Lógica Computacional

Nelma Moreira
Departamento de Ciência de Computadores
Faculdade de Ciências, Universidade do Porto
email: `nam@ncc.up.pt`

Versão: 2010

Conteúdo

Capítulo 1

Lógica proposicional

A lógica proposicional remonta a Aristóteles, e teve como objectivo modelizar o raciocínio humano. Partindo de frases declarativas (*proposições*), que podem ser *verdadeiras* (**V**) ou *falsas* (**F**) estuda-se o processo de construção e a veracidade de outras proposições usando conectivas como *ou* ($\vee$), *e* ($\wedge$), *não* ($\neg$), *se... então...* ($\rightarrow$). Se p e q são proposições, $p \wedge q$ é uma proposição *verdadeira* se p e q o forem, e é uma proposição *falsa*, caso contrário; $p \vee q$ é uma proposição *verdadeira* se p ou q o forem, e *falsa*, caso contrário;$\neg p$ é uma proposição verdadeira se p for *falsa*, e *falsa* se p for *verdadeira*.
Considera as seguintes frases declarativas (com o respectivo valor de verdade):

- *Os gorilas são mamíferos* **V**
- *O Porto é uma cidade* **V**
- $2+3=6$ **F**
- $3 \in \mathbb{N}$ **V**
- $3 > 7$ **F**
- *Um quadrado tem 6 lados* **F**

Então, podemos concluir que:

- *Os gorilas são mamíferos* e *O Porto é uma cidade* **V**

  porque é uma conjunção de proposições **V**

- $2+3=6$ ou $3 \in \mathbb{N}$ **V**

  porque é uma disjunção de proposições das quais uma é **V**

- não $3 > 7$ **V**

  porque é uma negação de uma proposição **F**

- Se $7 > 3$ então $3 + 3 = 6$ **V**

  porque uma implicação é **V** sse o consequente é **V** sempre que o antecedente é **V**

- Se $3 > 7$ então $2 + 3 = 6$ **V**

  porque é uma implicação cujo antecedente é **F**.

Cada proposição vai ser representada por uma *variável proposicional* (p, q, s, t, p_1, ...) e as conectivas lógicas por *símbolos n-ários*. Em particular temos:

Conectiva	Símbolos	Aridade	Outros símbolos equivalentes
Conjunção	$\wedge$	2	&, &&, $\cdot$
Disjunção	$\vee$	2	$\|$, $+$
Negação	$\neg$	1	$\sim$, $^{-}$, !
Implicação	$\rightarrow$	2	$\Rightarrow$, $\supset$

1.1 Linguagens da lógica proposicional

Uma *linguagem da lógica proposicional* é formada a partir dos seguintes conjuntos de símbolos primitivos:

- um conjunto numerável de variáveis proposicionais

  $\mathcal{V}_{Prop} = \{p, q, r, \ldots, p_1, \ldots\}$

- conectivas lógicas $\wedge$, $\vee$, $\neg$ e $\rightarrow$

- os parêntesis (e)

Definição 1.1. *Uma fórmula* bem-formada *(ϕ,ψ,θ,...) é definida indutivamente pelas seguintes regras:*

i) uma variável proposicional p é uma fórmula

ii) se ϕ é uma fórmula então $(\neg\phi)$ é uma fórmula

iii) se ϕ e ψ são fórmulas então $(\phi \wedge \psi)$, $(\phi \vee \psi)$ e $(\phi \rightarrow \psi)$ são fórmulas

Exemplos de fórmulas são:

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

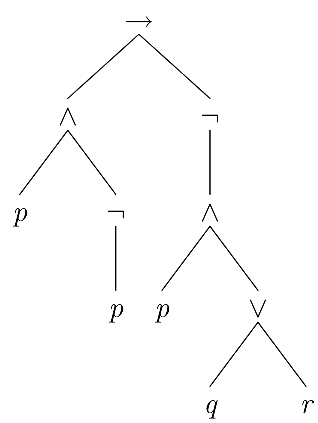
- $(((\neg p) \wedge q) \rightarrow (p \wedge (q \vee (\neg r))))$

O conjunto das fórmulas da lógica proposicional pode também ser descrito pqela seguinte gramática independente de contexto, em notação BNF:

$$\phi := p \ \mid (\neg\phi) \ \mid (\phi \ \wedge \ \phi) \ \mid (\phi \ \vee \ \phi) \ \mid (\phi \rightarrow \phi)$$

uma gramática independente de contexto que gere a linguagem das fórmulas da lógica proporcional, onde $p \in \mathcal{V}_{Prop}$. Usando a gramática podemos associar a cada fórmula uma árvore sintáctica.

