Noções de Lógica Matemática

Domingos Moreira Cardoso
Maria Paula Carvalho

Janeiro, 2007

(versão revista em março de 2015)

Conteúdo

1.1 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.
Trata-se de 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.

1.2 Noções fundamentais

> *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**

1.2.1 Interpretação e inferência no 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.

Como já referimos, a lógica proposicional utiliza proposições atómicas, também designadas por átomos ou fórmulas atómicas (denotadas geralmente por letras minúsculas) e as cinco conexões: $\neg$ (não), $\wedge$ (e), $\vee$ (ou), $\Rightarrow$ (se... então), $\Leftrightarrow$ (sse) para representar as suas fórmulas bem formadas (fbf).

Ao longo do capítulo, dizemos 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

$$F\ :\ ((p \Rightarrow q) \vee p) \Rightarrow q$$

e tendo em conta que os átomos desta fórmula são p e q, tem-se que G admite as $2^2 = 4$ interpretações (atribuições de valores lógicos a todas as variáveis proposicionais) apresentadas na tabela de verdade que a seguir se indica.

p	q	$p \Rightarrow q$	$(p \Rightarrow q) \wedge p$	$((p \Rightarrow q) \wedge p) \Rightarrow$q
1	1	1	1	1
1	0	0	0	1
0	1	1	0	1
0	0	1	0	1

Concluímos assim que a fórmula F é verdadeira qualquer que seja a interpretação de entre as quatro possíveis. As fórmulas com estas características (fórmulas que são verdadeiras qualquer que seja a interpretação) designam-se por tautologias. Por sua vez,

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

é uma fórmula falsa qualquer que seja a interpretação, ou seja, é uma contradição.

Definição 1.2.1 (Formula válida, não válida, inconsistente e consistente). *Uma fbf diz-se válida se é verdadeira sobre qualquer das suas possíveis interpretações e diz-se não válida (ou inválida) se não é válida. Uma fbf diz-se inconsistente se é falsa qualquer que seja a interpretação e diz-se consistente se não é inconsistente.*

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

Seja F um conjunto de fbfs, $F_n \in F$ uma fórmula composta por n átomos e $T = \{1, 0\}$. Então F_n induz uma função $F_n : T^n \to T$, para a qual se verifica que

F_n é uma tautologia sse

$$\begin{array}{rcl} F_n : T^n & \to & T \\ I & \mapsto & F_n(I) = 1. \end{array}$$

Por sua vez, F_n é uma contradição sse

$$\begin{array}{rcl} F_n : T^n & \to & T \\ I & \mapsto & F_n(I) = 0. \end{array}$$

Dado 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 verificação exaustiva de todas as suas possíveis interpretações.

Definição 1.2.2 (Fórmulas equivalentes). *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.*

Definição 1.2.3 (Literal e literal complementar). *Um literal é um átomo ou a negação de um átomo. Dois literais dizem-se complementares quando um é a negação do outro.*

Definição 1.2.4 (Formas normais conjuntiva e disjuntiva). *Seja F uma fórmula.*

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

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

Exemplo 1.2.1. *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.*

Definição 1.2.5 (Consequência lógica). *Dadas as fórmulas $F_1, F_2, \ldots, F_n$ e uma fórmula G, dizemos que G é consequência lógica de $F_1, F_2, \ldots, F_n$ (também se diz que G se obtém logicamente de $F_1, F_2, \ldots, F_n$) e escreve-se $F_1, F_2, \ldots, F_n \models G$, se para toda a interpretação I na qual $F_1 \wedge F_2 \wedge \cdots \wedge F_n$ é verdadeira, G também é verdadeira. As fórmulas $F_1, F_2, \ldots, F_n$ designam-se por axiomas, postulados, hipóteses ou premissas de G.*

Teorema 1.2.1. *Dadas as fórmulas* $F_1, F_2, \ldots, F_n$ *e* G*, a fórmula* G *é consequência lógica de* $F_1, F_2, \ldots, F_n$ *sse* $(F_1 \wedge F_2 \wedge \cdots \wedge F_n) \Rightarrow G$ *é uma fórmula válida (ou seja, é uma tautologia).*

Demonstração. Vamos fazer a prova das condições necessária e suficiente, separadamente.

> ($\Rightarrow$) Suponhamos que G é consequência lógica de $F_1, F_2, \ldots, F_n$ 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, a fórmula $(F_1 \wedge F_2 \wedge \cdots \wedge F_n \Rightarrow G$ 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 (recorde-se 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 $F_1, F_2, \ldots, F_n$.

■

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

Muitas vezes, para concluirmos que G é consequência lógica de $F_1, F_2, \ldots, F_n$ (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 $F_1, F_2, \ldots, F_n$, que F_{n+2} é consequência lógica de $F_1, F_2, \ldots, F_n, F_{n+1}$, etc, até concluirmos que G é consequência lógica de $F_1, F_2, \ldots, F_n, F_{n+1}, \ldots, F_{n+m}$, o que nos permite deduzir o inicialmente pretendido, ou seja, que G é consequência lógica de $F_1, F_2, \ldots, F_n$. 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 $F_1, F_2, \ldots, F_n$, sendo F_{n+2} consequência lógica de $F_1, F_2, \ldots, F_n, F_{n+1}$, etc, e sendo F_{n+m} consequência lógica de $F_1, F_2, \ldots, F_n, F_{n+1}, \ldots, F_{n+m-1}$ com $(m \geq 1)$, então sempre que $F_1 \wedge F_2 \wedge \cdots \wedge F_n$ é verdadeira, vem que $F_{n+1}, \ldots, F_{n+m}$ 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 uma tautologia e reciprocamente.

Teorema 1.2.2. *Dadas as fórmulas* $F_1, F_2, \ldots, F_n$ *e* G*, a fórmula* G *é consequência lógica de* $F_1, F_2, \ldots, F_n$ *sse* $(F_1 \wedge F_2 \wedge \cdots \wedge F_n \wedge \neg G)$ *é inconsistente.*

Demonstração. Pelo Teorema 1.2.1, 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 $F_1, F_2, \ldots, F_n$ 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).

