# Noções de Lógica Matemática

Domingos Moreira Cardoso
Maria Paula Carvalho

Janeiro, 2007

# Índice

# 1 Introdução

Estas notas têm por objectivo o estabelecimento de um primeiro contacto entre os curiosos da computação simbólica e as técnicas matemáticas que a fundamentam. Pretende-se justificar a generalidade dos procedimentos que estão na base da grande maioria dos sistemas computacionais especialmente desenvolvidos para a obtenção de deduções automáticas. Entende-se que a melhor ajuda que se pode dar a alguém que pretende utilizar e/ou aprofundar a utilização deste tipo de instrumentos computacionais, consiste em desvendar os princípios que lhe estão na base e, deste modo, quebrar os mitos muitas vezes associados a certas designações, algo controversas como Inteligência Artificial, Sistemas Periciais, Demonstração Automática de Teoremas, etc.

Esta notas são uma adaptação de parte do texto com o título "Programação em Lógica e Demonstração Automática de Teoremas" [1], publicado em 1995 na colecção Cadernos de Matemática do Departamento de Matemática da UA, e têm como destinatários os estudantes do primeiro ano da disciplina de Matemática Discreta.

# 2 Noções Fundamentais de Lógica Matemática

*Se nada quisermos omitir, estaremos condenados a uma insuportável prolixidade. Quase sempre a linguagem deixa adivinhar as relações lógicas sem as expressar senão de uma forma alusiva.*

**Frege**

## 2.1 Uma introdução ao cálculo Proposicional

A Lógica Matemática tem origem no matemático britânico George Boole (1815-1964), cujo livro: "An Investigation of the Laws of Thought" publicado em Londres em 1854, mostra como o valor verdadeiro e/ou falso das proposições pode ser manipulado de forma algébrica.

A Lógica Proposicional utiliza proposições, também designadas por átomos ou fórmulas atómicas (denotadas geralmente por letras minúsculas) e cinco conexões : $\neg$ (não), $\wedge$ (e), $\vee$ (ou), $\Rightarrow$ (se... então), $\Leftrightarrow$ (sse) para representar as suas fórmulas bem formadas (fbf) as quais se definem recursivamente do seguinte modo:

**Definição 2.1** *1. Um átomo é uma fbf.*

*2. Se $G$ é uma fbf então $(\neg G)$ é uma fbf.*

*3. Se $G$ e $H$ são fbf então $G \wedge H$, $G \vee H$, $G \Rightarrow H$ e $G \Leftrightarrow H$ são fbf.*

*4. Todas as fbf são geradas por aplicação das regras 1.,2. e 3. e só por estas.*

As cadeias de símbolos $(p \Rightarrow)$ e $(P\vee)$ são exemplos de fórmulas sintaticamente incorrectas, ou seja, fórmulas que não são fbf. As fbf são as fórmulas que fazem sentido gramaticalmente. A pontuação básica utilizada consiste nos parentesis.

Ao longo do texto, diremos que uma fórmula é verdadeira (falsa) quando é avaliada em verdadeiro (falso), i.e., quando o seu valor lógico é verdadeiro (falso).

Considerando a fórmula

$$G \, : \, ((p \Rightarrow q) \wedge p) \Rightarrow q$$

e tendo em conta que os átomos desta fórmula são $p$ e $q$, tem-se que $G$ admite $2^2 = 4$ interpretações a saber:

| $p$ | $q$ | $p \Rightarrow q$ | $(p \Rightarrow q) \wedge p$ | $((p \Rightarrow q) \wedge p) \Rightarrow$ q |
|---|---|---|---|---|
| V | V | V | V | V |
| V | F | F | F | V |
| F | V | V | F | V |
| F | F | V | F | V |

Concluí-se assim que $G$ é verdadeiro qualquer que seja a interpretação feita de entre as quatro possíveis. As fórmulas com estas características (fórmulas que são verdadeiras quaisquer que sejam as interpretações) designam-se por tautologias.

Por sua vez a fórmula

$$G \, : \, ((p \Rightarrow q) \wedge (p \wedge \neg q))$$

é falsa qualquer que seja a interpretação. As fórmulas em que tal acontece (fórmulas que são falsas qualquer que seja a interpretação) designam-se por inconsistentes ou contraditórias.

**Definição 2.2** *Uma fbf diz-se válida se e só se é verdadeira sobre qualquer das suas possíveis interpretações. Uma fbf diz-se não válida (ou inválida) se e só se não é válida.*

*Uma fbf diz-se inconsistente se e só se é falsa qualquer que seja a interpretação.*

*Uma fbf diz-se consistente se e só se não é inconsistente.*

**Exercício 2.1** *Mostre que*

1. $(p \wedge \neg p)$ *é inconsistente; conclua que é, também, inválida.*

2. $(p \vee \neg p)$ *é válida; conclua que é, também, consistente.*

3. $(p \Rightarrow \neg p)$ *é inválida, ainda que seja consistente.*

Quando uma fórmula $F$ é verdadeira para uma interpretação $I$, então dizemos que $I$ satisfaz $F$. Inversamente se uma fórmula $F$ é falsa para uma dada interpretação $I$ , então dizemos que $I$ não satisfaz $F$.

Seja $F$ o conjunto de fbf, $F_n$ $(F_n \in F)$ uma fórmula composta por $n$ átomos e $T = \{V, F\}$, então $F_n$ induz uma função $F_n : \; T^n \longrightarrow T$ , para a qual se verifica que

$F_n$ é uma tautologia sse

$$\begin{array}{rccl} F_n : & T^n & \longrightarrow & T \\ & I & \mapsto & F_n(I) = V \end{array}$$

$F_n$ é uma contradição sse

$$\begin{array}{rccl} F_n : & T^n & \longrightarrow & T \\ & I & \mapsto & F_n(I) = F \end{array}$$

Uma vez que na Lógica Proposicional o número de interpretações de uma dada fórmula é finito, para valores de $n$ aceitáveis, podemos decidir quando é que uma fórmula do Cálculo Proposicional é ou não válida, por exame exaustivo de todas as suas possíveis interpretações.

**Definição 2.3** *Duas fórmulas $F$ e $G$ dizem-se equivalentes (ou $F$ é equivalente a $G$),denotando-se por $F \equiv G$, sse ambas têm a mesma tabela de verdade.*

**Exercício 2.2** *Mostre que são equivalentes as fórmulas:*

$$F: \; p \Rightarrow q \tag{1}$$

$$G: \; \neg p \vee q \tag{2}$$

**Definição 2.4** *Um literal é um átomo ou a negação de um átomo.*

*Dois literais dizem-se complementares quando um é a negação do outro.*

**Definição 2.5** *Uma fórmula $F$ diz-se na forma normal conjuntiva sse $F \equiv \bigwedge_{i=1}^{n} F_i$, com $n \geq 1$, onde cada $F_i \in \{F_1, F_2, \cdots, F_n\}$ é uma disjunção de literais.*

**Definição 2.6** *Uma fórmula $F$ diz-se na forma normal disjuntiva sse $F \equiv \bigvee_{i=1}^{n} F_i$, com $n \geq 1$, onde cada $F_i \in \{F_1, F_2, \cdots, F_n\}$ é uma conjunção de literais.*

**Exemplo 2.1** *Sendo $p, q, r$ fórmulas atómicas,*

$F_1 :\ (p \vee \neg q \vee r) \wedge (\neg p \vee q)$ *está na forma normal conjuntiva;*

$F_2 :\ (p \wedge r) \vee (\neg p \wedge q)$ *está na forma normal disjuntiva.*

**Exercício 2.3** *Sendo $p, q, r, s$ fórmulas atómicas, e $F : \neg p \vee (q \wedge r) \Rightarrow \neg r$*

1. *Obtenha a forma normal disjuntiva de $F$;*
2. *Obtenha a forma normal conjuntiva de $F$.*

Uma teoria numa linguagem é um conjunto de fórmulas da linguagem aceites como axiomas. Os teoremas da teoria são as fórmulas obtidas, pelas regras de inferência, a partir dos axiomas. Assim, para se ter uma teoria é necessário:

- uma linguagem (vocabulário e gramática);
- um sistema de postulados (axiomas ou permissas);
- regras de inferência.

Um sistema de postulados deve ser compatível (consistente ou não contraditório) e tanto quanto possível económico, ou seja, reduzido ao conjunto de postulados independentes (dado que os que não são se podem derivar desse conjunto). Ao retirar-se um dos postulados independentes enfraquece-se o correspondente sistema de postulados, ficando empobrecido em compreensão mas enriquecido em extensão, uma vez que se abrem certas possibilidades que o postulado retirado excluía. Se reforçamos o número de postulados independentes compatíveis, então a certo momento depararemos com um obstáculo, em que a adição de um postulado independente (seja ele qual for) torna o sistema contraditório. Nestas condições um tal sistema diz-se saturado (um exemplo de um sistema saturado é a geometria euclideana).

**Definição 2.7** *Dadas as fórmulas* $F_1, F_2, \cdots, F_n$ *e uma fórmula* $G$*, dizemos que* $G$ *é consequência lógica de* $F_1, F_2, \cdots, F_n$ *(também se diz que* $G$ *se obtém logicamente de* $F_1, F_2, \cdots, F_n$*), escreve-se* $F_1, F_2, \cdots, F_n \models G$*, sse para toda a interpretação* $I$ *na qual* $F_1 \wedge F_2 \wedge \cdots \wedge F_n$ *é verdadeira,* $G$ *também é verdadeira (*$F_1, F_2, \cdots, F_n$ *designam-se por axiomas, postulados, hipóteses ou permissas de* $G$*).*

**Teorema 2.1** *Dadas as fórmulas* $F_1, F_2, \cdots, F_n$ *e uma fórmula* $G$*,* $G$ *é consequência lógica de* $F_1, F_2, \cdots, F_n$ *sse* $((F_1 \wedge F_2 \wedge \cdots \wedge F_n) \Rightarrow G)$ *é uma fórmula válida (ou seja, é uma tautologia).*

*Demonstração.*
$(\Rightarrow)$ Suponhamos que $G$ é consequência lógica de $F_1, F_2, \cdots, F_n$ e seja $I$ uma interpretação arbitrária. Se $(F_1 \wedge F_2 \wedge \cdots \wedge F_n)$ é verdadeira para $I$, por definição de consequência lógica, então $G$ é verdadeira para $I$. Logo $((F_1 \wedge F_2 \wedge \cdots \wedge F_n) \Rightarrow G)$ é verdadeira para $I$.

Por outro lado, se $(F_1 \wedge F_2 \wedge \cdots \wedge F_n)$ é falsa para $I$ então $((F_1 \wedge F_2 \wedge \cdots \wedge F_n) \Rightarrow G)$ é verdadeira (Recordar a tabela de verdade da implicação). Assim, qualquer que seja a interpretação $I$, a fórmula $((F_1 \wedge F_2 \wedge \cdots \wedge F_n) \Rightarrow G)$ é verdadeira.

$(\Leftarrow)$ Suponhamos que $((F_1 \wedge F_2 \wedge \cdots \wedge F_n) \Rightarrow G)$ é uma fórmula válida. Então para cada interpretação $I$, se $(F_1 \wedge F_2 \wedge \cdots \wedge F_n)$ é verdadeira em $I$, $G$ também é verdadeira em $I$. Logo $G$ é consequência lógica de $F_1, F_2, \cdots, F_n$. $\square$

Se $G$ é consequência lógica de $F_1, F_2, \cdots, F_n$, a fórmula $((F_1 \wedge F_2 \wedge \cdots \wedge F_n) \Rightarrow G)$ é designada por teorema, e $G$ é designado por conclusão do teorema.