Exemplo 1.1. *Omitindo os parêntesis, temos a seguinte árvore pra a fórmula* $((p \wedge (\neg p)) \rightarrow \neg(p \wedge (q \vee r)))$*:*



Para legibilidade das fórmulas, os parêntesis podem-se omitir, considerando as seguintes regras:

- os parêntesis exteriores são omitidos
- $\neg$ tem precedência sobre $\wedge$
- $\wedge$ tem precedência sobre $\vee$
- $\vee$ tem precedência sobre $\rightarrow$
- $\wedge$ e $\vee$ são associativas à esquerda
- $\rightarrow$ é associativa à direita

Por exemplo, $\phi \wedge \psi \vee \theta$ é uma abreviatura de $((\phi \wedge \psi) \vee \theta)$ e $\psi \wedge \phi \wedge \theta$ corresponde a $((\psi \wedge \phi) \wedge \theta)$

Definição 1.2. *Uma* sub-fórmula imediata *é definida indutivamente pelas seguintes regras:*

1. *uma variável proposicional não tem sub-fórmulas imediatas;*

2. $(\neg\phi)$ *tem* ϕ *como sub-fórmula imediata;*

3. *as fórmulas* $(\phi \wedge \psi)$*,* $(\phi \vee \psi)$ *e* $(\phi \rightarrow \psi)$ *têm* ϕ *e* ψ *como sub-fórmulas imediatas.*

Definição 1.3. *Uma fórmula ϕ é uma* sub-fórmula *duma fórmula ψ se e só se:*

1. *ϕ é uma sub-fórmula imediata de ψ*
2. *existe uma fórmula θ tal que ϕ é uma sub-fórmula de θ e θ é uma sub-fórmula de ψ*

Dada uma árvore sintáctica duma fórmula, cada nó define uma sua sub fórmula.

Exemplo 1.2. *Para o exemplo 1.1 as sub-fórmulas são:*$p, q, \neg p, r, p \wedge \neg p, q \vee r, p \wedge (q \vee r)$*e*$\neg(p \wedge (q \vee r))$.

Exercício 1.1. *Constrói a árvore sintáctica (omitindo parêntisis) e determina quais as sub-fórmulas e quais as sub-fórmulas imediatas de:*
$((\neg p \wedge q) \rightarrow (p \wedge (q \vee \neg r)))$
$\diamond$

1.2 Semântica da lógica proposicional

Os valores de verdade são **V** e **F**, onde **V** representa o valor lógico *verdadeiro* e **F**, *falso*.

Definição 1.4. *Uma* atribuição de valores de verdade *(ou valorização) é uma função $v : \mathcal{V}_{Prop} \longrightarrow \{\boldsymbol{V}, \boldsymbol{F}\}$ que atribuí um valor de verdade a cada variável proposicional. Uma valorização v pode ser estendida ao conjunto das fórmulas:*

i. *para $p \in \mathcal{V}_{Prop}$ $v(p)$ já está definido*

ii.

$$
\begin{aligned}
v(\neg\phi) &= \boldsymbol{V} \quad \textit{se} \quad v(\phi) = \boldsymbol{F} \\
v(\neg\phi) &= \boldsymbol{F} \quad \textit{se} \quad v(\phi) = \boldsymbol{V}
\end{aligned}
$$

iii.

$$
\begin{aligned}
v(\phi \wedge \psi) &= \boldsymbol{V} \quad \textit{se} \quad v(\phi) = \boldsymbol{V} \textit{ e } v(\psi) = \boldsymbol{V} \\
v(\phi \wedge \psi) &= \boldsymbol{F} \qquad\quad \textit{caso contrário}
\end{aligned}
$$

iv.