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

■

Neste contexto, o símbolo "$\equiv$" 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), ou seja, são fórmulas equivalentes.

Este teorema tem a especial importância de nos permitir 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 1.2.1 e 1.2.2.

Exemplo 1.2.2. *Dadas as fórmulas* $F_1 : (p \Rightarrow q)$, $F_2 : \neg q$ *e* $G : \neg p$, *vamos provar que* G *é consequência lógica de* F_1 *e* F_2.

Solução. Para a obtenção desta prova podemos actuar de qualquer uma das três formas seguintes:

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

2. Utilizar o Teorema 1.2.1 e avaliar a tabela de verdade da fórmula
$$((p \Rightarrow q) \wedge \neg q) \Rightarrow \neg p.$$

3. Utilizar o Teorema 1.2.2 e avaliar a tabela de verdade da fórmula
$$((p \Rightarrow q) \wedge \neg q) \wedge p.$$

□

1.2.2 Introdução à lógica de primeira 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 pelo cálculo proposicional.

Exemplo 1.2.3. *Vamos analisar a seguinte frase: Todo o homem é mortal. Uma vez que Confucios é homem então é mortal. Nesta frase existem as seguintes três proposições:*

p: "todo o homem é mortal";

q: "Confucios é homem";

r: "Confucios é mortal".

Embora o nosso raciocínio nos leve a concluir que a proposição r é consequência lógica das proposições p e q, dentro da metodologia de trabalho do cálculo proposicional, r não é consequência lógica de p e q.

Relativamente ao cálculo proposicional, a lógica de primeira ordem lida com mais três noções lógicas que se designam por

termos, predicados e quantificadores.

Na Lógica de primeira ordem, podemos utilizar símbolos de funções (por exemplo, *plus(x,y)* para significar $x+y$), podemos utilizar símbolos de relações (por exemplo, *Greater(x,y)* para significar $x > y$), etc.

Para construirmos um átomo na lógica de primeira ordem, utilizamos, em geral, os seguintes símbolos:

(a) Símbolos individuais de variáveis e de constantes (nomes de objectos).

(b) Símbolos de funções.

(c) Símbolos de predicados.

Definição 1.2.6 (Termos da lógica de primeira ordem)**.** *Os termos da lógica de primeira 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 $t_1, t_2, \ldots, t_n$ são termos, então $f(t_1, t_2, \ldots, t_n)$ é um termo;*

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

Exemplo 1.2.4. *Seguem-es quatro exemplos de termos.*

1. 12;

2. y;

3. *adicionar(adicionar(x,1),x);*

4. *pai_ de(luisa).*

Definição 1.2.7 (Predicado)**.** *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 1.2.8 (Átomo da lógica de primeira ordem)**.** *Se P é um predicado com n argumentos e $t_1, t_2, \ldots, t_n$ são termos, então $P(t_1, t_2, \cdots, t_n)$ é um átomo. Nenhuma outra expressão pode ser um átomo.*

Uma vez introduzido o conceito de átomo no contexto mais amplo da lógica de primeira ordem, as definições dos conceitos de literal e literal complementar anteriormente apresentadas (Definição 1.2.3) mantêm-se válidas neste novo contexto e podemos utilizar os conectivos lógicos do cálculo proposicional para construirmos fórmulas, acrescentadas dos respectivos quantificadores universal e existencial (dado que se consideram variáveis).

Exemplo 1.2.5. *Apresentam-se três exemplos de fórmulas da lógica de primeira ordem (ou de predicados).*

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

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

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

Um conceito importante, relacionado com os quantificadores, é o de *alcance* (*scope*) do quantificador, o qual corresponde à parte da fórmula 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 1.2.9 (Ocorrências ligadas e ocorrências livres)**.** *Uma ocorrência de uma variável numa fórmula é ligada ("bound") se a ocorrência está dentro do alcance do quantificador usado nessa variável. Uma ocorrência de uma variável numa fórmula é livre se a ocorrência dessa variável não é ligada.*

Definição 1.2.10 (Variável livre e variável ligada)**.** *Uma variável é livre numa fórmula se no mínimo uma sua ocorrência na fórmula é livre. Uma variável é ligada se 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 1.2.11 (Fórmulas bem formadas da lógica de primeira ordem)**.** *As fórmulas bem formadas (fbfs), da lógica de primeira 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.

1.2.3 Interpretação das fórmulas da lógica de primeira ordem

Enquanto na lógica proposicional, as interpretações ficam completamente determinadas pelas tabelas de verdade dos átomos, na lógica de primeira ordem, a existência de termos, nomeadamente constantes, variáveis e funções, obriga a outras considerações.

Definição 1.2.12. *Uma interpretação de uma fórmula F na lógica de primeira 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 $\{0,1\}$.*

Por vezes, quando nos referimos ao domínio 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".

Qualquer que seja a interpretação de uma fórmula sobre um domínio D, a fórmula é avaliada em verdadeiro (1) ou falso (0), 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 avaliados.

2. $(\forall x)G$ é avaliado em 1 se G é avaliada em 1 para todas as concretizações possíveis de x em D. Caso contrário o seu valor é 0.

3. $(\exists x)G$ é avaliado em 1 se G é avaliada em 1 para pelo menos uma concretização de x em D. Caso contrário, o seu valor é 0.

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

	$D = \{1,2\}$	
$P(1) = 1$	*e*	$P(2) = 0.$

Para esta interpretação, com facilidade se conclui que a primeira fórmula é falsa e a segunda é verdadeira.

