrow (\exists Z\ p(Z,Z,d)))))$

k) $(\forall X(\exists Y(p(f(a,b,c),Y,X))) \rightarrow (\exists Y(\forall X\ q(h(Y),g(X,Y)))))$

l) $(\forall X(f(X,b,c) \leftrightarrow q(h(X))))$

m) $((a \vee b \vee c) \leftrightarrow (b \vee d))$

n) $(\forall X(q(X,Y) \vee \neg p(f(a,b,X),g(Z,Z,X),h(a))))$

2) Considere um alfabeto A com:

Conectivos: $\neg$, $\wedge$, $\vee$, $\rightarrow$ e $\leftrightarrow$

Quantificadores: $\forall$ e $\exists$

Símbolos de pontuação: ( ) e ,

Constantes: {a,b,c}

Variáveis: {X,Z}

Símbolos funcionais: {f/2,g/3,h/1}

Símbolos predicados: {p/3, q/1}

a) Crie pelo menos 10 termos (explique por que são termos).

b) Crie pelo menos 10 fórmulas bem-formadas (explique por que são wwfs).

c) Crie pelo menos 5 expressões lógicas que não são fórmulas e explique por que.

3) Identifique as variáveis livres e ligas nas fórmulas a seguir, e quais fórmulas são fechadas.

a) $(\forall X(p(X,Y) \rightarrow (\exists Z(q(a,Z,M) \leftrightarrow r(Z,T,X)))))$

b) $(\forall X(\forall Y(\neg p(a) \vee q(X,Y)))) \wedge (\forall X(p(X))) \wedge (\forall Y(q(a,Y) \vee \neg p(Y)))$

c) $(\exists Z(p(Z,a) \leftrightarrow q(Z,Z))) \rightarrow (\forall Y(\exists X(p(a,X) \vee \neg r(X,Y) \vee \neg\, q(Y,b))))$.

d) $(\forall X(\forall Y(\exists Z(r(X,Y) \wedge \neg r(a,Y))) \rightarrow (\forall Z\ s(Z,b,Z))))$

e) $(\forall X((\forall Y(p(X,Y,Z) \wedge \neg r(Y,X) \vee \neg q(X))) \leftrightarrow (\forall Z\ q(Z))))$

4) Construa o fechamento universal e o existencial das fórmulas:

a) $(\forall X(p(X,Y))$

b) $(\forall X(p(X,Y)) \wedge (\exists Y\ q(Y))$

c) $(\forall X(\exists Y(q(X,Y) \vee p(Z))))$

d) $(\forall X(\forall Z(p(X,Y,Z,T) \leftrightarrow q(Z,M))))$

e) $(\exists M(\exists X(\forall Y(p(a,b,Y) \vee \neg q(X,b)))))$

5) Coloque na notação de Kowalski as cláusulas:

a) $(\forall X(\forall Y(\forall Z(\neg p(X,a,b) \vee \neg p(Y,Z) \vee r(X,) \vee s(Z,c)))))$

b) $(\forall M(\forall N(\forall T(\forall Z(\neg p(T,T,Z) \vee \neg p(M,N,Z,T))))))$

c) $(\forall Y(\forall Z(\forall X(p(X,Z,Y) \vee r(X,X) \vee q(Z)))))$

d) $(\forall X(\forall Y(\forall Z(p(X,Y,b) \vee \neg r(a,Z,X) \vee \neg s(X,Z)))))$

e) $(\forall X(\forall Y(\forall Z\ p(X,Y,Z))))$

6ª LISTA DE EXERCÍCIOS

1) Considere uma linguagem λ cujo alfabeto tem os seguintes símbolos:

Constantes: {a,b,c,d}

Variáveis: {X,Y,Z,W}

Símbolos funcionais: {f/1, g/1}

Símbolos predicados: {p/1, q/2,s/1}

Quantificadores: ∀, ∃

Conectivos: ¬, ∧, ∨, →, ↔

Símbolos de pontuação: ( ) ,

Seja I a seguinte interpretação:

Domínio: {1,2}

Atribuição a constantes:

a	b	c	d
1	2	1	2

Atribuição a variáveis:

X	Y	Z	W
2	1	1	2

Atribuição a símbolos funcionais:

f(1)	f(2)	g(1)	g(2)d
2	1	2	1

Atribuição a símbolos predicados

p(1)	p(2)	q(1,1)	q(1,2)d	q(2,1)	q(2,2)	s(1)	s(2)
v	f	f	v	v	f	f	v

Avalie cada uma das fórmulas, a seguir, na interpretação I.