Muitas vezes para concluirmos que $G$ é consequência lógica de $F_1, F_2, \cdots, F_n$ (ou seja, que a fórmula $((F_1 \wedge F_2 \wedge \cdots \wedge F_n) \Rightarrow G)$ é um teorema), começamos por concluir que $F_{n+1}$ é consequência lógica de $F_1, F_2, \cdots, F_n$, que $F_{n+2}$ é consequência lógica de $F_1, F_2, \cdots, F_n, F_{n+1}$, etc., até que concluímos que $G$ é consequência lógica de $F_1, F_2, \cdots, F_n, F_{n+1}, \cdots, F_{n+m}$, o que nos permite deduzir o inicialmente pretendido, ou seja, que $G$ é consequência lógica de $F_1, F_2, \cdots, F_n$. A sequência de conclusões intermédias constituem o que vulgarmente se designa por demonstração do teorema $((F_1 \wedge F_2 \wedge \cdots \wedge F_n) \Rightarrow G)$.

Note-se que sendo $F_{n+1}$ consequência lógica de $F_1, F_2, \cdots, F_n$, sendo $F_{n+2}$ é consequência lógica de $F_1, F_2, \cdots, F_n, F_{n+1}$, etc, e sendo $F_{n+m}$ consequência lógica de $F_1, F_2, \cdots, F_n, F_{n+1}, \cdots, F_{n+m-1}$ $(m \geq 1)$, então sempre que $(F_1 \wedge F_2 \wedge \cdots \wedge F_n)$ é verdadeira, vem que $F_{n+1}, \cdots, F_{n+m}$ também é verdadeira. Logo, se $((F_1 \wedge F_2 \wedge \cdots \wedge F_{n+m}) \Rightarrow G)$ é uma tautologia, então $((F_1 \wedge F_2 \wedge \cdots \wedge F_n) \Rightarrow G)$ é também é uma tautologia e vice-versa.

**Teorema 2.2** *Dadas as fórmulas $F_1, F_2, \cdots, F_n$ e uma fórmula $G$, $G$ é consequência lógica de $F_1, F_2, \cdots, F_n$ sse $(F_1 \wedge F_2 \wedge \cdots \wedge F_n \wedge \neg G)$ é inconsistente.*

*Demonstração.*

Pelo Teorema 2.1, $G$ é consequência lógica de $(F_1, F_2, \cdots, F_n)$ sse a fórmula

$$((F_1 \wedge F_2 \wedge \cdots \wedge F_n) \Rightarrow G)$$

é válida. Então $G$ é consequência lógica de $(F_1, F_2, \cdots, F_n)$ sse a fórmula

$$\neg((F_1 \wedge F_2 \wedge \cdots \wedge F_n) \Rightarrow G)$$

é inconsistente (ou seja, é falsa qualquer que seja a interpretação).

$$\begin{aligned}\neg((F_1 \wedge F_2 \wedge \cdots \wedge F_n) \Rightarrow G) &\equiv \neg(\neg(F_1 \wedge F_2 \wedge \cdots \wedge F_n) \vee G)\\ &\equiv \neg(\neg(F_1 \wedge F_2 \wedge \cdots \wedge F_n)) \wedge \neg G\\ &\equiv F_1 \wedge F_2 \wedge \cdots \wedge F_n \wedge \neg G.\end{aligned}$$

□

O símbolo "≡" deve ser entendido do seguinte modo: dadas duas fórmulas $F$ e $G$, $F \equiv G$ significa que para as mesmas interpretações , $F$ e $G$ tomam o mesmo valor lógico (verdadeiro ou falso).

Este teorema tem particular importância pois permite concluir que, mostrar que uma fórmula particular é consequência lógica de um conjunto finito de fórmulas, é equivalente a mostrar que uma certa fórmula é inconsistente.

Segue-se um exemplo de aplicação dos Teoremas 2.1 e 2.2.

**Exemplo 2.2** *Dadas as fórmulas* $F_1 : (p \Rightarrow q)$, $F_2 : \neg q$ *e* $G : \neg p$, *provar que* $G$ *é uma consequência lógica de* $F_1$ *e* $F_2$.

*Resolução.*

Para a obtenção desta prova poder-se-ía actuar das uma de três formas a seguir indicadas:

1. Utilizar a tabela de verdade associada às fórmulas $((p \Rightarrow q) \wedge \neg q)$ e $(\neg p)$ e verificar que $G$ é verdadeira sempre que $((p \Rightarrow q) \wedge \neg q)$ é verdadeira.

2. Utilizar o Teorema 2.1 e avaliar a tabela de verdade da fórmula
$$((p \Rightarrow q) \wedge \neg q) \Rightarrow (\neg p).$$

3. Utilizar o Teorema 2.2 e avaliar a tabela de verdade da fórmula
$$((p \Rightarrow q) \wedge \neg q) \wedge p.$$

## 2.2 Alguns tópicos sobre lógica de 1ª ordem

Na lógica proposicional os elementos básicos (as unidades mais simples) são os átomos, os quais representam sentenças declarativas que podem ser verdadeiras ou falsas mas nunca ambas as coisas.

Existem, no entanto, muitas ideias que não podem ser tratadas deste modo.

**Exemplo 2.3** *Todo o homem é mortal. Uma vez que Confucios é homem então é mortal.*

*p: "todo o homem é mortal";*

*q: "Confucios é homem";*

*r: "Confucios é mortal".*

Embora o nosso raciocínio nos leve a concluir que $r$ é consequência lógica de $p$ e $q$, dentro da metodologia de trabalho do Cálculo Proposional, $r$ não é consequência lógica de $p$ e $q$.

Relativamente ao Cálculo Proposional, a Lógica de 1ª ordem lida com mais três noções lógicas designadas por

*termos, predicados e quantificadores.*

Na Lógica de 1ª ordem podem-se usar símbolos de função, por exemplo *plus(x,y)*(para significar $x + y$) e podem-se utilizar relações, como por exemplo *greater (x,y)* (para significar $x > y$), etc.

Para construirmos um átomo na Lógica de 1ª ordem, utilizamos, em geral, os seguintes símbolos:

(a) símbolos individuais de variáveis e constantes (nomes de objectos);

(b) símbolos de funções;

(c) símbolos de predicados.

**Definição 2.8** *Os termos da Lógica de 1ª ordem definem-se recursivamente da seguinte forma:*

1 *Uma constante é um termo;*

2 *uma variável é um termo;*

3 *se* $f$ *é um símbolo de uma função com* $n$ *argumentos e,* $t_1, t_2, \cdots, t_n$ *são termos, então* $f(t_1, t_2, \cdots, t_n)$ *é um termo;*

4 *todos os termos são gerados por aplicação da regras* 1, 2 *e* 3 *e só por estas.*

De acordo com esta Definição são exemplos de termos:

12

$y$

adicionar(adicionar(x,1),x)

pai_de(luisa)

Um predicado é uma aplicação que a uma dada lista de constantes faz corresponder o valor verdadeiro ou falso, como seja, *greater(5,3)* (é verdadeiro), *greater(4,7)* (é falso).

**Definição 2.9** *Se* $P$ *é um predicado com* $n$ *argumentos e* $t_1, t_2, \cdots, t_n$ *são termos, então* $P(t_1, t_2, \cdots, t_n)$ *é um átomo. Nenhuma outra expressão pode ser um átomo.*

Uma vez que os átomos estão definidos, podemos utilizar as mesmas conexões lógicas, referidas no Cálculo Proposicional, para construirmos fórmulas, acrescentadas dos respectivos quantificadores universal e existencial (dado que se consideram variáveis). São exemplos deste tipo de fórmulas as seguintes:

$(\forall x)(Q(x) \Rightarrow R(x))$.

$(\exists x)(P(x))$.

$(\exists x)(\exists y)(Maior(x, y))$.

Um conceito importante relacionado com os quantificadores, é o de alcance (scope) do quantificador, o qual corresponde à parte da fórmula em que está integrado (sobre a qual actua). Por exemplo, o alcance do quantificador universal na fórmula $(\forall x)(\exists y)(M(x, y))$ é a fórmula $(\exists y)(M(x, y))$ e o alcance do quantificador existencial é a fórmula $(M(x, y))$.

**Definição 2.10** *Uma ocorrência de uma variável numa fórmula é ligada ("bound") sse a ocorrência está dentro do alcance do quantificador usado nessa variável. Uma ocorrência de uma variável numa fórmula é livre sse a ocorrência dessa variável não é ligada.*

**Definição 2.11** *Uma variável é livre numa fórmula se no mínimo uma ocorrência dela na fórmula é livre. Uma variável é ligada sse no mínimo uma ocorrência dela é ligada.*

Na fórmula $(\forall x)(P(x, y))$, $x$ é ligada e $y$ é livre. Note-se, no entanto, que uma variável pode ser livre e ligada numa mesma fórmula, por exemplo, na fórmula:

$$(\forall x)(P(x, y) \vee (\forall y)(Q(y))$$

$y$ é uma variável livre e ligada.

**Definição 2.12** *As fórmulas bem formadas (fbfs), da Lógica de $1^a$ ordem, são definidas recursivamente do seguinte modo:*

1 *um átomo (ou fórmula atómica) é uma fbf;*

2 *se $F$ e $G$ são fbfs, então $\neg(F)$, $F \vee G$, $F \wedge G$, $(F \Rightarrow G)$ e $(F \Leftrightarrow G)$ são fbfs;*

3 *se $F$ é uma fbf e $x$ é uma variável em $F$ então $(\forall x)F$ e $(\exists x)F$ são fbfs;*

4 *as fbfs são geradas somente por aplicação de um número finito de vezes de* 1, 2 *e* 3.

**Exercício 2.4** *Exprima por meio de fórmulas as seguintes afirmações:*

1. *Toda a gente gosta de alguém.*
2. *Todo o ser vivo que não é animal é vegetal.*
3. *Todos os números racionais são números reais.*
4. *Existem números primos.*
5. *O conjunto dos números reais é infinito.*

## 2.3 Interpretação das Fórmulas na Lógica de 1ª ordem

Enquanto na Lógica Proposicional as interpretações ficam completamente determinadas pelas tabelas de verdade dos átomos, na Lógica de 1ª ordem, a existência de variáveis obriga a outras considerações.

**Definição 2.13** *Uma interpretação de uma fórmula $F$, na lógica de 1ª ordem, consiste num domínio não vazio $D$ e numa associação de valores para cada constante, símbolo de função e símbolo de predicado que ocorra em $F$, conforme segue:*

1 *Para cada constante associamos um elemento de $D$;*

2 *Para cada símbolo de função com $n$ argumentos associamos uma função de $D^n$ em $D$;*

3 *Para cada símbolo de predicado com $n$ argumentos associamos uma função de $D^n$ em $\{V, F\}$.*

Às vezes, ao referirmo-nos a $D$ falamos numa interpretação da fórmula $F$ sobre $D$, e quando avaliamos o valor verdadeiro da fórmula numa interpretação sobre $D$, $\forall x$ significa "para todo o $x$ em $D$" e $\exists x$ significa "existe $x$ em $D$".

Para qualquer interpretação de uma fórmula sobre um domínio $D$, a fórmula pode ser avaliada em V ou F, de acordo com as seguintes regras:

1. se os valores verdadeiros ou falsos das fórmulas $G$ e $H$ estão avaliados, então os valores verdadeiros ou falsos das fórmulas $\neg G$, $(G \wedge H)$, $(G \vee H)$, $(G \Rightarrow H)$ e $(G \Leftrightarrow H)$ ficam também avaliadas;

2. $(\forall x)G$ é avaliado em V se $G$ é avaliada em V para todas as concretizações possíveis de $x$ em $D$, caso contrário o seu valor é falso;

3. $(\exists x)G$ é avaliado em V se $G$ é avaliada em V para pelo menos uma concretização de $x$ em $D$. Caso contrário o seu valor é falso.

   Note-se que qualquer fórmula que contenha variáveis livres não pode ser avaliada, a menos que se introduza uma função que atribua valores em $D$ às variáveis livres.