Exemplo 1.2.7. *Considerando 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, $\underline{Q}$, vamos determinar uma interpretação para a qual esta fórmula seja verdadeira.*

Solução. Considere-se 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 $\underline{Q}$:	$P(1)$	$P(2)$	$Q(1,1)$	$Q(1,2)$	$Q(2,1)$	$Q(2,2)$
	0	1	1	1	0	1

Com esta interpretação,

para $x = 1$, a fórmula $P(x) \Rightarrow Q(f(x), a)$ tem a avaliação

$$(P(1) \Rightarrow Q(f(1), a)) \equiv (P(1) \Rightarrow Q(2, 1)) \equiv (0 \Rightarrow 0) \equiv 1,$$

e para $x = 2$, a fórmula $P(x) \Rightarrow Q(f(x), a)$ tem a avaliação

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

Logo, 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 a fórmula $(\forall x)(P(x) \Rightarrow Q(f(x), a))$ é verdadeira para a interpretação I. □

Definição 1.2.13 (Fórmula consistente e modelo de uma fórmula)**.** *Uma fórmula G é consistente se existe uma interpretação I tal que G é avaliada em 1 para I. Se uma fórmula G toma o valor 1 numa interpretação I, dizemos que I é um modelo de G e que I satisfaz G.*

Definição 1.2.14 (Fórmula inconsistente e fórmula válida)**.** *Uma fórmula G é inconsistente se não existe uma interpretação de G que satisfaça G. Uma fórmula G diz-se válida sse toda a interpretação de G satisfaz G.*

Definição 1.2.15 (Consequência lógica). *Uma fórmula G é consequência lógica das fórmulas $F_1, F_2, \ldots, F_n$ quando para toda a interpretação I, se $\bigwedge_{i=1}^{n} F_i$ é verdadeira em I então G também é verdadeira.*

Os Teoremas 1.2.1 e 1.2.2 sobre validade (inconsistência) e consequência lógica no cálculo proposicional são também válidos para a lógica de primeira ordem.

A lógica de primeira ordem pode ser considerada como uma extensão da lógica proposional. Quando uma fórmula da lógica de primeira ordem não contém variáveis nem quantificadores, pode ser tratada como uma fórmula do cálculo proposicional.

1.2.4 Formas normais prenex da lógica de primeira ordem

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

Definição 1.2.16. *Uma fórmula F da lógica de primeira ordem diz-se na forma normal prenex se a fórmula F tem a forma*

$$(Q_1x_1)(Q_2x_2)\cdots(Q_nx_n)M$$

onde (Q_ix_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 1.2.8. *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, quando essa particularidade é relevante, vamos identificar por $F[x]$ e sendo G uma fórmula que não contém x, a transformação de fórmulas da lógica de primeira ordem na forma normal disjuntiva prenex pode fazer-se de acordo com o seguinte procedimento:

Passo 1 Eliminação das conexões $\Rightarrow$ e $\Leftrightarrow$:

$$F \Leftrightarrow G \equiv (F \Rightarrow G) \wedge (G \Rightarrow F) \tag{1.1}$$

$$F \Rightarrow G \equiv \neg F \vee G. \tag{1.2}$$

Passo 2 Aplicação das Leis de Morgan e dupla negação:

$$\begin{aligned} \neg(\neg F) &\equiv F && (1.3)\\ \neg(F \vee G) &\equiv \neg F \wedge \neg G && (1.4)\\ \neg(F \wedge G) &\equiv \neg F \vee \neg G && (1.5) \end{aligned}$$

Passo 3 Posicionamento das negações imediatamente antes dos átomos:

$$\begin{aligned} \neg((\forall x)F[x]) &\equiv (\exists x)(\neg F[x]) && (1.6)\\ \neg((\exists x)F[x]) &\equiv (\forall x)(\neg F[x]) && (1.7) \end{aligned}$$

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

$$\begin{aligned} (Qx)F[x] \vee G &\equiv (Qx)(F[x] \vee G) && (1.8)\\ (Qx)F[x] \wedge G &\equiv (Qx)(F[x] \wedge G) && (1.9)\\ (\forall x)F[x] \wedge (\forall x)G[x] &\equiv (\forall x)(F[x] \wedge G[x]) && (1.10)\\ (\exists x)F[x] \vee (\exists x)G[x] &\equiv (\exists x)(F[x] \vee G[x]) && (1.11)\\ (Q_1x)F[x] \wedge (Q_2x)G[x] &\equiv (Q_1x)(Q_2z)(F[x] \wedge G[z]) && (1.12)\\ (Q_3x)F[x] \vee (Q_4x)G[x] &\equiv (Q_3x)(Q_4z)(F[x] \vee G[z]) && (1.13) \end{aligned}$$

onde Q_1, Q_2, Q_3 e Q_4 denotam os quantificadores universal ($\forall$) ou existencial ($\exists$)

Seguem-se dois exemplos de aplicação deste procedimento.

Exemplo 1.2.9. *Vamos transformar a fórmula* $(\forall x)P(x) \Rightarrow (\exists x)Q(x)$ *na forma normal conjuntiva prenex.*

Solução.

$$\begin{aligned} (\forall x)P(x) \Rightarrow (\exists x)Q(x) &\equiv \neg((\forall x)P(x)) \vee (\exists x)Q(x) && \text{por (1.2)}\\ &\equiv (\exists x)(\neg P(x)) \vee (\exists x)Q(x) && \text{por (1.6)}\\ &\equiv (\exists x)(\neg P(x) \vee Q(x)) && \text{por (1.11).} \end{aligned}$$

□

Exemplo 1.2.10. *Vamos 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 normal conjuntiva prenex.

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

$$\begin{aligned}
&\equiv (\forall x)(\forall y)\big(\neg((\exists z)(P(x,z) \wedge P(y,z)) \vee (\exists u)Q(x,y,u)\big) && \text{por (1.2)}\\
&\equiv (\forall x)(\forall y)\big((\forall z)(\neg P(x,z) \vee \neg P(y,z)) \vee (\exists u)Q(x,y,u)\big) && \text{por (1.5) e (1.7)}\\
&\equiv (\forall x)(\forall y)(\forall z)(\exists u)(\neg P(x,z) \vee \neg P(y,z) \vee Q(x,y,u)) && \text{por (1.13)}
\end{aligned}$$

□

1.3 Dedução computacional

> *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 primeira 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 formulas 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. No entanto, muitas fórmulas da lógica de primeira ordem, intuitivamente válidas, permanecem por provar em computador num tempo razoável.