a) (∀X(∀Y(p(X) ∨ q(Y,c) ∨ ¬q(X,a) ∨ s(f(W))))

b) (∃X(∀Z(p(X) → q(Z,c))))

c) (∀W(∃Y(∃Z(p(Y) ∨ q(W,W) ∨ ¬s(Z)))

d) (∀X(∃Y(q(X,Y) ∨ s(X)))) ∨ (∀W q(a,W))

e) (∃X(∃Y((p(X) ∨ s(Y)) → q(X,X)))) ↔ (∀X q(X,W))

f) (∃W p(g(W))) ∨ (∀X(∃Y q(f(X),Y))) ∨ s(g(Z))

g) (∃X(p(X) ↔ q(X,d))) → (∀X(∃W q(g(X),f(W))))

2) Determine, em cada caso, o resultado da aplicação da substituição à fórmula:

	Fórmula	Substituição
a)	gosta(X,pai(Y)	θ = {mãe(Y)/X, maria/Y}
b)	gosta(X,pai(Y))	θ = {mãe(maria)/X, maria/Y}
c)	arvore(t(X,t(Y,Y)))	θ = {t(U,U)/X, U/Y}
d)	p(X,Y,a)	θ = {Z/X, M/Y,K/T,a/N}

3) Dada a substituição:

$$\theta = \{f(a,Y)/X,\ f(b,Z)/Y, c/Z\}$$

determine o menor valor de n> 0 para o qual θ^n seja idempotente.

4) Determine a composição $\theta^* = \theta_1\theta_2$, tal que

$$\theta_1 = \{f(a,Y)/X,\ f(U,Z)/W\}$$

$$\theta_2 = \{f(a,a)/X,\ b/Y,\ c/V\}$$

e confirme que $W\theta^* = (W\theta_1)\,\theta_2$ para W = p(V,X,Y,W).

5) Discuta se as substituições a seguir são válidas e se satisfazem as propriedades de funcionalidade e idempotência.

a) {mãe(Y)/X, maria/Y}

b) {X/X}

c) {2/X, 3/X}

d) {2/X, 2/X}

e) {Y/X, X/Y}

f) {2/X, 2/Y, X/Y}

g) {f(X)/f(Y)}

6) Uma substituição é idempotente se e somente se $\theta\theta = \theta$. Verifique se a seguinte definição é também válida: uma substituição θ é idempotente se e somente se dom(θ) $\cap$ contradomínio(θ) = $\varnothing$.

7) Considerando que δ, σ e θ são substituições, quais das seguintes implicações são verdadeiras?

a) Se $\sigma\theta = \delta\theta$, então $\sigma = \delta$.

b) Se $\theta\sigma = \theta\delta$, então $\sigma = \delta$.

c) Se $\sigma = \delta$, então $\sigma\theta = \delta\theta$.

7ª LISTA DE EXERCÍCIOS

1) Verifique se cada um dos conjuntos de expressões a seguir é unificável. Para aqueles unificáveis, escreva a substituição unificadora e diga se a substituição é a unificadora mais geral.

a) {p(X,g(Y),a), p(Z,M,N), p(c,K,T), p(X1,X2,X3)}

b) {q(a,b), q(M,Z), q(T,A), q(a,N)}

c) {q(g(M),K,L), q(a,b,c), q(N,b,Z)}

d) {p(f(f(g(a))),c,g(b)), p(X,Y,Z)}

e) {r(a,b,f(Z)),r(X,Y,Z), r(b,b,M)}

f) {s(Z1,Z2,f(f(Z4))), s(g(a),M,N,T), r(g(Z),c,d,K)}

2) Classifique as cláusulas a seguir em: definidas, indefinidas, unitárias e meta.

a) definida(X) ∨ indefinida(X) ∨ negativa(X) ∨ ¬clausula(X).

b) ¬temperatura(frio) ∨ ¬temperatura(quente).

c) ¬amigo(X).

d) rico(X) ∨ ¬rico(X) ∨ mesquinho(X).

e) ¬irma(X,Y) ∨ gosta(X,y).

f) ¬gosta(antonio,física) ∨ gosta(antonio,futebol) ∨ ¬gosta(antonio,arroz).

g) ¬gosta(ana,feijoada) ∨ ¬gosta(ana,futebol) ∨ ¬gosta(ana,carnaval).

h) mãe(X,Y) ∨ ¬filha(Y,X) ∨ ¬mulher(X).

i) p(X,Y) ∨ p(Y,Z) ∨ p(Z,M).

3) Identifique quais cláusulas do exercício 2) são cláusulas de Horn.