**Exemplo 2.4** *Sejam as fórmulas* $(\forall x)\,(P(x))$ *e* $(\exists x)\,(\neg P(x))$, *e considere-se a seguinte interpretação*

| $D = \{1, 2\}$ | | |
|---|---|---|
| $P(1) = V$ | *e* | $P(2) = F.$ |

*Para esta interpretação, com facilidade se conclui que a 1ª fórmula é falsa e a 2ª é verdadeira.*

**Exemplo 2.5** *Considere-se a fórmula* $(\forall x)(P(x) \Rightarrow Q(f(x), a))$, *na qual se tem uma constante* $\underline{a}$, *um símbolo de função com um argumento,* $\underline{f}$, *um símbolo de predicado com um argumento* $\underline{P}$ *e um símbolo de predicado com dois argumentos,* $\underline{Q}$ *e considere-se ainda a seguinte interpretação* $I$*:*

| | | | | | | |
|---|---|---|---|---|---|---|
| *Domínio* | | | | $D = \{1, 2\}$ | | |
| *Afectação para* $\underline{a}$ | | | | $a = 1$ | | |
| *Afectações para* $\underline{f}$ | | | $f(1) = 2$ | | $f(2) = 1$ | |
| *Afectações de* $\underline{P}$ *e* $\underline{Q}$ | $P(1)$ | $P(2)$ | $Q(1,1)$ | $Q(1,2)$ | $Q(2,1)$ | $Q(2,2)$ |
| | $F$ | $V$ | $V$ | $V$ | $F$ | $V$ |

Com esta interpretação, para $x = 1$, a fórmula $P(x) \Rightarrow Q(f(x), a)$, vem

$$(P(1) \Rightarrow Q(f(1), a)) \equiv (P(1) \Rightarrow Q(2, 1)) \equiv (F \Rightarrow F) \equiv V.$$

Com esta interpretação, para $x = 2$, a fórmula $P(x) \Rightarrow Q(f(x), a)$, vem

$$(P(2) \Rightarrow Q(f(2), a)) \equiv (P(2) \Rightarrow Q(1, 1)) \equiv (V \Rightarrow V) \equiv V.$$

Uma vez que a fórmula $P(x) \Rightarrow Q(f(x), a)$ é verdadeira para todas as concretizações de $x$ no domínio $D$, conclui-se que $(\forall x)(P(x) \Rightarrow Q(f(x), a))$ é verdadeira para a interpretação $I$ (acima referida).

**Definição 2.14** *Uma fórmula $G$ é consistente sse existe uma interpretação $I$ tal que $G$ é avaliada em $V$ para $I$. Se uma fórmula $G$ toma o valor $V$ numa interpretação $I$, dizemos que $I$ é um modelo de $G$ e que $I$ satisfaz $G$.*

**Definição 2.15** *Uma fórmula $G$ é inconsistente sse não existe uma interpretação de $G$ que satisfaça $G$. Uma fórmula $G$ diz-se válida sse toda a interpretação de $G$ satisfaz $G$.*

**Definição 2.16** *Uma fórmula $G$ é uma consequência lógica das fórmulas $F_1, F_2, \cdots, F_n$ sse para toda a interpretação $I$, se $\bigwedge_{i=1}^{n} F_i$ é verdadeira em $I$ então $G$ também é verdadeira.*

Os Teoremas 2.1 e 2.2 sobre validade (inconsistência) e consequência lógica de Cálculo Proposicional são também válidos para a Lógica de 1ª ordem.

A Lógica de 1ª ordem pode ser considerada como uma extensão da Lógica Proposicional. Quando uma fórmula da Lógica de 1ª ordem não contém variáveis nem quantificadores, pode ser tratada como uma fórmula da Lógica Proposicional.

## 2.4 Formas Normais Prenex na Lógica de 1ª ordem

A razão que leva à consideração de formas normais "prenex" tem a ver com a necessidade de simplificação dos procedimentos de demonstração.

**Definição 2.17** *Uma fórmula $F$ da Lógica de 1ª ordem diz-se na forma normal prenex sse a fórmula $F$ está na forma:*

$$(Q_1 x_1)(Q_2 x_2) \cdots (Q_n x_n) M$$

*onde $(Q_i x_i)$, para $i = 1, 2 \cdots, n$, identifica $(\forall x_i)$ ou $(\exists x_i)$ e $M$ é uma fórmula que não contém quantificadores.*

**Exemplo 2.6** *Exemplo de uma fórmula na forma normal prenex:*

$$(\forall x)(\forall y)(\exists z)(Q(x, y) \Rightarrow R(z)).$$

Sendo $F$ uma fórmula que contém uma variável livre $x$, a qual vamos identificar por $F[x]$ e sendo $G$ uma fórmula que não contém $x$, a transformação de fórmulas da Lógica de 1ª ordem na forma normal disjuntiva prenex pode fazer-se de acordo com o seguinte procedimento:

**Passo 1** Eliminar as conexões $\Rightarrow$ e $\Leftrightarrow$, fazendo

$$F \Leftrightarrow G \equiv (F \Rightarrow G) \wedge (G \Rightarrow F) \tag{3}$$

$$F \Rightarrow G \equiv \neg F \vee G. \tag{4}$$

**Passo 2** Aplicação das Leis de Morgan e dupla negação

$$\neg(\neg F) \equiv F \tag{5}$$
$$\neg(F \vee G) \equiv \neg F \wedge \neg G \tag{6}$$
$$\neg(F \wedge G) \equiv \neg F \vee \neg G \tag{7}$$

**Passo 3** Posicionamento das negações imediatamente antes dos átomos

$$\neg((\forall x)F[x]) \equiv (\exists x)(\neg F[x]) \tag{8}$$
$$\neg((\exists x)F[x]) \equiv (\forall x)(\neg F[x]) \tag{9}$$

**Passo 4** Movimentação dos quantificadores com mudança da denominação de certas variáveis, se necessário.

$$(Qx)F[x] \vee G \equiv (Qx)(F[x] \vee G) \tag{10}$$
$$(Qx)F[x] \wedge G \equiv (Qx)(F[x] \wedge G) \tag{11}$$
$$(\forall x)F[x] \wedge (\forall x)G[x] \equiv (\forall x)(F[x] \wedge G[x]) \tag{12}$$
$$(\exists x)F[x] \vee (\exists x)G[x] \equiv (\exists x)(F[x] \vee G[x]) \tag{13}$$
$$(Q_1 x)F[x] \wedge (Q_2 x)G[x] \equiv (Q_1 x)(Q_2 z)(F[x] \wedge G[z]) \tag{14}$$
$$(Q_3 x)F[x] \vee (Q_4 x)G[x] \equiv (Q_3 x)(Q_4 z)(F[x] \vee G[z]) \tag{15}$$

onde $Q, Q_1, Q_2, Q_3$ e $Q_4$ identificam $\forall$ ou $\exists$.

Seguem-se dois exemplos de aplicação deste procedimento.

**Exemplo 2.7** *Transformar a fórmula* $(\forall x)P(x) \Rightarrow (\exists x)Q(x)$ *na forma normal prenex.*

*Resolução.*

$$
\begin{aligned}
(\forall x)(P(x) \Rightarrow (\exists x)Q(x) &\equiv \neg((\forall x)P(x)) \vee (\exists x)Q(x) && \text{por (4)}\\
&\equiv (\exists x)(\neg P(x)) \vee (\exists x)Q(x) && \text{por (8)}\\
&\equiv (\exists x)(\neg P(x) \vee Q(x)) && \text{por (13).}
\end{aligned}
$$

**Exemplo 2.8** *Transformar a fórmula*

$$(\forall x)(\forall y)((\exists x)(P(x,z) \wedge P(y,z)) \Rightarrow (\exists u)Q(x,y,u))$$

*na forma prenex.*

*Resolução.*
$(\forall x)(\forall y)\big((\exists z)(P(x,z) \wedge P(y,z)) \Rightarrow (\exists u)Q(x,y,u)\big)$

$$
\begin{aligned}
&\equiv (\forall x)(\forall y)\big(\neg((\exists z)(P(x,z)) \wedge P(y,z)) \vee (\exists u)Q(x,y,u)\big) && \text{por (4)}\\
&\equiv (\forall x)(\forall y)\big((\forall z)(\neg P(x,z) \vee \neg P(y,z)) \vee (\exists u)Q(x,y,u)\big) && \text{por (7) e (9)}\\
&\equiv (\forall x)(\forall y)(\forall z)(\exists u)(\neg P(x,z) \vee \neg P(y,z) \vee Q(x,y,u)) && \text{por (15)}
\end{aligned}
$$

**Exercício 2.5** *Transforme as seguintes fórmulas na forma normal prenex:*

1. $(\forall x)\big(P(x) \Rightarrow (\exists y)Q(x,y)\big)$
2. $(\exists x)\Big(\neg\big((\exists y)P(x,y)\big) \Rightarrow \big((\exists z)Q(z) \Rightarrow R(x)\big)\Big)$

# 3 Demonstração automática de Teoremas

*Quando resolvemos problemas (em geral) o que fazemos é transformar cadeias de símbolos noutras cadeias de símbolos, aplicando certas regras pré-estabelecidas.*

**Emil Post**

Embora Turing e Church [3, 8] tenham mostrado, independentemente um do outro, que não existe (não é possível encontrar) um procedimento geral para decidir (num número finito de passos) se uma fórmula da Lógica de 1ª ordem é ou não demonstrável, existem procedimentos de prova com os quais podemos constatar que uma fórmula é válida, se efectivamente ela é válida. Nas fórmulas que não são válidas, em geral, estes procedimentos não terminam.

Herbrand ([6], 1930) desenvolveu um algoritmo para determinar uma interpretação para a qual uma dada fórmula tem o valor falso. Contudo, se essa fórmula é válida (não existe uma tal interpretação) o algoritmo termina após um número limite de tentativas sem que nada se possa concluir. Este método de Herbrand constitui a base de muitos dos procedimentos de demonstração, os quais se designam usualmente por procedimentos de refutação.

Gilmore ([4], 1960) foi um dos pioneiros na implementação do algoritmo de Herbrand em computador. Uma vez que uma fórmula é válida sse a sua negação é inconsistente, o programa desenvolvido por Gilmore tenta detectar a inconsistência da negação da fórmula em estudo.

Muitas fórmulas da Lógica de 1ª ordem, intuitivamente válidas, permanecem por provar em computador num tempo razoável.

Convém, nesta altura, notar que, embora os conceitos de fórmula demonstrável, ou seja, de teorema e fórmula válida sejam diferentes (dado que a primeira se constata à custa de procedimentos puramente sintáticos enquanto a segunda se conclui à custa de procedimentos semânticos, isto é, de interpretações), o facto de Godel (resp., Post) ter provado em 1930 (1921) que uma fórmula da Lógica de 1ª ordem (resp., Cálculo Proposicional) é um teorema sse é válida, permite-nos lidar indiferentemente com qualquer destes conceitos.

## 3.1 Eliminação de Quantificadores Existenciais

Os procedimentos de refutação aplicados a fórmulas da Lógica de 1ª ordem, assentam nos seguintes procedimentos:

1. Qualquer fórmula da Lógica de 1ª ordem se pode transformar na forma normal prenex.
2. A parte da fórmula que não contém quantificadores pode-se transformar na forma normal conjuntiva.
3. É possível eliminar os quantificadores existenciais sem se alterar as propriedades de inconsistência.

Para se conseguir a eliminação dos quantificadores existênciais , em geral, utilizam-se as designadas funções de Skolem, conforme a seguir se indica.

Seja $F$ uma fórmula na forma normal prenex

$$(Q_1 x_1) \cdots (Q_n x_n) M,$$

onde $M$ está na forma normal conjuntiva.

Suponhamos que $Q_r$ é um quantificador existencial. Se nenhum quantificador universal aparece à esquerda de $Q_r$, escolhemos uma nova constante, c, diferente de qualquer das outras que ocorrem em $M$, substituimos $x_r$ por $c$ e eliminamos $Q_r x_r$. Se $Q_{s_1}, Q_{s_2}, \cdots, Q_{s_m}$ são todos quantificadores universais que ocorrem à esquerda de $Q_r$ $(1 \leq s_1 < s_2 \ldots < s_m \leq n)$ escolhemos um novo símbolo de função, $f$, com $m$ argumentos, diferente dos já existentes e substituímos $x_r$ em $M$ por $f(x_{s_1}, x_{s_2}, \ldots, x_{s_m})$ e eliminamos $(Q_r x_r)$.

O procedimento anteriormente descrito deverá ser aplicado a todos os quantificadores existênciais de modo a obter-se a fórmula no que se designa por "Skolem standard form", ou simplesmente forma standard.