Nesta altura, convém notar que embora os conceitos de fórmula demonstrável (ou seja, teorema) e fórmula válida sejam diferentes (dado que a primeira se constata à custa de procedimentos puramente sintácticos enquanto a segunda se conclui à custa de procedimentos semânticos, isto é, de interpretações), o facto de Godel (resp., Post) ter provado em ([5], 1930) (resp. 1921) que uma fórmula da lógica de primeira ordem (resp., cálculo proposicional) é um teorema sse é válida, permite-nos lidar indiferentemente com qualquer destes conceitos.

1.3.1 Eliminação de quantificadores existenciais

Os procedimentos de refutação aplicados a fórmulas da lógica de primeira ordem, assentam nos seguintes procedimentos:

1. Qualquer fórmula da lógica de primieira 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. Com esse objectivo, em geral, utilizam-se as designadas funções de Skolem, conforme a seguir se indica.

   Seja F a fórmula na forma normal prenex

   $$(Q_1x_1)\cdots(Q_nx_n)M,$$

   onde M está na forma normal conjuntiva e suponhamos que Q_r é um quantificador existencial.

   (a) Se nenhum quantificador universal aparece à esquerda de Q_r, então escolhemos uma nova constante c diferente de qualquer das outras que ocorrem em M, substituímos x_r por c e eliminamos Q_rx_r.

   (b) Se $Q_{s_1},\ldots,Q_{s_m}$ são todos os quantificadores universais que ocorrem à esquerda de Q_r $(1 \leq s_1 < \cdots < s_m \leq n)$, então escolhemos um novo símbolo de função f diferente dos já existentes, com m argumentos nas variáveis $x_{s_1}, \ldots, x_{s_m}$ e substituímos x_r em M por $f(x_{s_1}, \ldots, x_{s_m})$ e eliminamos $(Q_r x_r)$.

Este procedimento deve ser aplicado a todos os quantificadores existenciais de modo a obter-se a fórmula no que se designa por *forma standard de Skolem* ou, simplesmente, *forma standard*.

Exemplo 1.3.1. *Vamos reduzir cada uma das fórmulas a seguir indicadas à forma standard de Skolem.*

1. $(\exists x)(\forall y)(\forall z)(\exists u)(\forall v)(\exists w)P(x,y,z,u,v,w)$.

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

Solução. Vamos aplicar o conjunto de regras anteriormente descritas a cada uma das fórmulas.

1. No primeiro caso, obtém-se

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

2. No segundo caso, a redução à forma standard de Skolem, numa fase intermédia, dá

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

   Posteriormente, obtém-se

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

$\square$

Antes de passarmos à próxima definição, convém lembrar que um literal é um átomo ou a negação de um átomo.

Definição 1.3.1 (Cláusula, r-cláusula e cláusula vazia)**.** *Uma cláusula é uma disjunção finita de (um ou mais) literais. Uma r-cláusula é uma cláusula de r literais, designando-se por cláusula vazia a cláusula sem literais.*

Exemplo 1.3.2. *São exemplos de cláusulas as seguintes disjunções de predicados:*

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

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.

Exemplo 1.3.3. *O conjunto de cláusulas*

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

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.

1.3.2 Princípio da resolução de Robinson

A ideia essencial do princípio da resolução, introduzido por Robinson ([7], 1963), consiste em verificar se um dado conjunto de cláusulas S contém a cláusula vazia (que denotamos por $\diamondsuit$) ou se ela pode ser deduzida de S.

> **Observação**: Note-se que uma cláusula é verdadeira se no mínimo um dos seus literais é verdadeiro. Como a cláusula vazia, $\diamondsuit$, não tem literais, podemos afirmar que nenhum dos seus literais é verdadeiro e, consequentemente, a cláusula vazia é falsa.

O *princípio da resolução* pode ser visto como uma regra de inferência que deve ser utilizada para gerar novas cláusulas, a partir de um conjunto S de cláusulas, de acordo com o seguinte procedimento:

Procedimento da resolução.

- 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 $C_{1;2}$, assim construída, diz-se uma resolvente de C_1 e C_2.
- Considerando cada cláusula como sendo o conjunto dos seus literais e, para simplificar, utilizando a a mesma notação, quer para a cláusula propriamente dita (disjunção de literais), quer para o conjunto dos seus literais, podemos descrever esta operação de resolução escrevendo

$$C_{1;2} = (C_1 - L_1) \cup (C_2 - L_2).$$

Exemplo 1.3.4. *Vamos verificar se os pares de cláusulas a seguir indicados admitem uma resolvente.*

1. $C_1 : P \vee R$ *e* $C_2 : \neg P \vee Q\}$;
2. $C_3 : \neg P \vee Q \vee R$ *e* $C_4 : \neg Q \vee S\}$;
3. $C_5 : \neg P \vee Q$ *e* $C_6 : \neg P \vee R\}$.

Solução. Vamos analisar cada um dos pares.