$$
\begin{aligned}
v(\phi \vee \psi) &= \boldsymbol{V} \quad \textit{se} \quad v(\phi) = \boldsymbol{V} \textit{ ou } v(\psi) = \boldsymbol{V} \\
v(\phi \vee \psi) &= \boldsymbol{F} \qquad\quad \textit{caso contrário}
\end{aligned}
$$

v.

$$v(\phi \to \psi) = \boldsymbol{F} \quad se \quad v(\phi) = \boldsymbol{V} \text{ e } v(\psi) = \boldsymbol{F}$$
$$v(\phi \to \psi) = \boldsymbol{V} \quad\quad caso\ contrário$$

Podemos resumir usando as seguintes *tabelas de verdade*:

ϕ	$\neg\,\phi$
F	**V**
V	**F**

ϕ	ψ	$\phi \land\ \psi$
F	**F**	**F**
F	**V**	**F**
V	**F**	**F**
V	**V**	**V**

ϕ	ψ	$\phi \lor \psi$
F	**F**	**F**
F	**V**	**V**
V	**F**	**V**
V	**V**	**V**

ϕ	ψ	$\phi{\to}\psi$
F	**F**	**V**
F	**V**	**V**
V	**F**	**F**
V	**V**	**V**

Dada uma fórmula ϕ, o valor de $v(\phi)$ pode assim ser recursivamente calculdado a partir dos valores atribuídos às variáveis da fórmula ϕ.
Por exemplo se $v(p) = \mathbf{V}$, $v(q) = \mathbf{F}$ e $v(r) = \mathbf{V}$, podemos calcular o valor de $v((p\ \land\ q)\ \lor\ \neg r)$:

p	q	r	(	p	∧	q	)	∨	¬	r
V	**F**	**V**			**F**			**F**	**F**	

Uma fórmula ϕ com n variáveis proposicionais tem 2^n valorizações. Porquê?
Podemos construir uma tabela em que cada linha corresponde a uma delas. Seja ϕ a fórmula $(p\ \land\ q)\ \lor\ \neg r$, temos as seguintes valorizações:

p	q	r	(	p	∧	q	)	∨	¬	r
V	**V**	**V**			**V**			**V**	**F**	
V	**V**	**F**			**V**			**V**	**V**	
V	**F**	**V**			**F**			**F**	**F**	
V	**F**	**F**			**F**			**V**	**V**	
F	**V**	**V**			**F**			**F**	**F**	
F	**V**	**F**			**F**			**V**	**V**	
F	**F**	**V**			**F**			**F**	**F**	
F	**F**	**F**			**F**			**V**	**V**	

1.2.1 Satisfazibilidade, validade e consequência

Definição 1.5. *Uma fórmula ϕ é*

satisfazível *se existe uma valorização v tal que $v(\phi) = \boldsymbol{V}$. Escreve-se $\models_v \phi$ e diz-se que v satisfaz ϕ*

uma tautologia *se para todas as valorizações v, $v(\phi) = \boldsymbol{V}$ e escreve-se $\models \phi$. Também se diz que ϕ é* válida. *Ex: $\models p \vee \neg p$ (Terceiro excluído)*

uma contradição *se para todas as valorizações v, $v(\phi) = \boldsymbol{F}$. Escreve-se $\not\models \phi$. Ex: $\not\models p \wedge \neg p$.*

Proposição 1.1. *Uma fórmula ϕ é uma tautologia se e só se $\neg\phi$ é uma contradição. Uma fórmula ϕ é satisfazível se e só se $\neg\phi$ não é uma tautologia. Uma fórmula é* não-satisfazível *se e só se é uma contradição.*

Demonstração. Seja v uma atribuição de valores de verdade. Por definição, $v(\phi) = \mathbf{V}$ se e só se $v(\neg\phi) = \mathbf{F}$. Se ϕ é verdade para todas as valorizações então $\neg\phi$ é falsa para todas elas. Do mesmo modo se concluem a segunda e terceira afirmação. □

Definição 1.6. *Seja Γ um conjunto de fórmulas. Uma valorização v* satisfaz Γ *se e só se v satisfaz toda a fórmula $\psi \in \Gamma$. O conjunto Γ é* satisfazível *se existe uma valorização que o satisfaz. Uma fórmula ϕ é uma* consequência semântica *de Γ, se para toda a valorização v que satisfaz Γ, se tem $v(\phi) = \boldsymbol{V}$; e escreve-se $\Gamma \models \phi$.*

Se $\Gamma = \emptyset$, então $\emptyset \models \phi$ é equivalente a $\models \phi$. Nota que $\models \phi$ se e só se ϕ é uma tautologia. Se $\Gamma = \{\psi\}$ e $\Gamma \models \phi$ então diz-s que ϕ é *consequência semântica* de ψ (e escreve-se $\psi \models \phi$).