**Exemplo 3.1** *Aplicando-se à fórmula* $(\exists x)(\forall y)(\forall z)(\exists u)(\forall v)(\exists w)P(x,y,z,u,v,w)$ *o conjunto de regras anteriormente descritas obtém-se:*

$$(\forall y)(\forall z)(\forall v)P(a,y,z,f(y,z),v,g(y,z,v)).$$

*Por sua vez a redução da fórmula*

$$(\forall x)(\exists y)(\exists z)\Big(\big(\neg P(x,y)\wedge Q(x,y)\big)\vee R(x,y,z)\Big)$$

*à "Skolem standard form" dá, numa fase intermédia,*

$$(\forall x)(\exists y)(\exists z)\Big(\big(\neg P(x,y)\vee R(x,y,z)\big)\wedge(Q(x,y)\vee R(x,y,z)\Big)$$

*e posteriormente*

$$(\forall x)((\neg P(x,f(x))\vee R(x,f(x),g(x))\wedge(Q(x,f(x))\vee R(x,f(x),g(x))).$$

**Exercício 3.1** *Encontre a forma standard das seguintes fórmulas:*

1. $\neg\big((\forall x)P(x)\ \Rightarrow\ (\exists y)(\forall z)Q(y,z)\big)$

2. $\neg((\forall x)P(x)\ \Rightarrow\ (\exists y)P(y))$

**Definição 3.1** *Uma cláusula é uma disjunção finita de (uma ou mais) literais. Uma $r$-cláusula é um cláusula de $r$ literais, designando-se por cláusula vazia uma cláusula sem literais.*

**Exemplo 3.2** *As disjunções*

$$\neg P(x,f(x))\vee R\big(x,f(x),g(x)\big)\ \text{ e } Q(x,g(x))\vee R\big(x,f(x),g(x)\big)$$

*são exemplos de cláusulas.*

Um conjunto $S$ de cláusulas é visto como a conjunção de todas as cláusulas de $S$, onde qualquer variável é considerada como sendo governada por quantificadores universais . Como exemplo podemos considerar o conjunto

$$S = \{\neg P(x, f(x)) \vee R\big(x, f(x), g(x)\big), Q(x, g(x)\big) \vee R\big(x, f(x), g(x)\big)\}$$

que corresponde à fórmula

$$\Big(\neg P(x, f(x)) \vee R\big(x, f(x), g(x)\big) \wedge \big(Q(x, g(x)) \vee R\big(x, f(x), g(x)\big)\big)\Big).$$

Qualquer fórmula $F$ pode ser representada pelo conjunto de cláusulas que constituem a forma standard de $F$.

**Teorema 3.1** *Seja $S$ um conjunto de cláusulas que representam uma fórmula $F$ na forma standard. Então $F$ é inconsistente sse $S$ é inconsistente.*

Neste teorema entenda-se inconsistência de $S$ como sendo a inconsistência da fórmula que $S$ representa.

Dada a extensão da prova deste teorema e o facto de ir além dos objectivos deste texto não a abordaremos, no entanto, os mais curiosos (e/ou saudavelmente desconfiados) poderão encontrá-la em [2, pág. 48].

## 3.2 Princípio da Resolução de Robinson

Robinson ([7], 1965) introduziu o princípio de resolução, a qual foi claramente a mais eficiente das regras, até então adoptadas em computação simbólica, para a determinação da inconsistência.

A ideia essencial do princípio da resolução consiste em verificar se um dado conjunto de cláusulas $S$ contém a cláusula vazia $\diamondsuit$ ou se ela pode ser deduzida de $S$.

O princípio da resolução pode ser visto como uma regra de inferência que deve ser usada para gerar novas cláusulas a partir de um conjunto $S$ de acordo com o procedimento que a seguir se descreve.

Para quaisquer duas cláusulas $C_1$ e $C_2$, se existe um literal $L_1$ em $C_1$ que é complementar relativamente a um literal $L_2$ em $C_2$, então deve eliminar-se $L_1$ e $L_2$ de $C_1$ e $C_2$, respectivamente, e construir-se a disjunção do que resta das cláusulas. A cláusula assim construida diz-se uma resolvente de $C_1$ e $C_2$.

**Exemplo 3.3**

$$\begin{array}{l} C_1 : P \vee R \\ C_2 : \neg P \vee Q \\ \hline \qquad\qquad C_{12} : R \vee Q \end{array}$$

$$\begin{array}{l} C_1 : \neg P \vee Q \vee R \\ C_2 : \neg Q \vee S \\ \hline \qquad\qquad C_{12} : \neg P \vee R \vee S \end{array}$$

*Para o conjunto formado pelas cláusulas $C_1 :\ \neg P \vee Q$ e $C_2 :\ \neg P \vee R$, não existe qualquer resolvente.*

Uma propriedade importante associada às resolventes é a de que qualquer resolvente de duas cláusulas $C_1$ e $C_2$ é consequência lógica de $C_1$ e $C_2$, conforme se estabelece no teorema a seguir.

**Teorema 3.2** *Dadas duas cláusulas $C_1$ e $C_2$, uma resolvente, $C$, de $C_1$ e $C_2$ é uma consequência lógica de $C_1$ e $C_2$.*

*Demonstração.*

Sejam as cláusulas $C_1$, $C_2$ e $C$ tais que $C_1 \,:\, L \vee C_{1'}$, $C_2 \,:\, \neg L \vee C_{2'}$ e $C \,:\, C_{1'} \vee C_{2'}$, onde $L$ é um literal e $C_{1'}$ e $C_{2'}$ são disjunções literais.

Suponha-se que $C_1$ e $C_2$ são verdadeiras para uma interpretação $I$. Pretendemos provar que a resolvente de $C_1$ e $C_2$ é também verdadeira para $I$. Note-se que para $I$, ou $L$ é falso ou $\neg L$ é falso. Seja $L$ falso para $I$, então $C_{1'}$ é verdadeiro para $I$ e como $C \equiv C_{1'} \vee C_{2'}$, tem-se que $C$ é verdadeiro para $I$. De um modo análogo se prova que se $\neg L$ é falso para $I$, então $C_{2'}$ é verdadeiro para $I$ e $C$ é verdadeiro para $I$. $\square$

Se temos duas cláusulas unitárias que admitem uma resolvente, então essa resolvente é a cláusula vazia $\diamondsuit$. Mais importante ainda é que se um conjunto $S$ de cláusulas é inconsistente, então poderemos aplicar o princípio da resolução para gerar a cláusula $\diamondsuit$ a partir de $S$.

**Definição 3.2** *Dado um conjunto de cláusulas, $S$, uma dedução (resolução) de $C$ obtida a partir de $S$ é uma sequência finita de cláusulas $C_1, C_2, \cdots C_k$ tais que cada $C_i$ ou é uma cláusula em $S$ ou uma resolvente de cláusulas que precedem $C_i$ e $C_k = C$. A dedução de $\diamondsuit$ a partir de $S$ é designada por refutação ou prova da inconsistência de $S$.*

*Dizemos que uma cláusula $C$ pode ser deduzida ou derivada de $S$ se ela é uma dedução obtida a partir de $S$.*

Note-se que, sendo a cláusula vazia gerada a partir de duas cláusulas unitárias, isso significa que o conjunto de clausulas que estamos a testar, ou contém as cláusulas $C$ e $\neg C$, ou estas podem ser geradas a partir desse conjunto. Em qualquer dos casos, o conjunto será um conjunto inconsistente.

**Exemplo 3.4 (Aplicação do Princípio da Resolução)** *Considerando o conjunto $S$ de cláusulas:*

$$(1)\, \neg P \vee Q \qquad (2)\, \neg Q \qquad (3)\, P$$

*obtém-se:*

*de (1) e (2) vem (4)* $\neg P$

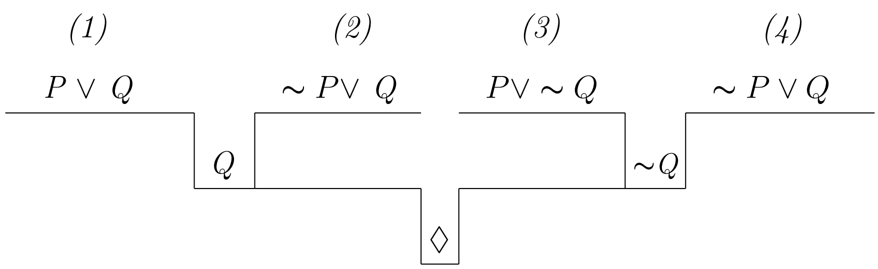
*de (4) e (3) vem $\diamondsuit$.*

*Logo pode concluir-se que $S$ é inconsistente.*

**Exemplo 3.5 (Aplicação do Princípio da Resolução)** *Considerando o conjunto $S$ de cláusulas:*

$$(1)\, P \vee Q \quad (2)\, \neg P \vee Q \quad (3)\, P \vee \neg Q \quad \neg P \vee \neg Q$$

*obtém-se*




*Logo $S$ é inconsistente.*

**Exercício 3.2** *Mostre a inconsistência do seguinte conjunto de cláusulas:*

$$S = \{P \vee Q, \neg Q \vee R, \neg P \vee Q, \neg R\}.$$

### 3.2.1 Substituição e Unificação das Fórmulas da Lógica de 1ª ordem

Embora o princípio da resolução seja uma poderosa regra de inferência, uma vez que o pretendemos aplicar à Lógica de 1ª ordem, teremos que efectuar certas modificações nas respectivas fórmulas. Para o fazermos será necessário definir certos conceitos, entre os quais, talvez o mais importante seja o de substituição. Para facilitar esta tarefa, vamos introduzir a seguinte notação:

(i) $VAR = \{v : v$ é uma variável individual$\}$, ou seja, $VAR$ identificará o conjunto das variáveis individuais.

(ii) $CONST = \{c : c \text{ é o símbolo de uma constante}\}$.

(iii) $TERM = \{t : t \text{ é um termo}\}$, isto é, $TERM$ identificará o conjunto dos termos da Lógica de 1ª ordem.

**Nota 3.1** *$TERM \supset CONST \cup VAR$ de acordo com a Definição 2.8.*

**Definição 3.3** *Uma substituição é uma função $\varphi_V : VAR \longrightarrow TERM$ tal que, sendo $U_\varphi = \{v \in VAR : \varphi_V(v) \neq v\}$ e supondo que $U_\varphi = \{v_1, v_2, \cdots, v_n\}$, podemos descrever através do conjunto $\{t_1/v_1, t_2/v_2, \cdots, t_n/v_n\}$, com $t_i = \varphi_V(v_i) \neq v_i$, para $i = 1, 2, \cdots, n$. Este modo de descrever a função $\varphi_V$ leva-nos (com algum abuso de linguagem) a escrever $\varphi_V = \{t_1/v_1, t_2/v_2, \cdots, t_n/v_n\}$, para indicar que :*

*(i) se $v_i \in U_\varphi$, então $\varphi_V(v_i) = t_i$;*

*(ii) se $v_i \notin U_\varphi$ então $\varphi_V(v) = v$.*

*A substituição identidade representa-se por $\varepsilon_V$ e, tendo em conta que $\forall v \in VAR$ $\varepsilon_V(v) = v$, de acordo com a notação anterior, escreve-se $\varepsilon_V = \emptyset$ já que $U_\varepsilon = \emptyset$. Por esta razão também se designa a substituição identidade por substituição vazia.*
*Ao longo deste texto utilizaremos letras gregas para representar substituições.*

São exemplos de substituições,

$$\Theta_V = \{f(z)/x, y/z\} \quad \text{e } \delta_V = \{a/x, g(y)/y, f(g(z))/z\}.$$

**Definição 3.4** *Sendo $\Theta_V = \{\Theta_V(v_1)/v_1, \Theta_V(v_2)/v_2, \cdots, \Theta_V(v_n)/v_n\}$ uma substituição, $\Theta_V$ induz uma função $\Theta_T : TERM \longrightarrow TERM$, definida, recursivamente, do seguinte modo:*

*(i) se $t_i \in VAR$, então $\Theta_T(t_i) = \Theta_V(t_i)$;*

*(ii) se $t_i \in CONST$, então $\Theta_T(t_i) = t_i$;*

*(iii)* *se* $t_i \notin VAR \cup CONST$, *ou seja,* $t_i$ *é um termo da forma* $f(t_{i_1}, t_{i_2}, \cdots, t_{i_k})$ *onde* $f$ *é um símbolo de função com* $k$ *argumentos, então*

$$\Theta_T(f(t_{i_1}, t_{i_2}, \cdots, t_{i_k})) = f(\Theta_T(t_{i_1}), \Theta_T(t_{i_2}) \cdots, \Theta_T(t_{i_k})).$$

**Exemplo 3.6** *Seja* $t = s(x, f(y,u), h(x,z))$ *e*

$$\varphi_V = \{f(x,z)/x, g(y, f(x,y))/y, h(x,y)/z, v/u\},$$

*então obtém-se:*

$$\begin{aligned}
\varphi_T(t) &= \varphi_T(s(x, f(y,u), h(x,z))) \\
&= s(\varphi_T(x), \varphi_T(f(y,u), \varphi_T(h(x,z)))) \\
&= s(\varphi_V(x), f(\varphi_V(y), \varphi_V(u)), h(\varphi_V(x), \varphi_V(z))) \\
&= s(f(x,z), f(g(y, f(x,y)), v), h(f(x,z), h(x,y))).
\end{aligned}$$

Embora com abuso de linguagem, sempre que o contexto não der lugar a confusões, referir-nos-emos a uma substituição $\varphi_V$ e/ou à correspondente função induzida $\varphi_T$ utilizando, unica e indistintamente, a respectiva letra grega que a denota, neste caso $\varphi$.