1. Para o primeiro par vem que $\begin{array}{l} C_1 : P \vee R \\ C_2 : \neg P \vee Q \\ \hline \hfill C_{1;2} : R \vee Q \end{array}$

2. Para o segundo par obtém-se $\begin{array}{l} C_3 : \neg P \vee Q \vee R \\ C_4 : \neg Q \vee S \\ \hline \hfill C_{1;2} : \neg P \vee R \vee S \end{array}$

3. Para o terceiro par de cláusulas $C_5 : \neg P \vee Q$ e $C_6 : \neg P \vee R$, não existe qualquer resolvente.

□

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

Teorema 1.3.1. *Dadas duas cláusulas* C_1 *e* C_2*, uma resolvente,* C*, de* C_1 *e* C_2 *é consequência lógica de* C_1 *e* C_2*.*

Demonstração. Considerem-se 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 de literais. Suponhamos que C_1 e C_2 são verdadeiras para uma interpretação I. Pretendemos provar que então a resolvente de C_1 e C_2 também é verdadeira para I. Note-se que para I, ou L é falso ou $\neg L$ é falso. Seja L falso para I, então a cláusula $C_{1'}$ é verdadeira para I, pelo que (uma vez que $C \equiv C_{1'} \vee C_{2'}$) C é verdadeira para I. De um modo análogo se prova que se $\neg L$ é falso para I, então a cláusula $C_{2'}$ é verdadeiro para I e, consequentemente, a cláusula C é verdadeira para I. ∎

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 podemos aplicar o princípio da resolução para gerar a cláusula vaziz $\diamondsuit$ a partir de S.

Definição 1.3.2 (Dedução). *Dado um conjunto S de cláusulas uma dedução (resolução) de C obtida a partir de S, é uma sequência finita de cláusulas $C_1, C_2, \ldots C_k$ 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 da cláusula vazia $\diamondsuit$ 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 1.3.5. *Vamos aplicar o princípio da resolução ao conjunto S de cláusulas:*

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

Solução. Vamos resolver os seguinte pares de cláusulas:

de (1) e (2) obtém-se a cláusula (4) $\neg P$

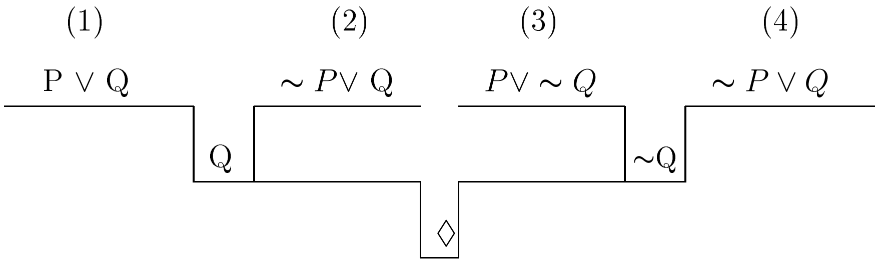
de (4) e (3) obtém-se $\diamondsuit$.

Logo, podemos concluir que S é inconsistente. □

Exemplo 1.3.6. *Vamos aplicar o princípio da resolução ao conjunto S de cláusulas:*

$$(1)\, P \vee Q \quad (2)\, \neg P \vee Q \quad (3)\, P \vee \neg Q \quad (4)\, \neg P \vee \neg Q$$

Solução. Esquematicamente, a aplicação do princípio da resolução ao conjunto S pode representar-se da seguinte forma:



Logo, S é inconsistente. □

1.3.3 Substituição e unificação na lógica de primeira ordem

Embora o princípio da resolução seja uma regra de inferência muito poderosa, uma vez que o pretendemos aplicar à lógica de primeira ordem, temos que efectuar certas modificações nas respectivas fórmulas. Para o fazermos porém, é necessário definir mais alguns conceitos, entre os quais, talvez o mais importante seja o conceito de *substituição*. Para facilitar esta tarefa, vamos introduzir a seguinte notação:

(i) $VAR = \{v : v \text{ é uma variável individual}\}$, ou seja, VAR identifica 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, $TERM$ identifica o conjunto dos termos da lógica de primeira ordem.

Nota: Observe-se que de acordo com a Definição 1.2.6,

$$(CONST \cup VAR) \subset TERM.$$

Definição 1.3.3 (Substituição)**.** *Uma substituição é uma função $\varphi_V : VAR \to TERM$ tal que, sendo $U_\varphi = \{v \in VAR : \varphi_V(v) \neq v\}$ e supondo que $U_\varphi = \{v_1, \cdots, v_n\}$, podemos descreve-la por intermédio do conjunto $\{t_1/v_1, \cdots, t_n/v_n\}$, com $t_i = \varphi_V(v_i) \neq v_i$, para $i = 1, \ldots, n$.*

O modo indicado na definição para se descrever a função φ_V, leva-nos (com algum abuso de linguagem) a escrever $\varphi_V = \{t_1/v_1, \dots, t_n/v_n\}$, com o seguinte significado:

Dada uma variável $v_i \in VAR$,

(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 ε_V e dado 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 do texto, utilizaremos letras gregas para representar substituições.

Exemplo 1.3.7. *Seguem-se dois exemplos de substituições.*

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

Definição 1.3.4 (Substituição induzida)**.** *Sendo $\theta_V = \{\theta_V(v_1)/v_1, \dots, \theta_V(v_n)/v_n\}$ uma substituição, θ_V induz a função $\theta_T : TERM \to TERM$ que se define recursivamente do seguinte modo:*

Dado um termo $t_i \in TERM$,

(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 $f(t_{i_1}, t_{i_2}, \dots, t_{i_k})$ onde f é um símbolo de função com k argumentos, então

$$\theta_T(f(t_{i_1}, \dots, t_{i_k})) = f(\theta_T(t_{i_1}), \dots, \theta_T(t_{i_k})).$$

Exemplo 1.3.8. *Considerando o termo $t = s(x, f(y, u), h(x, z))$ e a substituição*

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

vamos determinar $\varphi_T(t)$.

Solução. Aplicando sucessivamente a Definição 1.3.4, 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, identificamos uma substituição φ_V e/ou a correspondente função induzida φ_T utilizando, única e indistintamente, a respectiva letra grega que a denota, neste caso φ.