Todos estes conceitos podem ser avaliados usando tabelas de verdade. Uma fórmula ϕ é *satisfazível*, se houver uma linha da tabela para a qual o seu valor é $\mathbf{V}$; uma *tautologia* se para todas as linhas o seu valor é $\mathbf{V}$; uma *contradição* se para nenhuma linha o seu valor é $\mathbf{V}$. A fórmula ϕ é *consequência semântica* de um conjunto finito de fórmulas $\{\psi_1, \ldots, \psi_n\}$ se para todas as linhas que todos os ψ_i são $\mathbf{V}$, então ϕ também é $\mathbf{V}$.

Se $\psi \models \phi$ e $\phi \models \psi$ então ψ e ϕ são *semânticamente equivalentes*(i.e as suas tabelas de verdade são iguais), e escreve-se $\psi \equiv \phi$.

Alguns exemplos de fórmulas semânticamente equivalentes:

$$
\begin{array}{rcll}
\phi \wedge \psi & \equiv & \psi \wedge \phi & \text{(comutatividade)} \\
\phi \vee \psi & \equiv & \psi \vee \phi & \text{(comutatividade)} \\
\neg(\phi \wedge \psi) & \equiv & (\neg\phi \vee \neg\psi) & \text{(Lei de de DeMorgan)} \\
\neg(\phi \vee \psi) & \equiv & (\neg\phi \wedge \neg\psi) & \text{(Lei de DeMorgan)} \\
(\phi \wedge \psi) \wedge \theta & \equiv & \phi \wedge (\psi \wedge \theta) & \text{(associatividade)} \\
(\phi \vee \psi) \vee \theta & \equiv & \phi \vee (\psi \vee \theta) & \text{(associatividade)} \\
(\phi \vee \phi) & \equiv & \phi & \text{(idempotência)} \\
(\phi \wedge \phi) & \equiv & \phi & \text{(idempotência)} \\
(\phi \wedge \psi) \vee \theta & \equiv & (\phi \vee \theta) \wedge (\psi \vee \theta) & \text{(distributividade)} \\
(\phi \vee \psi) \wedge \theta & \equiv & (\phi \wedge \theta) \vee (\psi \wedge \theta) & \text{(distributividade)} \\
\neg\neg\phi & \equiv & \phi & \text{(Dupla negação)} \\
\phi \to \psi & \equiv & \neg\phi \vee \psi &
\end{array}
$$

Exercício 1.2. *Verifica as equivalências anteriores.* $\diamond$

Exercício 1.3. *Justifica a veracidade ou falsidade de cada uma das afirmações seguintes, onde Γ e Σ representam conjuntos de fórmulas e $\phi, \psi, \theta, \gamma$ representam fórmulas da lógica proposicional:*

1. *Se $\phi, \psi \models \theta \to \gamma$ e $\phi \models \theta$ então $\phi, \psi \models \gamma$;*

2. *Se $\Gamma \models \theta$ e $\Gamma \subseteq \Sigma$, então $\Sigma \models \theta$;*

3. *Se $\Gamma \models \theta$ e $\Sigma \models \theta$, então $\Gamma \cap \Sigma \models \theta$;*

4. *Se $\Gamma \cup \{\phi\}$ é satisfazível então $\Gamma \cup \{\neg\phi\}$ não é satisfazível;*

5. *$\phi \models \psi \to \theta$ e $\gamma \models \psi$ se e só se $\phi, \gamma \models \theta$;*

6. *Se Σ é um conjunto de fórmulas satisfazível, então $\Sigma \cup \{\phi\}$ é satisfazível ou $\Sigma \cup \{\neg\phi\}$ é satisfazível qualquer que seja ϕ.*