**Definição 3.5** *Seja* $\Theta = \{\Theta_V(v_1)/v_1, \cdots, \Theta_V(v_n)/v_n\}$ *uma substituição e* $E$ *uma expressão, então* $E\Theta$ *é a expressão obtida de* $E$ *aplicando a cada termo de* $E$ *a função induzida pela substituição* $\Theta$*, ou seja, substituindo, simultaneamente, cada ocorrência da variável* $v_i$ $(1 \leq i \leq n)$ *em* $E$ *pelo termo* $t_i = \Theta_V(v_i)$*. Diz-se que* $E\Theta$ *é uma concretização (exemplo) de* $E$*. Supondo que* $W$ *denota o conjunto de expressões* $\{E_1, \cdots, E_p\}$*,* $W\Theta$ *denota o conjunto* $\{E_1\Theta, \cdots, E_p\Theta\}$*.*

**Exemplo 3.7** *Considerando-se a substituição* $\Theta = \{a/x, f(b)/y, c/z\}$ *e a expressão* $E = F(x, y, g(z))$*, vem que*

$$\begin{aligned}
E\Theta = F(\Theta_T(x), \Theta_T(y), \Theta_T(g(z))) &= F(\Theta_V(x), \Theta_V(y), g(\Theta_V(z))) \\
&= F(a, f(b), g(c)).
\end{aligned}$$

**Exercício 3.3** *Seja* $\Theta = \{a/x, f(z)/y, g(x)/z\}$, *E=P(h(x),z,f(z)). Calcule* $E\Theta$.

**Definição 3.6** *Sejam* $\Theta_V$ *e* $\lambda_V$ *duas substituições, então a composição de* $\Theta_V$ *e* $\lambda_V$ *que se denota por* $\Theta_V \Delta \lambda_V$*, define-se como sendo* $\Theta_V \Delta \lambda_V = \Theta_T o \lambda_T$ *onde o símbolo o identifica a composição habitual de funções.*

De acordo com esta definição, dadas as substituições $\Theta_V$ e $\lambda_V$, a sua composição $\Theta_V \Delta \lambda_V$ poder-se-ia descrever esquematicamente pelo diagrama a seguir.

$$\begin{tikzcd} \mathrm{VAR} \arrow[r, "\lambda_V"] \arrow[dr, "\theta_V \Delta \lambda_V = \theta_T o \lambda_V"'] & \mathrm{TERM} \arrow[d, "\theta_T"] & \mathrm{VAR} \arrow[l, "\theta_V"'] \\ & \mathrm{TERM} & \end{tikzcd}$$

Da Definição 3.6 decorre, ainda, um método prático e muito simples, para a determinação da composição de duas substituições.

Se $\Theta = \{\Theta_V(y_1)/y_1, \cdots, \Theta_V(y_p)/y_p\}$ e $\lambda = \{\lambda_V(z_1)/z_1, \cdots, \lambda_V(z_p)/z_p\}$ são duas substituições, então, supondo que $\{y_1, \cdots, y_p\} \cup \{z_1, \cdots, z_q\} = \{x_1, \cdots, x_n\}$, a composição de $\Theta$ com $\lambda$ corresponde à substituição:

$$\Theta \Delta \lambda = \Theta_T o \lambda_V = \{\Theta_T\big(\lambda_V(x_1)\big)/x_1, \Theta_T(\lambda_V(x_2))/x_2, \cdots, \Theta_T(\lambda_V(x_n))/x_n\},$$

onde para cada $i \in \{1, 2, \cdots, n\}$, se $x_i \notin \{z_1, \cdots, z_q\}$ então $\lambda_V(x_i) = x_i$.

**Exemplo 3.8** *Sejam as substituições* $\Theta = \{f(y)/x, z/y\}$ *e* $\lambda = \{a/x, g(x)/y, y/z\}$.

*Então*

$$
\begin{aligned}
\Theta\Delta\lambda &= \Theta_T o \lambda_V = \{\Theta_T(\lambda_V(x))/x, \Theta_T(\lambda_V(y))/y, \Theta_T(\lambda_V(z))/z\} \\
&= \{\Theta_T(a)/x, \Theta_T(g(x))/y, \Theta_T(y)/z\} = \{a/x, g(\Theta_V(x))/y, \Theta_V(y)/z\} \\
&= \{a/x, g(f(y))/y, z/z\} \\
&= \{a/x, g(f(y))/y\}
\end{aligned}
$$

**Lema 3.1** *Dadas as substituições*

$$\Theta = \{\Theta_V(y_1)/y_1, \cdots, \Theta_V(y_p)/y_p\} \text{ e } \lambda = \{\lambda_V(z_1)/z_1, \cdots, \lambda_V(z_p)/z_p\}$$

*tem-se que se* $\tau_V = \Theta\Delta\lambda$ *(ou seja,* $\tau_V = \Theta_T o \lambda_V$*) então* $\tau_T = \Theta_T o \lambda_V$*.*

*Demonstração.*
Considere-se a substituição $\tau_V = \Theta_T o \lambda_V$ e um termo arbitrário $t$.

a) Se $t$ é uma constante $c$ então, de acordo com a definição de função induzida por uma substituição, vem $\tau_T(c) = c$ e $\Theta_T o \lambda_T(c) = \Theta_T(\lambda_T(c)) = \Theta_T(c) = c$.

b) Se $t$ é uma variável $x$ então vem que $\tau_T(x) = \tau_V(x) = \Theta_T o \lambda_V(x) = \Theta_T(\lambda_V(x))$ e $\Theta_T o \lambda_T(x) = \Theta_T(\lambda_T(x)) = \Theta_T(\lambda_V(x))$ (note-se que sendo $t$ uma variável $\lambda_T(x) = \lambda_V(x)$).

c) Suponha que $t = f(t_1, t_2, \cdots, t_n)$. Então $\tau_T(t) = f(\tau_T(t_1), \tau_T(t_2), \cdots, \tau_T(t_n))$. Nestas condições, $\forall i \in \{1, 2, \cdots, n\}$ se $t_i$ é uma constante ou uma variável, então, por a) e b), $\tau_T(t_i) = \Theta_T o \lambda_T(t_i)$, caso contrário, $t_i$ volta a ser da forma $t_i = f_i(t_{i_1}, t_{i_2}, \cdots, t_{i_k})$ e o processo repete-se (isto é, $\tau_T(t_i) = f_i(t_{i_1}, t_{i_2}, \cdots, t_{i_k})$) até que se obtenham constantes e/ou variáveis. Em qualquer dos casos vem que $\tau_T(t) = \Theta_T o \lambda_T(t)$.

$\square$

**Teorema 3.3** *Sendo* $S$ *o conjunto das substituições,* $< S, \Delta >$ *é um monóide, no qual o elemento neutro é a substituição identidade (também designada por vazia).*

*Demonstração.*

É claro que o conjunto das substituições é fechado relativamente à operação $\Delta$.

Para provar que a operação $\Delta$ é associativa, considerem-se três substituições arbitrárias $\varphi, \psi$ e $\zeta$; Então vem que

$$\begin{aligned}
(\varphi_V \Delta \psi_V)\Delta \zeta_V &= (\varphi_T o \psi_V)\Delta \zeta_V && \text{(dado que } \varphi_V \Delta \psi_V = \varphi_T \Delta \psi_V\text{, por definição)}\\
&= \tau_V \Delta \zeta_V && \text{(fazendo } \tau_V = \varphi_T o \psi_V\text{)}\\
&= \tau_T \Delta \zeta_V && \text{(dado que } \tau_V \Delta \zeta_V = \tau_T o \zeta_V\text{, por definição)}\\
&= (\varphi_T o \psi_T) o \zeta_V && \text{(dado que } \tau_T = \varphi_T o \psi_T\text{, pelo Lema 3.1)}\\
&= \varphi_T o(\psi_T o \zeta_V) && \text{(associatividade de } o\text{, e composição de apl.)}\\
&= \varphi_T o(\psi_V \Delta \zeta_V) && \text{(dado que } \psi_V \Delta \zeta_V = \psi_T o \zeta_V\text{, por definição)}\\
&= \varphi_T o \lambda_V && \text{(fazendo } \lambda_V = \psi_V \Delta \zeta_V\text{)}\\
&= \varphi_V \Delta \lambda_V && \text{(dado que } \varphi_T o \lambda_V = \varphi_V \Delta \lambda_V\text{, por definição)}\\
&= \varphi_V \Delta(\psi_V \Delta \zeta_V) && \text{(dado que } \lambda_V = \psi_V \Delta \zeta_V\text{)}
\end{aligned}$$

Resta provar que a substituição identidade (vazia) é o elemento neutro à esquerda e à direita relativamente a esta operação.

Seja $\Theta$ uma substituição arbitrária, $\varepsilon$ a substituição identidade e $t$ um termo arbitrário. Seja $\tau = \Theta_V \Delta \varepsilon_V = \Theta_T o \varepsilon_V$. Então vem que $\tau_T(t) = t$ se $t$ é uma constante ou uma variável e $\tau_T(t) = f(\tau_T(t_1), \tau_T(t_2), \cdots, \tau_T(t_n))$ se $t = f(t_1, t_2, \cdots, t_n)$. No primeiro caso, vem que $\tau_T(t) = (\Theta_T o \varepsilon_V)(t) = \Theta_T(\varepsilon_V(t)) = \Theta_T(t)$.

No segundo, vem que $\forall i \in \{1, 2, \cdots, n\}$ se $t_i$ é uma constante ou uma variável, então tal como anteriormente, vem que $\tau_T(t_i) = \Theta_T(t_i)$. Caso contrário, de um modo recursivo o processo repete-se até que se obtém $\tau_T(t) = \Theta_T(t)$.

De um modo idêntico se prova que $\varepsilon$ é o elemento neutro à esquerda relativamente à composição de substituições. □

**Definição 3.7** *Um substituição $\Theta$ designa-se por unificadora para o conjunto de expressões $W = \{E_1, E_2, \cdots, E_p\}$ sse $W\Theta = \{E\Theta\}$, tal que $E\Theta = E_1\Theta = \cdots = E_p\Theta$. O conjunto $W$ diz-se unificável se existe um unificador para ele.*

**Exemplo 3.9** *O conjunto das expressões* $\{P(a,y), P(x, f(b))\}$ *é unificável, uma vez que a substituição* $\Theta = \{a/x, f(b)/y\}$ *é um unificador para este conjunto.*

**Definição 3.8** *Um unificador* $\sigma$ *para um conjunto de expressões* $W = \{E_1, E_2, \cdots, E_p\}$ *é designado como unificador mais geral sse qualquer que seja o unificador* $\Theta$ *para o conjunto de expressões* $W$ *existe uma substituição* $\lambda$ *tal que* $\Theta = \sigma\Delta\lambda$.