Definição 1.3.5 (Concretização de uma expressão ou conjunto de expressões)**.** *Sendo $\theta = \{\theta_V(v_1)/v_1, \ldots, \theta_V(v_n)/v_n\}$ uma substituição e E uma expressão, $E\theta$ é a expressão obtida de E aplicando a cada termo de E a função induzida pela substituição θ, ou seja, substituindo, simultaneamente, cada ocorrência da variável v_i $(1 \leq i \leq n)$ em E pelo termo $t_i = \theta_V(v_i)$. Nestas condições diz-se que $E\theta$ é uma concretização (exemplo) de E. Supondo que W denota o conjunto de expressões $\{E_1, \ldots, E_p\}$, $W\theta$ denota o conjunto $\{E_1\theta, \ldots, E_p\theta\}$.*

Exemplo 1.3.9. *Vamos aplicar a substituição $\theta_V = \{a/x, f(b)/y, c/z\}$ à expressão $E = F(x, y, g(z))$.*

Solução. Tendo presente a definição de substituição induzida θ_T, obtém-se

$$\begin{aligned}E\theta &= F(\theta_T(x), \theta_T(y), \theta_T(g(z))) \\ &= F(\theta_V(x), \theta_V(y), g(\theta_V(z))) \\ &= F(a, f(b), g(c)).\end{aligned}$$

□

Definição 1.3.6 (Composição de substituições)**.** *Sejam θ_V e λ_V duas substituições. A composição de θ_V e λ_V que se denota por $\theta_V \Delta \lambda_V$, define-se como sendo*

$$\theta_V \Delta \lambda_V = \theta_T o \lambda_V,$$

onde o símbolo $\circ$ denota a composição habitual de funções.

De acordo com esta definição, dadas as substituições θ_V e λ_V, a sua composição $\theta_V \Delta \lambda_V$ poder-se-ia descrever esquematicamente pelo seguinte diagrama:

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

Da Definição 1.3.6 decorre um método prático e muito simples para a determinação da composição de duas substituições. Com efeito, sendo

$$\theta = \{\theta_V(y_1)/y_1, \ldots, \theta_V(y_p)/y_p\} \text{ e } \lambda = \{\lambda_V(z_1)/z_1, \ldots, \lambda_V(z_p)/z_p\}$$

duas substituições. Supondo que $\{y_1, \ldots, y_p\} \cup \{z_1, \ldots, z_q\} = \{x_1, \ldots, x_n\}$, a composição de θ com λ corresponde à substituição:

$$\begin{aligned}
\theta \Delta \lambda &= \theta_T \circ \lambda_V \\
&= \{\theta_T\big(\lambda_V(x_1)\big)/x_1, \theta_T(\lambda_V(x_2))/x_2, \ldots, \theta_T(\lambda_V(x_n))/x_n\},
\end{aligned}$$

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

Exemplo 1.3.10. *Vamos determinar a composição das substituições* $\theta = \{f(y)/x, z/y\}$ *e* $\lambda = \{a/x, g(x)/y, y/z\}$.

Solução. Vamos fazer esta determinação do modo sequencial que se passa a descrever.

$$\begin{aligned}
\theta \Delta \lambda &= \theta_T o \lambda_V \\
&= \{\theta_T(\lambda_V(x))/x, \theta_T(\lambda_V(y))/y, \theta_T(\lambda_V(z))/z\} \\
&= \{\theta_T(a)/x, \theta_T(g(x))/y, \theta_T(y)/z\} = \{a/x, g(\theta_V(x))/y, \theta_V(y)/z\} \\
&= \{a/x, g(f(y))/y, z/z\} \\
&= \{a/x, g(f(y))/y\}.
\end{aligned}$$

□

Teorema 1.3.2. *A composição de substituições é associativa, ou seja, dadas três substituições φ, ψ e ζ, verifica-se que $(\varphi_V \Delta \psi_V)\Delta\zeta_V = \varphi_V \Delta(\psi_V \Delta \zeta_V)$. Adicionalmente, qualquer que seja a substituição θ_V, verifica-se que $\theta_V \Delta \varepsilon_V = \theta_V = \varepsilon_V \Delta \theta_V$, onde ε_V é a substituição vazia.*

Demonstração. Considerem-se três substituições arbitrárias φ, ψ e ζ. Então vem que

$$\begin{aligned}
(\varphi_V \Delta \psi_V)\Delta\zeta_V &= (\varphi_T \circ \psi_V)\Delta\zeta_V && \text{(dado que } \varphi_V\Delta\psi_V = \varphi_T \circ \psi_V)\\
&= \tau_V \Delta \zeta_V && \text{(fazendo } \tau_V = \varphi_T \circ \psi_V)\\
&= \tau_T \Delta \zeta_V && \text{(dado que } \tau_V\Delta\zeta_V = \tau_T \circ \zeta_V)\\
&= (\varphi_T \circ \psi_T) \circ \zeta_V && \text{(dado que } \tau_T = \varphi_T \circ \psi_T)\\
&= \varphi_T \circ (\psi_T \circ \zeta_V) && \text{(pela associatividade de } \circ)\\
&= \varphi_T \circ (\psi_V \Delta \zeta_V) && \text{(dado que } \psi_V\Delta\zeta_V = \psi_T \circ \zeta_V)\\
&= \varphi_T \circ \lambda_V && \text{(fazendo } \lambda_V = \psi_V\Delta\zeta_V)\\
&= \varphi_V \Delta \lambda_V && \text{(dado que } \varphi_T \circ \lambda_V = \varphi_V\Delta\lambda_V)\\
&= \varphi_V \Delta(\psi_V \Delta \zeta_V) && \text{(dado que } \lambda_V = \psi_V\Delta\zeta_V).
\end{aligned}$$

Para provarmos a segunda parte, vamos considerar uma substituição arbitrária θ e um termo arbitrário t.
Seja $\tau = \theta_V \Delta \varepsilon_V = \theta_T o \varepsilon_V$. Então vem que (i) $\tau_T(t) = t$ se t é uma constante ou uma variável e (ii) $\tau_T(t) = f(\tau_T(t_1), \tau_T(t_2), \ldots, \tau_T(t_n))$ se $t = f(t_1, t_2, \ldots, t_n)$.