4) Usando o Procedimento 4.1 (unify), mostre todos os passos da tentativa de unificação (quando bem-sucedida, mostre também todos os passos na obtenção da substituição final que viabiliza a unificação) de:

a) p(a,b,Z) & p(X,K,g(X))

b) q(f(f(b)),M,g(a,b)) & q(M,N,g(Y,Z))

c) p(X,X,g(X,Y)) & p(a,b,K)

d) p(X,X,h(a,b,Z)) & p(X,Y,K)

e) q(a,b) & q(X,c)

f) q(g(a,b,c),h(Z),Z) & q(M,K,P)

g) q(U,V) q(f(W),W)

h) p(U,f(V)) & p(f(W),W)

5) Considere o seguinte conjunto de axiomas, escritos na forma clausal, como:

¬p(X) ∨ q(X) ∨ r(f(X))

$\neg q(Y) \vee s(Y)$

$\neg q(Z) \vee t(Z)$

$\neg r(W) \vee s(W)$

$\neg r(T) \vee u(T)$

$p(g(U)) \vee q(h(U))$

Verifique, usando resolução, se a seguinte assertiva lógica segue logicamente dos axiomas:

$$(\exists M(\exists N((s(M) \wedge t(M)) \vee (u(N) \wedge s(N)))))$$

6) Suponha que sejam válidas as seguintes assertivas:

a) $\neg$cachorro(rex) $\vee$ (late(rex) $\wedge$ morde(rex))

b) todos os terriers são cachorros: $(\forall X\ (\text{terrier}(X) \rightarrow \text{cachorro}(X)))$

c) $(\forall Y(\text{late}(Y) \rightarrow \text{barulhento}(Y)))$

Usando resolução, prove se a seguinte conclusão segue logicamente das assertivas:

$$(\exists Z(\neg \text{terrier}(Z) \vee \text{barulhento}\ (Z)))$$

7) Considere o axioma: $(\forall X(\exists Y\ p(X,Y)))$.

Usando resolução, prove que a expressão lógica a seguir é verdade:

$$(\forall X(\exists Y(\exists Z(p(Z,Y) \wedge p(Y,X)))))$$

8) Dadas as premissas:

(a) $(\forall X(\text{estuda}(X) \rightarrow \text{aprende}(X)))$

(b) $(\forall X(\text{aprende}\ (X) \rightarrow \text{professional}(X)))$

prove que a conclusão a seguir segue das premissas:

$$(\forall X(\text{estuda}\ (X) \rightarrow \text{professional}(X)))$$

9) Usando resolução, prove para cada item a seguir a conclusão indicada. Em cada item estão representadas: as sentenças em língua natural e sua tradução em expressões da Lógica de Predicados. Vários desses argumentos encontram-se descritos em Hegenberg (1976).

a) Todos os poetas são sensíveis. Há poetas; logo, há (pessoas) sensíveis.

$(\forall X(\text{poeta}(X) \rightarrow \text{sensível}(X)))$

$(\exists X\ \text{poeta}(X))$

logo

$(\exists X\ \text{sensivel}(X))$

b) Alguns felinos são tigres. Todos os tigres são belos, logo alguns felinos são belos.

$(\exists X(felino(X) \wedge tigre(X)))$

$(\forall X(tigre(X) \rightarrow belo(X)))$

logo

$(\exists X\ (felino(X) \wedge belo(X)))$

c) Todos os são-carlenses são paulistas; todos os paulistas são brasileiros; logo, todos os são-carlenses são brasileiros.

$(\forall X(sancarlense(X) \rightarrow paulista(X)))$

$(\forall X(paulista(X) \rightarrow brasileiro(X)))$

logo

$(\forall X(sancarlense(X) \rightarrow brasileiro(X)))$

d) Nenhuma baleia é peixe. Moby Dick é baleia; logo, Moby Dick não é peixe.

$(\forall X(baleia(X) \rightarrow \neg peixe(X)))$

baleia(mobydick)

logo

$\neg peixe(mobydick)$

e) Nenhum jogador é pobre. Alguns pobres são alegres; logo, alguns jogadores não são alegres.

$(\forall X(jogador(X) \rightarrow \neg pobre(X)))$

$(\exists X\ (pobre(X) \wedge alegre(X)))$

logo

$(\exists X\ (alegre(X) \wedge \neg jogador(X)))$

f) Há uma pessoa em quem ninguém acredita. Logo, há uma pessoa que não acredita em si mesma.

$(\exists Y(\forall X(\ \neg acredita(X,Y))))$

logo

$\neg acredita(a,a)$

g) Somente os répteis são cobras. Algumas cobras são perigosas. Assim, nem todo réptil deixa de ser perigoso.

$(\forall X(cobra(X) \rightarrow reptil(X)))$

$(\exists X\ (cobra(X) \wedge perigosa(X)))$

assim

$\neg(\forall X(reptil(X) \rightarrow \neg perigoso(X)))$

h) Ou alguns carros são velozes ou não há carro que não seja bom. Ora, não se dá que todos os carros sejam bons. Logo, alguns carros são velozes.