A ideia base do algoritmo de unificação consiste em, dadas duas expressões, detectar se são ou não idênticas e, no caso de não serem, determinar aquilo em que diferem para posteriormente se tentar a unificação.

Considerem-se as expressões não idênticas $P(a)$ e $P(x)$. Facilmente se reconhece que elas diferem no facto de que, enquanto $\underline{a}$ ocorre na primeira expressão, $\underline{x}$ ocorre na segunda. De modo a proceder-se à respectiva unificação de $P(a)$ e $P(x)$, teremos de encontrar primeiro as diferenças. Para $P(a)$ e $P(x)$, o conjunto de diferenças é $\{a, x\}$. Uma vez que $\underline{x}$ é uma variável, $\underline{x}$ pode ser substituido por $\underline{a}$, e consequentemente as diferenças acabam.

**Definição 3.9** *O conjunto de diferenças de um conjunto de expressões não vazio,* $W$*, obtém-se determinando o primeiro símbolo (a contar da esquerda), no qual nem todas as expressões em* $W$ *têm exactamente os mesmos símbolos, e então extrair para cada expressão em* $W$ *a subexpressão que começa com o símbolo em causa e ocupa essa posição. O conjunto dessas subexpressões corresponde ao conjunto das diferenças de* $W$.

Por exemplo considere-se $W = \{P(x, \underline{f(y,z)}), P(x, \underline{a}), P(x, \underline{g(h, k(x))})\}$, então o primeiro símbolo a partir do qual nem todos os átomos em $W$ são os mesmos é o 5º. Logo, vem que o conjunto de diferenças corresponde a $D = \{f(y,z), a, g(h, (k(x)))\}$.

### 3.2.2 Algoritmo de Unificação

I. Fazer $k = 0, W_k = W$ e $\sigma_k = \varepsilon$;

II. Se $W_k$ é um conjunto unitário, então `stop` ( $\sigma_k$ é o unificador mais geral para $W$). Caso contrário, determinar o conjunto de diferenças de $W_k$, $D_k$;

III. Se existem elementos $v_k$ e $t_k$ em $D_k$, tais que $v_k$ é uma variável que não ocorre em $t_k$, então saltar para IV. Caso contrário, `stop` ($W$ não é unificável );

IV. Fazer $\sigma_{k+1} = \sigma_k\{t_k/v_k\}$ e $W_{k+1} = W_k\{t_k/v_k\}$ ( note-se que $W_{k+1} = W_{\sigma_{k+1}}$);

V. Fazer $k = k + 1$ e voltar a II.

**Exemplo 3.10** *Determinar o unificador mais geral de*

$$W = \{P(a, x, f(g(y))), P(z, f(z), f(u))\}.$$

*Resolução.*

1. Seja $\sigma_0 = \varepsilon$ e $W_0 = W$.

2. Uma vez que $W_0$ não é um conjunto unitário, $\sigma_0$ não é o unificador mais geral para $W$, pelo que se deve determinar $D_0 = \{a, z\}$.

3. Seja $\sigma_1 = \{\sigma(v_0)/v_0\}o\varepsilon = \{a/z\}o\varepsilon = \{a/z\}$,

$$\begin{aligned} W_1 = W_0\{t_0/v_0\} &= \{P(a, x, f(g(y))), P(z, f(z), f(u))\}\{a/z\} \\ &= \{P(a, x, f(g(y))), P(a, f(a), f(u))\}. \end{aligned}$$

4. Uma vez que $W_1$ não é um conjunto unitário, determina-se o conjunto de diferenças de $W_1$, $D_1 = \{x, f(a)\}$

5. A partir de $D_1$ encontra-se que $v_1 = x$ e $t_1 = f(a)$.

6. Seja

$$\begin{aligned}
\sigma_2 = \{\sigma(v_1)/v_1\} o \sigma_1 &= \{f(a)/x\} o \{a/z\} \\
&= \{\sigma(\sigma_1(x))/x, \sigma(\sigma_1(z))/z\} \\
&= \{\sigma(x)/x, \sigma(a)/z\} = \{f(a)/x, a/z\} \\
W_2 = W_1\{t_1/v_1\} &= \{P(a, x, f(g(y))), P(a, f(a), f(u))\}\{f(a)/x\} \\
&= \{P(a, f(a), f(g(y))), P(a, f(a), f(u))\}
\end{aligned}$$

7. Mais uma vez $W_2$ não é um conjunto unitário, pelo que se deve determinar o conjunto de diferenças de $W_2$, $D_2 = \{g(y), u\}$.

8. A partir de $D_2$ conclui-se que $v_2 = u$ e $t_2 = g(y)$.

9. Seja

$$\begin{aligned}
\sigma_3 = \{\sigma(v_2)/v_2\} o \sigma_2 &= \{g(y)/u\} o \{a/z, f(a)/x\} \\
&= \{\sigma(\sigma_2(x))/x, \sigma(\sigma_2(z))/z, \sigma(\sigma_2(u))/u\} \\
&= \{\sigma(f(a))/x, \sigma(a)/z, \sigma(u)/u\} \\
&= \{f(a)/x, a/z, g(y)/u\}, \\
W_3 = W_2\{t_2/v_2\} &= \{P(a, f(a), f(g(y))), P(a, f(a), f(u))\}\{g(y)/u\} \\
&= \{P(a, f(a), f(g(y))), P(a, f(a), f(g(y)))\} \\
&= \{P(a, f(a), f(g(y)))\}.
\end{aligned}$$

10. Uma vez que $W_3$ é um conjunto unitário, $\sigma_3 = \{f(a)/x, a/z, g(y)/u\}$ é o unificador mais geral para $W$.

**Exemplo 3.11** *Verificar se o conjunto* $W = \{Q(f(a), g(x)), Q(y, y)\}$ *é ou não unificável.*

*Resolução.*

1. Seja $\sigma_0 = \varepsilon$ e $W_0 = W$.

2. Uma vez que $W_0$ não é um conjunto unitário devemos determinar o conjunto de diferenças $D_0 = \{f(a), y\}$.

3. A partir de $D_0$, tem-se que $v_0 = y$ e $t_0 = f(a)$

4. Seja

$$\begin{aligned}
\sigma_1 = \{\sigma(v_0)/v_0\} o \sigma_0 &= \{f(a)/y\} o \varepsilon \\
&= \{f(a)/y\} \\
W_1 = W_0\{t_0/v_0\} &= \{Q(f(a), g(x)), Q(y, y)\}\{f(a)/y\} \\
&= \{Q(f(a), g(x)), Q(f(a), f(a))\}.
\end{aligned}$$

5. Uma vez que $W_1$ não é um conjunto unitário determina-se o conjunto de diferenças $D_1 = \{g(x), f(a)\}$.

6. Contudo nenhum dos elementos de $D_1$ é uma variável. Consequentemente o algoritmo de unificação termina concluindo-se que $W$ não é unificável.

**Exercício 3.4** *Averigue quais dos seguintes conjuntos são unificáveis. Se a resposta for afirmativa, obtenha o unificador mais geral.*

*1.* $W = \{Q(a, x, f(x)), Q(a, y, y)\}$

*2.* $W = \{Q(x, y, z), Q(u, h(v, v), u)\}$

Devemos observar que o algoritmo de unificação anteriormente descrito terminará sempre, para um número finito e não vazio de expressões. Caso contrário, gerar-se-ia uma sequência infinita $W_{\sigma_0}, W_{\sigma_1}, \cdots$, de conjuntos finitos não vazios de expressões com a propriedade de que cada conjunto da sequência contém menos uma variável do que o seu predecessor (nomeadamente, $W_{\sigma_k}$ contém a variável $v_{k+1}$ mas $W_{\sigma_{k+1}}$ não a contém). Porém, isto não é possível, uma vez que $W$ contém apenas um número finito de variáveis.

Prova-se ainda que (ver [2], pág. 79) se $W$ é um conjunto de expressões não vazio, finito e unificável, então o algoritmo de unificação, anteriormente descrito, determina um unificador mais geral para $W$.

### 3.2.3 Aplicação do Princípio da Resolução à Lógica de 1ª ordem

Uma vez introduzido o algoritmo de unificação, é agora possível aplicar o princípio de resolução à Lógica de 1ª ordem.

**Definição 3.10** *Se dois ou mais literais (com o mesmo sinal) de uma cláusula $C$ têm o mesmo unificador mais geral $\sigma$, então $C\sigma$ designa-se por factor de $C$.*
*Se $C\sigma$ é uma cláusula unitária, então ela é um factor unitário de $C$.*

**Exemplo 3.12** *Sendo $C : \underline{P(x)} \vee \underline{P(f(y))} \vee \neg Q(x)$, o primeiro e o segundo literal (sublinhados) têm o unificador mais geral $\sigma = \{f(y)/x\}$.*
*Então $C\sigma = P(f(y)) \vee \neg Q(f(y))$ é um factor de $C$.*

**Exercício 3.5** *Averigue se as seguintes cláusulas admitem um factor. Em caso afirmativo, determine-o.*

1. $P(x) \vee P(a) \vee (Qf(x)) \vee Q(f(a))$

2. $P(x) \vee P(f(y)) \vee Q(x, y)$

**Definição 3.11** *Sejam $C_1$ e $C_2$ duas cláusulas com nenhuma variável em comum. Sejam $L_1$ e $L_2$ dois literais em $C_1$ e $C_2$, respectivamente. Se $L_1$ e $\neg L_2$ têm o unificador mais geral $\sigma$, então a cláusula $(C_1\sigma - L_1\sigma) \cup (C_2\sigma - L_2\sigma)$ designa-se por resolvente binária de $C_1$ e $C_2$.*

**Exemplo 3.13** *Considerem-se as cláusulas $C_1 = P(x) \vee Q(x)$ e $C_2 = \neg P(a) \vee R(x)$. Uma vez que $x$ aparece tanto em $C_1$ como em $C_2$, nesta última cláusula vamos mudar $x$ para $y$, obtendo-se $C_2 = \neg P(a) \vee R(y)$.*

*Escolha-se $L_1 = P(x)$ e $L_2 = \neg P(a)$. Dado que $\neg L_2 \equiv P(a)$, $L_1$ e $\neg L_2$ têm o unificdor mais geral $\sigma = \{a/x\}$. Nestas condições,*

$$\begin{aligned}(C_1\sigma - L_1\sigma) \cup (C_2\sigma - L_2\sigma) &= (\{P(a), Q(a)\} - \{P(a)\}) \cup (\{\neg P(a), R(a)\} - \{\neg P(a)\}) \\ &= \{Q(a)\} \cup \{R(y)\} = \{Q(a), R(y)\} = Q(a) \vee R(y).\end{aligned}$$

*Consequentemente $Q(a) \vee R(y)$ é uma resolvente binária de $C_1$ e $C_2$, sendo $P(x)$ e $\neg P(a)$ os literais resolvidos.*

**Definição 3.12** *Uma resolvente das cláusulas de $C_1$ e $C_2$ é uma das seguintes resolventes:*

1. *uma resolvente binária de $C_1$ e $C_2$,*
2. *uma resolvente binária de $C_1$ e $C_2\sigma$,*
3. *uma resolvente binária de $C_1\sigma$ e $C_2$,*
4. *uma resolvente binária de $C_1\sigma$ e de $C_2\sigma$.*

**Exemplo 3.14** *Seja $C_1 = P(x) \vee P(f(y)) \vee R(g(y))$ e $C_2 = \neg P(f(g(a))) \vee Q(b)$. Um factor de $C_1$ é $C_{1'} = P(f(y)) \vee R(g(y))$. Uma resolvente binária de $C_{1'}$ e $C_2$ é $R(g(g(a))) \vee Q(b)$. Logo $R(g(g(a))) \vee Q(b)$ é uma resolvente de $C_1$ e $C_2$.*

**Exemplo 3.15** *Mostrar que os ângulos internos alternados formados pela diagonal de um trapézio são iguais.*

*Resolução.*

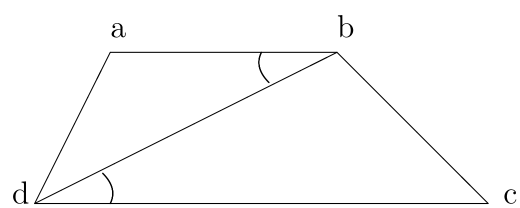
Para provar este teorema, vamos primeiramente axiomatizá-lo convenientemente.