7. *Se $\Sigma \models \phi$ e $\Gamma \models \phi$ então $\Sigma \cup \Gamma \models \phi$.*

8. *Se Σ é satisfazível então existe uma fórmula ϕ tal que $\Sigma \not\models \phi$.*

$\diamond$

1.2.2 Funções de verdade

A tabela de verdade duma fórmula ϕ, com $n > 0$ variáveis proposicionais $p_1, \ldots p_n$, define uma função de verdade

$$F_\phi : \{\mathbf{V}, \mathbf{F}\}^n \longrightarrow \{\mathbf{V}, \mathbf{F}\}$$

tal que $F_\phi(x_1, \ldots, x_n) = v_X(\phi)$, onde v_X é uma valorização tal que $v_X(p_i) = x_i$ para $i \in \{1 \ldots n\}$ e $x_i \in \{\mathbf{V}, \mathbf{F}\}$.

Qualquer função de $f : \{\mathbf{V}, \mathbf{F}\}^n \longrightarrow \{\mathbf{V}, \mathbf{F}\}$, com $n > 0$ diz-se uma *função de verdade* ou função booleana.

Exercício 1.4. *Existem 4 funções de verdade e aridade 1 e 16 funções de verdade de aridade 2. Constrói as tabelas de verdade correspondentes. E quantas funções existem de aridade n, para $n > 0$?* $\diamond$

Definição 1.7. *Um conjunto de conectivas C diz-se* completo *se para qualquer função de verdade f, existe uma fórmula ϕ com n variáveis proposicionais e contendo apenas conectivas de C, tal que $F_\phi = f$.*

Proposição 1.2. *O conjunto de conectivas $\{\wedge, \vee, \neg\}$ é completo.*

Demonstração. Mostramos por indução sobre n.

Base. Para $n = 1$ existem 4 funções de verdade:

x_1	f_1
V	**F**
F	**F**

x_1	f_2
V	**F**
F	**V**

x_1	f_3
V	**V**
F	**F**

x_1	f_4
V	**V**
F	**V**

Sendo $\phi_1 = p \wedge \neg p$, $\phi_2 = \neg p$, $\phi_3 = p$, $\phi_4 = p \vee \neg p$, tem-se que $F_{\phi_i} = f_i$ para $1 \leq i \leq 4$.

Indução. Supondo que a hipótese é válida para n, seja $f : \{\mathbf{V}, \mathbf{F}\}^{n+1} \longrightarrow \{\mathbf{V}, \mathbf{F}\}$. Construímos duas funções n-árias f_1 e f_2 tal que:

$f_1(x_1, \ldots, x_n) = f(x_1, \ldots, x_n, \mathbf{V})$

$f_2(x_1, \ldots, x_n) = f(x_1, \ldots, x_n, \mathbf{F})$.

Por hipótese de indução existem ϕ_i, com variáveis $p_1, \ldots, p_n$, tal que $F_{\phi_i} = f_i$ para $i = 1, 2$. Tome-se $\phi = (p_{n+1} \wedge \phi_1) \vee (\neg p_{n+1} \wedge \phi_2)$, então $F_\phi = f$.

$\square$

Proposição 1.3. *O conjunto de conectivas $\{\neg, \rightarrow\}$ é completo.*

Demonstração. Basta ver que $\phi \wedge \psi \equiv \neg(\phi \rightarrow \neg\psi)$ e $\phi \vee \psi \equiv (\neg\phi \rightarrow \psi)$. $\square$

Exercício 1.5. *Mostra que o conjunto de conectivas $\{\neg, \vee\}$ é completo.* $\diamond$

1.2.3 Uso de conjuntos completos de conectivas

Podemos restringuir-nos só a conjuntos completos de conectivas. Em particular, podiamos ter considerado na definição da linguagem da lógica proposicional, apenas ou:

- as conectivas $\wedge$, $\vee$ e $\neg$
- as conectivas $\rightarrow$ e $\neg$
- ...

E considerar as restantes abreviaturas.