(i) No primeiro caso, vem que $\tau_T(t) = (\theta_T o \varepsilon_V)(t) = \theta_T(\varepsilon_V(t)) = \theta_T(t)$.

(ii) No segundo caso, $\forall i \in \{1, 2, \ldots, n\}$ se t_i é uma constante ou uma variável, então, tal como anteriormente, $\tau_T(t_i) = \theta_T(t_i)$. 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 $\varepsilon_V \Delta \theta_V = \theta_V$. ■

Definição 1.3.7 (Substituição unificadora e conjunto unificável)**.** *Uma substituição θ designa-se por unificadora para o conjunto de expressões $W = \{E_1, E_2, \ldots, E_p\}$ se $W\theta = \{E\theta\}$, ou seja, se é tal que $E\theta = E_1\theta = \cdots = E_p\theta$. O conjunto W diz-se unificável se existe um unificador para ele.*

Exemplo 1.3.11. *O conjunto de 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 1.3.8 (Unificador mais geral)**.** *Um unificador σ para um conjunto de expressões $W = \{E_1, E_2, \ldots, E_p\}$ designa-se por unificador mais geral se qualquer que seja o unificador θ para W existe uma substituição λ 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 substituída por $\underline{a}$ e, consequentemente, as diferenças acabam.

Definição 1.3.9 (Conjunto de diferenças de um conjunto de expressões)**.** *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 de diferenças de W.*

Exemplo 1.3.12. *Vamos determinar um conjunto de diferenças do conjunto de expressões $W = \{P(x, \underline{f(y,z)}), P(x, \underline{a}), P(x, \underline{g(h, k(x))})\}$.*

Solução. O primeiro símbolo a partir do qual nem todos os símbolos em W são os mesmos é o terceiro. Logo, o conjunto de diferenças é $D = \{f(y, z), a, g(h, (k(x)))\}$. □

Algoritmo de unificação

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

II. Se W_k é um conjunto unitário, então `stop` (σ_k é um unificador 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 1.3.13. *Vamos determinar um unificador para o conjunto de fórmulas*

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

Solução. Aplicando o algoritmo passo a passo, obtém-se:

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

2. Uma vez que W_0 não é um conjunto unitário, σ_0 não é um unificador 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 $v_1 = x$ e $t_1 = f(a)$.

6. Seja

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

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

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

9. Seja

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

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

□

Exemplo 1.3.14. *Vamos verificar se o conjunto* $W = \{Q(f(a), g(x)), Q(y, y)\}$ *é ou não unificável.*

Solução. Aplicando o algoritmo passo a passo, obtém-se:

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 obtém-se $v_0 = y$ 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.

□

Deve-se 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 $W\sigma_0, W\sigma_1, \ldots$, 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 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.

1.3.4 Principio da resolução na lógica de primeira ordem

Uma vez introduzido o algoritmo de unificação, é agora possível aplicar o principio da resolução à lógica de primeira ordem.

Definição 1.3.10 (Factores de cláusulas)**.** *Se dois ou mais literais (com o mesmo sinal) de uma cláusula C têm o mesmo unificador mais geral σ, então $C\sigma$ designa-se por factor de C. Se $C\sigma$ é uma cláusula unitária (ou seja, com um único literal), então ela é um factor unitário de C.*

Exemplo 1.3.15. *Sendo $C : \underline{P(x)} \vee \underline{P(f(y))} \vee \neg Q(x)$, onde o primeiro e o segundo literal (sublinhados) têm o unificador mais geral $\sigma = \{f(y)/x\}$, $C\sigma = P(f(y)) \vee \neg Q(f(y))$ é um factor de C.*

Definição 1.3.11 (Resolvente binária de duas cláusulas)**.** *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 σ, 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 1.3.16. *Dadas as cláusulas $C_1 = P(x) \vee Q(x)$ e $C_2 = \neg P(a) \vee R(x)$, vamos determinar uma resolvente binária destas duas cláusulas.*

Solução. 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 $\neg L_2 \equiv P(a)$, então L_1 e $\neg L_2$ têm o unificdor mais geral $\sigma = \{a/x\}$. Logo,

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

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

Definição 1.3.12 (Resolventes de duas cláusulas)**.** *Uma resolvente das cláusulas C_1 e C_2 é uma das seguintes resolventes:*

1. *resolvente binária de C_1 e C_2;*
2. *resolvente binária de C_1 e um factor de C_2;*
3. *resolvente binária de um factor de C_1 e C_2;*
4. *resolvente binária de um factor de C_1 e um factor de C_2;*

Exemplo 1.3.17. *Sendo $C_1 = P(x) \vee P(f(y)) \vee R(g(y))$ e $C_2 = \neg P(f(g(a))) \vee Q(b)$, vamos determinar uma resolvente destas duas cláusulas.*

Solução. Dado que $C_{1'} = P(f(y)) \vee R(g(y))$ é um factor de C_1 e $R(g(g(a))) \vee Q(b)$ é uma resolvente binária de $C_{1'}$ e C_2, vem que $R(g(g(a))) \vee Q(b)$ é uma resolvente de C_1 e C_2. □

Exemplo 1.3.18. *Vamos mostrar que os ângulos alternos internos formados pela diagonal de um trapézio são iguais.*

Solução. Para provar este teorema, vamos primeiramente axiomatiza-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. Considere-se que $P(x, y, u, v)$ significa que a linha que une o segmento xy é paralela à linha que une o segmento uv e considere-se que $E(x, y, z, u, v, w)$ significa que o ângulo xyz é igual ao ângulo uvw. Então temos os seguintes axiomas:

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

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

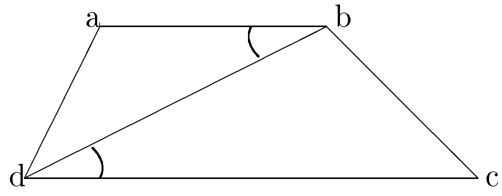


Figura 1.1: Trapézio $T(a,b,c,d)$.