Seja $T(x, y, u, v)$ um trapézio cujo vértice superior esquerdo é $x$, o superior direito é $y$, o inferior direito é $u$ e o inferior esquerdo é $v$; significando por $P(x, y, u, v)$ que a linha que une o segmento $xy$ é paralela à linha que une o segmento $uv$; e por $E(x, y, z, u, v, w)$ que o ângulo $xyz$ é igual ao ângulo $uvw$, então temos os seguintes axiomas:

$A_1$ $(\forall x)(\forall y)(\forall u)(\forall v)(T(x, y, u, v) \Rightarrow P(x, y, u, v))$ (definição de trapézio).

$A_2$ $(\forall x)(\forall y)(\forall u)(\forall v)(P(x, y, u, v) \Rightarrow E(x, y, v, u, v, y))$.

$A_3$ $T(a, b, c, d)$.




A partir destes axiomas deveremos estar em condições de concluir que $E(a, b, d, c, d, b)$ é verdadeiro, ou seja, $A_1 \wedge A_2 \wedge A_3 \Rightarrow E(a, b, d, c, d, b)$. Uma vez que pretendemos utilizar um lgoritmo de refutação deveremos provar que a fórmula

$$A_1 \wedge A_2 \wedge A_3 \wedge \neg E(a, b, d, c, b, d)$$

é inconsistente. Para o fazer, vamos transformar o conjunto constituído por esta fórmula e pelos axiomas no seguinte conjunto de claúsulas:

$$\begin{aligned} S \;=\; & \{\neg T(x, y, u, v) \vee P(x, y, u, v),\ \neg P(x, y, u, v) \vee E(x, y, v, u, v, y), \\ & \qquad T(a, b, c, d),\ \neg E(a, b, d, c, d, b)\}. \end{aligned}$$

| | | |
|---|---|---|
| (1) | $\neg T(x,y,u,v) \vee P(x,y,u,v)$ | (cláusula em $S$) |
| (2) | $\neg P(x,y,u,v) \vee E(x,y,v,u,v,y)$ | (cláusula em $S$) |
| (3) | $T(a,b,c,d)$ | (cláusula em $S$) |
| (4) | $\neg E(a,b,d,c,d,b)$ | (cláusula em $S$) |
| (5) | $\neg P(a,b,c,d)$ | (resolvente de (4) e (2)) |
| (6) | $\neg T(a,b,c,d)$ | (resolvente de (5) e (1)) |
| (7) | $\Diamond$ | (resolvente de (3) e (6)) |

Uma vez que se deduziu a cláusula vazia a partir de $S$, podemos concluir que $S$ é inconsistente, pelo que a prova fica completa.

**Exemplo 3.16** *Considere o conjunto de constatações (acerca de um dado hospital) que a seguir se indicam.*

1. *Alguns dos pacientes gostam de todos os médicos.*

2. *Nenhum paciente gosta de charlatões.*

3. *Nenhum médico é charlatão.*

*Conclua* 3. *a partir de* 1. *e* 2. *utilizando, adequadamente, fórmulas da Lógica de* $1^a$ *oredem que representem* 1., 2. *e* 3., *e verificando que a fórmula representativa de* 3. *é consequência lógica das fórmulas representativas de* 1. *e* 2.

*Resolução.*

Denotando-se por

| | |
|---|---|
| $P(x)$ : | "$x$ é paciente" |
| $M(x)$ : | "$x$ é médico" |
| $C(x)$ : | "$x$ é charlatão" |
| $G(x,y)$ : | "$x$ gosta de $y$" |

as fórmulas representativas de 1., 2. e 3., são as seguintes:

$$
\begin{aligned}
F_1 : &\quad (\exists x)(P(x) \wedge (\forall y)(M(y) \Rightarrow G(x,y))) \\
F_2 : &\quad (\forall x)\Big(P(x) \Rightarrow (\forall y)\big(C(y) \Rightarrow \neg G(x,y)\big)\Big) \\
F_3 : &\quad (\forall x)\Big(M(x) \Rightarrow \neg C(x)\Big)
\end{aligned}
$$

Seja $I$ uma interpretação arbitrária sobre um domínio $D$. Suponha-se que $F_1$ e $F_2$ são avaliadas em verdadeiro para $I$. Uma vez que $F_1$ é verdadeira, isto é, $(\exists x)(P(x) \wedge (\forall y)(M(y) \Rightarrow G(x,y)))$ é verdadeira para $I$, existe algum elemento, $\underline{e}$, em $D$ tal que

$$(a) \qquad P(e) \wedge (\forall y)\Big(M(y) \Rightarrow G(e,y)\Big)$$

é verdadeira para $I$. Por outro lado, uma vez que $\Big(P(x) \Rightarrow (\forall y)\big(C(y) \Rightarrow \neg G(x,y)\big)\Big)$ é verdadeira (para $I$) para todas as concretizações de $x$ em $D$, em particular, concluí-se que $\Big(P(e) \Rightarrow (\forall y)\big(C(y) \Rightarrow \neg G(e,y)\big)\Big)$ é verdadeira para $I$.

Uma vez que, de acordo com (a), $P(e)$ é verdadeira para $I$, pela avaliação em verdadeiro de $F_2$, vem que $(\forall y)\Big(C(y) \Rightarrow \neg G(e,y)\Big)$ é verdadeira para $I$. Consequenteente, para qualquer concretização de $y$ em $D$, tanto

$$(b) \qquad M(y) \Rightarrow G(e,y)$$

como

$$(c) \qquad C(y) \Rightarrow \neg G(e,y)$$

são verdadeiras para $I$.

Se $M(y)$ é verdadeira para $I$, então $G(e,y)$ tem de ser verdadeira para $I$ (de acordo com (a)) e, consequentemente, $C(y)$ tem de ser falsa para $I$ (de acordo com (c)).

Então

$$M(y) \Rightarrow \neg C(y)$$

é verdadeira para $I$.

Concluimos assim que $M(y) \Rightarrow \neg C(y)$ é verdadeira para toda a concretização de $y$ em $D$, o seja,

$$(\forall y)\Big(M(y) \Rightarrow \neg C(y)\Big)$$

é verdadeira para $I$.

Deste modo mostramos que se $F_1$ e $F_2$ são verdadeiras para uma interpretação arbitrária, $I$, então $(\forall y)(M(y) \Rightarrow \neg C(y))$ é verdadeira para $I$, ou seja, concluímos que $F_3$ é consequência lógica de $F_1$ e $F_2$.

Alternativamente a prova de que $F_3$ é consequência lógica de $F_1$ e $F_2$, pode ser obtida, por aplicação do Princípio da Resolução, actuando do seguinte modo.

Para se provar que $F_1 \wedge F_2 \Rightarrow F_3$ é uma fórmula válida, vamos provar que $F_1 \wedge F_2 \wedge \neg F_3$ é inconsistente.

$F_1$ dá origem às cláusulas:

(1) $P(a)$

(2) $\neg M(y) \vee G(a, y)$

$F_2$ dá origem à cláusula:

(3) $\neg P(x) \vee \neg C(y) \vee \neg G(x, y)$

$\neg F_3$ dá origem às cláusulas:

(4) $M(b)$

(5) $C(b)$

Utilizando o Princípio da Resolução obtém-se

| | | |
|---|---|---|
| (6) | $G(a, b)$ | (resolvente de (4) e (2)) |
| (7) | $\neg C(y) \vee \neg G(a, y)$ | (resolvente de (3) e (1)) |
| (8) | $\neg G(a, b)$ | (resolvente de (5) e (7)) |
| (9) | $\Diamond$ | (resolvente de (6) e (8)) |

Vamos tentar traduzir em linguagem natural a demonstração anterior.

(a) A partir de $F_1$, podemos asumir que existe um paciente $\underline{a}$ que gosta de todos os médicos (cláusulas (1) e (2)).

(b) Suponhamos que a conclusão pretendida ($F_3$) está errada, isto é, suponhamos que $\underline{b}$ é ao mesmo tempo um médico e um charlatão (cláusulas (4) e (5)).

(c) Uma vez que o paciente $\underline{a}$ gosta de todos os médicos, $\underline{a}$ gosta de $\underline{b}$ (cláusula (6)).

(d) Uma vez que $\underline{a}$ é um paciente $\underline{a}$ não gosta de nenhum charlatão (cláusula (7)).

(e) Contudo, $\underline{b}$ é um charlatão. Logo, $\underline{a}$ não gosta de $\underline{b}$ (cláusula (8)).

(f) Isto não é possível, devido a (c), completando-se assim a prova.

**Exercício 3.6** *Encontre as possíveis resolventes (se existirem) dos seguintes pares de cláusulas:*

1. $C\,:\,\neg P(x) \vee Q(x,b),\ D\,:\,P(a) \vee Q(a,b)$

2. $C\,:\,\neg P(x) \vee Q(x,x),\ D\,:\,\neg Q(a,f(a))$

**Teorema 3.4** *Se $C_1'$ e $C_2'$ são concretizações (exemplos) de $C_1$ e $C_2$, respectivamente, e se $C'$ é uma resolvente de $C_1'$ e $C_2'$, então existe uma resolvente de $C_1$ e $C_2$, $C$, tal que $C'$ é uma concretização (exemplo) de $C$.*

*Demonstração.*

Se necessário, mudem-se as designações das variáveis de $C_1$ e/ou $C_2$, de tal modo que as variáveis em $C_1$ sejam diferentes das variáveis em $C_2$. Sejam $L_1'$ e $L_2'$ os literais que dão origem à resolvente de $C_1'$ e $C_2'$, e seja,

$$C' = (C_1'\gamma - L_1'\gamma) \cup (C_2'\gamma - L_2'\gamma)$$

onde $\gamma$ é o unificador mais geral de $L'_1$ e $\neg L'_2$. Uma vez que $C'_1$ e $C'_2$ são concretizações de $C_1$ e $C_2$, respectivamente, existe uma substituição $\Theta$ tal que $C'_1 = C_1\Theta$ e $C'_2 = C_2\Theta$.

Sejam $L_i^1, \ldots, L_i^{k_i}$ os literais de $C_i$ que correspondem a $L'_i$ (ou seja, $L_i^1\theta = \cdots = L_i^{k_i}\theta = L'_i$), $i = 1, 2$.

Se $k_i > 1$, obtém-se $\lambda_i$, o unificador mais geral para $\{L_i^1, \ldots, L_i^{k_i}\}$ e $L_i^1 = L_i^1\lambda_i$, $i = 1, 2$. (Note-se que $L_i^1\lambda_i, \ldots, L_i^{k_i}\lambda_i$ são todos iguais, uma vez que $\lambda_i$ é o unificador mais geral). Então $L_i$ é um literal no factor de $C_i$, $C_i\lambda_i$.