$(\exists X\ (carro(X) \wedge veloz(X))) \vee \neg(\exists X\ (carro(X) \wedge \neg bom(X)))$

$\neg(\forall X(carro(X) \rightarrow veloz(X)))$

logo

$(\exists X\ (carro(X) \wedge veloz(X)))$

i) Todos os moradores do bairro são ciclistas ou pobres. Nem todos os moradores do bairro são pobres. Logo, algum ciclista não é pobre.

$(\forall X(morador(X) \rightarrow (ciclista(X) \vee pobre(X))))$

$(\exists X\ (morador(X) \wedge \neg pobre(X)))$

logo

$(\exists X\ (ciclista(X) \wedge \neg pobre(X)))$

j) Todos os franceses são amáveis. Só os generosos são amáveis. Para ser generoso é preciso ser honesto. Há industriais desonestos. Logo, nem todo industrial é francês.

$(\forall X(frances(X) \rightarrow amavel(X)))$

$(\forall X(amavel(X) \rightarrow generoso(X)))$

$(\forall X(generoso(X) \rightarrow honesto(X)))$

$(\exists X\ (industrial(X) \wedge \neg honesto(X)))$

logo

$(\exists X\ (industrial(X) \wedge \neg frances(X)))$

10) Usando resolução, verifique se é possível deduzir a conclusão: $\neg(\forall X(\neg(blabla(X) \wedge blabla1(X))))$ a partir das premissas:

$(\forall X(\neg(blabla(X) \wedge blabla1(X)))) \rightarrow (\exists Y(\neg(blabla1(Y) \vee blabla2(Y))))$

$(\forall X(blabla1(X) \vee blabla2(X)))$

BIBLIOGRAFIA E REFERÊNCIAS

BERZTISS, A. T. *Data Structures*. New York, USA: Academic Press, 1975.

CHANG, C. L.; LEE, C. -T. *Symbolic Logic and Mechanical Theorem Proving*. New York, USA: Academic Press, 1973.

COELHO, H.; COTTA, J. C. *Prolog by Examples*. Berlin, Germany: Springer-Verlag, 1988.

GRASSMANN, W. K.; TREMBLAY, J. -P. *Logic and Discrete Mathematics – a Computer Science Perspective*. New Jersey, USA: Prentice Hall, 1996.

HEGENBERG, L. *Simbolização no cálculo de predicados*. São Paulo, Brasil: Ed. USP, 1976. v. III.

HEGENBERG, L. *Dedução no cálculo sentencial*. São Paulo, Brasil: Ed. USP, 1977. v. II.

HEGENBERG, L. *Tabelas e argumentos*. São Paulo, Brasil: Ed. USP, 1978, v. I.

HEGENBERG, L. *Dedução no cálculo sentencial*. São Paulo, Brasil: Ed. USP, 1978. v. IV.

HOGGER, C. J. *Essentials of Logic Programming*. New York, USA: Oxford University Press, 1991.

HUTH, M.; RYAN, M. *Logic in Computer Science – Modelling and Reasoning about Systems*. Cambridge, England: Cambridge University Press, 2004.

KOWALSKI, R. A. *Logic for Problem Solving*. Elsevier-North Holland, London, England: Artificial Intelligence Series, 1979. v. 7.

MONARD, M. C.; NICOLETTI, M. C. Método sintático de prova de teoremas – algoritmo de Wang, *Notas do ICMSC-USP*, n. 62, 64 p., 1990.

MONARD, M. C.; NICOLETTI, M. C.; NOGUCHI, R. H. O cálculo proposicional: uma abordagem voltada à compreensão da linguagem Prolog – Versão 1.0. *Notas Didáticas do ICMSC-USP*, n. 5, 68 p., 1992.

NIENHUYS-CHENG, S. H.; WOLF, H. *Foundations of Inductive Logic Programming*. Heildelberg, Germany: Springer-Verlag, 1997.

NILSSON, N. J. *Principles of Artificial Intelligence*. Palo Alto, CA, USA: Tioga Publishing Co., 1972.

NILSSON, N. J.; MALUSZYNSKI, J. *Logic, Programming and Prolog*. New York, USA: John Wiley & Sons, 1990.

NOLT, J.; ROHATYN, D. *Lógica*. São Paulo, Brasil: McGraw-Hill, 1991. (Coleção Schaum.)

SILVA, F. S. C.; FINGER, M.; MELO, A. C. V. *Lógica para computação*. São Paulo, Brasil: Ed. Thomson, 2006.

SINGH, A. *Logics for Computer Science*. New Delhi, India: Prentice-Hall of India, 2004.

SOUZA, J. N. *Lógica para ciência da computação*. Rio de Janeiro, Brasil: Editora Campus, 2002.

WANG, H. Towards Mechanical Mathematics. *IBM Journal of Research and Development*, v. 4, p. 2-11, 1960.

WANG, H. *A Survey of Mathematical Logic*. NewYork, USA: North-Holland Publishing Company, 1964.