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

A partir destes axiomas, estamos em condições de concluir que $E(a,b,d,c,d,b)$ é verdadeiro, ou seja, que $(A_1 \wedge A_2 \wedge A_3) \Rightarrow E(a,b,d,c,d,b)$.
Uma vez que pretendemos fazer a prova por refutação, devemos provar que a fórmula

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

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

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

(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 e, consequentemente, que a prova fica completa. $\square$

Exemplo 1.3.19. *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 primeira ordem que representem* 1., 2. *e* 3., *e verificando que a fórmula representativa de* 3. *é consequência lógica das fórmulas representativas de* 1. *e* 2.

Solução. Denotando-se por

$$\begin{array}{ll} P(x): & \text{“}x \text{ é paciente”} \\ M(x): & \text{“}x \text{ é médico”} \\ C(x): & \text{“}x \text{ é charlatão”} \\ G(x,y): & \text{“}x \text{ gosta de } y\text{”} \end{array}$$

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

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

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, $(\exists x)(P(x)\exists(\forall y)(M(y) \Rightarrow G(x,y)))$ 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. Consequentemente, 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.

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

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

é verdadeira para I.

Deste modo mostramos que se as fórmulas 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.

Uma vez que pretendemos provar que $F_1 \wedge F_2 \Rightarrow F_3$ é uma fórmula válida, vamos provar que a fórmula $F_1 \wedge F_2 \wedge \neg F_3$ 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) $C(b)$.

Finalmente, 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))

$\square$

Vamos tentar traduzir em linguagem natural a demonstração anterior.

(a) A partir de F_1, podemos assumir 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.

Exemplo 1.3.20. *Encontre as possíveis resolventes (se existirem) dos seguintes pares de cláusulas:*

1. $C_1 : \neg P(x) \vee Q(x,b)$, $C_2 : P(a) \vee Q(a,b)$

2. $D_1 : \neg P(x) \vee Q(x,x)$, $D_2 : \neg Q(a, f(a))$

Solução. Vamos analisar cada um dos pares de cláusulas.

1. $C_{1;2} : Q(a,b)$ é uma resolvente do par de cláusulas C_1 e C_2.

2. O par de cláusulas D_1 e D_2 não tem resolventes.

$\square$

Teorema 1.3.3. *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 γ 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 θ 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 λ_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 λ_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 σ o unificador mais geral para L_1 e $\neg L_2$. Então

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

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

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

e $(\lambda_T \circ \sigma_V)$ é mais geral do que $(\theta_T \circ \gamma_V)$, completando-se assim a prova deste teorema. ■

A partir deste teorema concluímos 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 mais interessados deverão consultar o capítulo 4 de [2].

1.4 Implementação do princípio da resolução no cálculo proposicional

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

$S^0 = S$;

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

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 1.4.1. *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*
(1^a)	*1*	*0*	*0*	P
(2^a)	*0*	*0*	*-1*	$\neg U$
(3^a)	*0*	*-1*	*1*	$\neg S \vee U$
(4^a)	*-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.

Solução. Como ilustração apresenta-se a resolução do Exemplo 1.4.1.

	Nž da cláusula	P	S	U
S^0	(1^a)	1	0	0
	(2^a)	0	0	-1
	(3^a)	0	-1	1
	(4^a)	-1	1	0
S^1	$(5^a) = (1^a) \oplus (4^a)$	0	1	0
	$(6^a) = (2^a) \oplus (3^a)$	0	-1	0
	$(7^a) = (3^a) \oplus (4^a)$	-1	0	1
S^2	$(8^a) = (1^a) \oplus (7^a)$	0	0	1
	$(9^a) = (2^a) \oplus (7^a)$	-1	0	0
	$(10^a) = (3^a) \oplus (5^a)$	0	0	1
	$(11^a) = (4^a) \oplus (6^a)$	-1	0	0
	$(12^a) = (5^a) \oplus (6^a)$	0	0	0

□

Exemplo 1.4.2. *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*	$\neg P \vee \neg Q \vee S$

Solução. Segue-se a respectiva resolução:

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

$\square$

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].

1.5 Exercícios.

1.1. Mostre que

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

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

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

1.2. Mostre que as fórmulas a seguir indicadas são equivalentes.

(a) $F: \quad p \Rightarrow q.$

(b) $G: \quad \neg p \vee q.$

1.3. Sendo p, q, r, s fórmulas atómicas e $F: \neg p \vee (q \wedge r) \Rightarrow \neg r$

(a) Obtenha a forma normal disjuntiva de F;

(b) Obtenha a forma normal conjuntiva de F.

1.4. Exprima por meio de expressões da lógica de primeira ordem as seguintes afirmações:

(a) Toda a gente gosta de alguém.

(b) Todo o ser vivo que não é animal é vegetal.

(c) Todos os números racionais são números reais.

(d) Existem números primos.

(e) O conjunto dos números reais é infinito.

1.5. Transforme as seguintes fórmulas na forma normal disjuntiva prenex:

(a) $(\forall x)\big(P(x) \Rightarrow (\exists y)Q(x,y)\big)$.

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

1.6. Reduza as fórmulas a seguir indicadas à forma standard de Skolem.

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

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

1.7. Mostre a inconsistência do seguinte conjunto:

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

1.8. Sendo $\Theta = \{a/x, f(z)/y, g(x)/z\}$, $E = P(h(x), z, f(z))$, determine $E\Theta$.

1.9. Averigue quais dos seguintes conjuntos são unificáveis. Se a resposta for afirmativa, obtenha o respectivo unificador mais geral.

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

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

1.10. Averigue se as seguintes cláusulas admitem um factor e, no caso afirmativo, determine-o.

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

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

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.