Se $k_i = 1$, seja $\lambda_i = \varepsilon$ e $L_i = L_i^1\lambda_i$. Claramente, $L'_i$ é uma concretização de $L_i$. Uma vez que $L'_1$ e $\neg L'_2$ são unificáveis, $L_1$ e $\neg L_2$ são também unificáveis. Seja $\lambda = \lambda_1 \cup \lambda_2$ e seja $\sigma$ o unificador mais geral para $L_1$ e $\neg L_2$. Seja

$$\begin{aligned}
C &= ((C_1\lambda)\sigma - L_1\sigma) \cup (C_2\lambda)\sigma - L_2\sigma) \\
&= ((C_1\lambda)\sigma - (\{L_1^1, \cdots, L_1^{k_1}\}\lambda)\sigma \cup ((C_2\lambda)\sigma - (\{L_2^1, \cdots, L_2^{k_2}\}\lambda)\sigma \\
&= (C_1(\lambda o\sigma) - \{L_1^1, \cdots, L_1^{k_1}\}(\lambda o\sigma)) \cup (C_2(\lambda o\sigma) - \{L_2^1, \cdots, L_2^{k_2}\}(\lambda o\sigma)).
\end{aligned}$$

$C$ é uma resolvente de $C_1$ e $C_2$ e, claramente, $C'$ é uma concretização de $C$, dado que

$$\begin{aligned}
C' &= (C'_1\gamma - L'_1\gamma) \cup (C'_2\gamma - L'_2\gamma) \\
&= \Big((C_1\theta)\gamma - \big(\{L_1^1, \cdots, L_1^{k_1}\}\theta\big)\gamma\Big) \cup \Big((C_2\theta)\gamma - \big(\{L_2^2, \cdots, L_2^{k_2}\}\theta\big)\gamma\Big) \\
&= \Big(C_1(\theta o\gamma) - \{L_1^1, \cdots, L_1^{k_1}\}(\theta o\gamma)\Big) \cup \Big(C_2(\theta o\gamma) - \{L_2^2, \cdots, L_2^{k_2}\}(\theta o\gamma)\Big)
\end{aligned}$$

e $(\lambda o\sigma)$ é mais geral do que $(\theta o\gamma)$, completando-se assim a prova deste teorema. □

A partir deste teorema concluimos que, se uma dada cláusula é deduzida de um conjunto de exemplos (concretizações) do conjunto inicial, então essa cláusula será um exemplo (concretização) de uma certa cláusula que se pode deduzir directamente do conjunto inicial. Como consequência, deduzindo-se a cláusula vazia a partir de um conjunto de exemplos (concretizações), conclui-se que existe uma cláusula, que se pode deduzir de um conjunto inicial, relativamente à qual a cláusula vazia é um exemplo (concretização). Uma vez que a cláusula vazia só é um exemplo dela própria, este facto garante-nos a inconsistência do conjunto inicial.

Prova-se também que o princípio da resolução é completo, no sentido em que a sua aplicação, a um conjunto de cláusulas, conduz à dedução da cláusula vazia sse esse conjunto é inconsistente. No entanto, a prova deste resultado requer a introdução do conceito de árvore semântica, bem como uma série de conceitos associados (como sejam o de árvore semântica completa, árvore semântica fechada, etc.) e tem por base um teorema demonstrado por Herbrand que fundamenta certas técnicas dedutivas que lidam com o que se designa por universo de Herbrand de um conjunto de cláusulas. Os leitores mais interessados deverão consultar o capítulo 4 de [2].

# 4 Implementação do Princípio de Resolução

Um modo imediato de se implementar a aplicação do Princípio de Resolução a um conjunto de cláusulas, $S$, consiste na determinação de todas as resolventes dos pares de clásulas de $S$, juntando estas resolventes ao conjunto $S$, determinando-se, posteriormente, as resolventes que decorrem do conjunto obtido, e assim sucessivamente, até que se obtenha a clásula vazia $\Diamond$. Desta forma, gera-se a sucessão $S^0, S^1, S^2, \cdots$, onde

$$S^0 = S;$$

$$S^n = \{\text{resolvente de } C_1 \text{ e } C_2 \mid C_1 \in S^0 \cup S^1 \cup S^2 \cup \cdots \cup S^{n-1} \text{ e } C_2 \in S^{n-1}\},$$
$$n = 1, 2, \cdots$$

Embora o Princípio da Resolução seja mais eficiente do que os procedimentos de Herbrand, com esta técnica de implementação, podem ainda gerar-se muitas cláusulas redundantes, como sejam, cláusulas já existentes (repetição de cláusulas) e tautologias. Note-se que uma tautologia é verdadeira qualquer que seja a interpretação, pelo que se a eliminarmos, de um conjunto inconsistente de cláusulas, o restante subconjunto permanece inconsistente.

Se evitarmos a geração de tautologias, a técnica anteriormente referida pode ser eficientemente implementada numa folha de cálculo, associando-se cada coluna a cada um dos literais que compõem as cláusulas. As cláusulas, por sua vez, são determinadas pelas células que compõem uma mesma linha, as quais tomam o valor 1 se o correspondente literal (dessa coluna) aparece não negado na cláusula, o valor $-1$ no caso do literal em questão aparecer negado e o valor 0 se o respectivo literal não faz parte da cláusula.

**Exemplo 4.1** *Sendo* $S^0 = \{P, \neg U, \neg S \vee U, \neg P \vee S\}$ *a representação de* $S^0$ *na folha de cálculo vem dada por :*

| | *P* | *S* | *U* | *Cláusula* |
|---|---|---|---|---|
| *(1ª)* | *1* | *0* | *0* | $P$ |
| *(2ª)* | *0* | *0* | *-1* | $\neg U$ |
| *(3ª)* | *0* | *-1* | *1* | $\neg S \vee U$ |
| *(4ª)* | *-1* | *1* | *0* | $\neg P \vee S$ |

A obtenção de $S^1, S^2, \cdots$ , faz-se pela "adição" dos pares de linhas que contenham apenas uma das células, associadas à mesma coluna, com sinais contrários. Com esta "adição" (que denotaremos por $\oplus$), no entanto, devem obter-se os seguintes resultados:

| $\oplus$ | -1 | 0 | 1 |
|---|---|---|---|
| -1 | -1 | -1 | 0 |
| 0 | -1 | 0 | 1 |
| 1 | 0 | 1 | 1 |

Note-se que nos casos em que existem linhas com mais do que uma coluna contendo células de sinais contrários, isso significa que a correspondente resolvente seria uma tautologia, pelo que a sua dedução (de acordo com este procedimento) é evitada.

Com a finalidade de se facilitar a pesquisa de diferentes linhas com estas características, poder-se-à ordenar o conjunto inicial (de linhas) pela ordem lexicográfica.

Como ilustração apresenta-se a seguir a resolução do exemplo 4.1.

| | Nº da cláusula | P | S | U |
|---|---|---|---|---|
| $S^0$ | (1ª) | 1 | 0 | 0 |
| | (2ª) | 0 | 0 | -1 |
| | (3ª) | 0 | -1 | 1 |
| | (4ª) | -1 | 1 | 0 |
| $S^1$ | (5ª) = (1ª) $\oplus$ (4ª) | 0 | 1 | 0 |
| | (6ª) = (2ª) $\oplus$ (3ª) | 0 | -1 | 0 |
| | (7ª) = (3ª) $\oplus$ (4ª) | -1 | 0 | 1 |
| $S^2$ | (8ª) = (1ª) $\oplus$ (7ª) | 0 | 0 | 1 |
| | (9ª) = (2ª) $\oplus$ (7ª) | -1 | 0 | 0 |
| | (10ª) = (3ª) $\oplus$ (5ª) | 0 | 0 | 1 |
| | (11ª) = (4ª) $\oplus$ (6ª) | -1 | 0 | 0 |
| | (12ª) = (5ª) $\oplus$ (6ª) | 0 | 0 | 0 |

**Exemplo 4.2** *Sendo* $S^0 = \{P, Q, \neg S, \neg P \vee \neg Q \vee R, \neg P \vee \neg Q \vee S\}$*, a representação de* $S^0$ *na folha de cálculo vem dada por:*

| | $P$ | $Q$ | $R$ | $S$ | *Cláusula* |
|---|---|---|---|---|---|
| *(1ª)* | *1* | *0* | *0* | *0* | $P$ |
| *(2ª)* | *0* | *1* | *0* | *0* | $Q$ |
| *(3ª)* | *0* | *0* | *0* | *-1* | $\neg S$ |
| *(4ª)* | *-1* | *-1* | *1* | *0* | $\neg P \vee \neg Q \vee R$ |
| *(5ª)* | *-1* | *-1* | *0* | *1* | $\neg P \vee \neg Q \vee S$ |

Segue-se a respectiva resolução:

| | | | | | |
|---|---|---|---|---|---|
| $S^1$ | $(6^{\mathrm{a}}) = (1^{\mathrm{a}}) \oplus (4^{\mathrm{a}})$ | 0 | -1 | 1 | 0 |
| | $(7^{\mathrm{a}}) = (1^{\mathrm{a}}) \oplus (5^{\mathrm{a}})$ | 0 | -1 | 0 | 1 |
| | $(8^{\mathrm{a}}) = (2^{\mathrm{a}}) \oplus (4^{\mathrm{a}})$ | -1 | 0 | 1 | 0 |
| | $(9^{\mathrm{a}}) = (2^{\mathrm{a}}) \oplus (5^{\mathrm{a}})$ | -1 | 0 | 0 | 1 |
| | $(10^{\mathrm{a}}) = (3^{\mathrm{a}}) \oplus (5^{\mathrm{a}})$ | -1 | -1 | 0 | 0 |
| $S^2$ | $(11^{\mathrm{a}}) = (1^{\mathrm{a}}) \oplus (8^{\mathrm{a}})$ | 0 | 0 | 1 | 0 |
| | $(12^{\mathrm{a}}) = (1^{\mathrm{a}}) \oplus (9^{\mathrm{a}})$ | 0 | 0 | 0 | 1 |
| | $(13^{\mathrm{a}}) = (1^{\mathrm{a}}) \oplus (10^{\mathrm{a}})$ | 0 | -1 | 0 | 0 |
| | $(14^{\mathrm{a}}) = (2^{\mathrm{a}}) \oplus (6^{\mathrm{a}})$ | 0 | 0 | 1 | 0 |
| | $(15^{\mathrm{a}}) = (2^{\mathrm{a}}) \oplus (7^{\mathrm{a}})$ | 0 | 0 | 0 | 1 |
| | $(16^{\mathrm{a}}) = (2^{\mathrm{a}}) \oplus (10^{\mathrm{a}})$ | -1 | 0 | 0 | 0 |
| | $(17^{\mathrm{a}}) = (3^{\mathrm{a}}) \oplus (7^{\mathrm{a}})$ | 0 | -1 | 0 | 0 |
| | $(18^{\mathrm{a}}) = (3^{\mathrm{a}}) \oplus (9^{\mathrm{a}})$ | -1 | 0 | 0 | 0 |
| | $(19^{\mathrm{a}}) = (1^{\mathrm{a}}) \oplus (16^{\mathrm{a}})$ | 0 | 0 | 0 | 0 |

Como este exemplo ilustra, com esta técnica ainda se verifica o aparecimento de cláusulas repetidas, facto que prejudica a correspondente eficiência computacional. Com o intuito de a melhorar, desenvolveram-se algumas variantes da aplicação do princípio da resolução. Três destas variantes aparecem bem descritas em [2].

# Referências Bibliográficas


[1] Cardoso, D.M., Programação em Lógica e Demonstração Automática de Teoremas, Cadernos de Matemática CM/D-03,Universidade de Aveiro, 1995.

[2] Chang, C.L. e R.C.T. Lee, Symbolic Logic and Mechanical Theorem Proving, Academic Press, Inc, 1987.

[3] Church, A., An unsolvable problem of number theory, Amer. J. Math. 58, 2, 345–363, 1936.

[4] Gilmore, P.C., A proof method for quantification theory; its justification and realization,IBM J. Res. Develop., 28–35, 1960.

[5] Godel, K., The completness of the axioms of the functional calculus of logic, "From Frege to Gödel: a Source Book in Matematical logic", (J. van Heijenoor, ed.), Havard Univ. Press, Cambridge, Massachusetts, 1930.

[6] Herbrand, J., Investigation in proff theory: the properties of the propositions, "From Frege to Gödel: a Source Book in Matematical logic", (J. van Heijenoor, ed.), Havard Univ. Press, Cambridge, Massachusetts, 1930.

[7] Robinson, J.A., A machine oriented logic based on the resolution principle, J. ACM, 10, n.2, 163–174, 1963.

[8] Turing, A.M. , On computable numbers, with an application to the entscheindungs-problem, Proc. London Math. Soc., 42, 230–265, 1936.