Uma das vantagens de ter um número menor de tipos de fórmulas é o de facilitar as demonstrações... Mas também podemos definir outras conectivas. Por exemplo uma para cada uma das funções de verdade unárias ou binárias... As mais usuais são:

Designação	Conectiva	Fórmula semanticamente equivalente
Falso	**F**	$\phi \wedge \neg\phi$
Verdade	**V**	$\phi \vee \neg\phi$
Implicação	$\phi \rightarrow \psi$	$\neg\phi \vee \psi$
Equivalência	$\phi \leftrightarrow \psi$	$(\neg\phi \vee \psi) \wedge (\phi \vee \neg\psi)$
Ou Exclusivo	$\phi \dot{\vee} \psi$	$(\phi \wedge \neg\psi) \vee (\neg\phi \wedge \psi)$
Não-e	$\phi \tilde{\wedge} \psi$	$\neg(\phi \wedge \psi)$
Não-ou	$\phi \tilde{\vee} \psi$	$\neg(\phi \vee \psi)$

Exercício 1.6. *Constrói as tabelas de verdade associadas a cada uma das conectivas.* $\diamond$

Exercício 1.7. *Mostra que*

a) $\phi \leftrightarrow \psi \equiv (\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$

b) $\phi \rightarrow \psi \equiv \neg\psi \rightarrow \neg\phi$ *(contrapositivo)*

c) $\models \phi \rightarrow \psi$ *se e só se* $\phi \models \psi$

d) $\models \phi \leftrightarrow \psi$ *se e só se* $\phi \equiv \psi$

$\diamond$

Leituras suplementares [BE00] (Cap. 1, 3, 7) [HR00] (Cap. 1.3, 1.4.1)

1.3 Formas normais

Vamos ver que podemos transformar fórmulas em fórmulas semânticamente equivalentes, de modo a obter fórmulas de *formas especiais* e que nos permitam decidir mais fácilmente sobre a satisfabilidade ou validade das fórmulas originais. Algumas dessas *formas normais* existem para qualquer fórmula, outras apenas para certas classes de fórmulas.

1.3.1 Forma normal negativa

Um *literal* é uma variável proposicional (p) ou a sua negação ($\neg p$). Uma fórmula diz-se em forma normal negativa se a ocorrência de $\neg$ for só em literais.

Proposição 1.4. *Qualquer fórmula contendo apenas as conectivas* $\wedge$ *,* $\vee$ *e* $\neg$ *é semanticamente equivalente a uma fórmula em forma normal negativa.*

Demonstração. Basta usar as Leis de DeMorgan e eliminar as duplas negações. □

Exemplo 1.3. *Determinar a forma normal negativa de* $\neg((p \vee q) \wedge \neg p)$*.*

$$\begin{array}{rl}
\neg((p \vee q) \wedge \neg p) & \\
\neg(p \vee q) \vee \neg\neg p & \textit{(DeMorgan)} \\
(\neg p \wedge \neg q) \vee \neg\neg p & \textit{(DeMorgan)} \\
(\neg p \wedge \neg q) \vee p & \textit{(Dupla Negação)}
\end{array}$$

Exercício 1.8. *Determina a forma normal negativa de*

$$(p \leftrightarrow q) \leftrightarrow r) \wedge \neg(p \leftrightarrow (q \leftrightarrow r))$$

⋄

1.3.2 Forma normal disjuntiva

Uma fórmula diz-se em *forma normal disjuntiva* se for da forma:

$$(\alpha_{11} \wedge \ldots \wedge \alpha_{1k_1}) \vee \ldots \vee (\alpha_{n1} \wedge \ldots \wedge \alpha_{nk_n})$$

onde cada α_{ij} é um literal.

Lema 1.1. *Para qualquer função de verdade* $f : \{\boldsymbol{V}, \boldsymbol{F}\}^n \longrightarrow \{\boldsymbol{V}, \boldsymbol{F}\}$*, existe uma fórmula* ϕ *com* n *variáveis proposicionais e em forma normal disjuntiva, tal que* $F_\phi = f$*.*

Demonstração. Se $f = \mathbf{F}$, para todos os valores dos argumentos, então $\phi = p_1 \wedge \neg p_1$. Senão, para cada valorização v (correspondente a uma linha da tabela de verdade de f) seja:

$$\phi_v = l_1^v \wedge \ldots \wedge l_n^v$$

onde

$$l_i^v = \begin{cases} p_i & \text{se } v(p_i) = \mathbf{V} \\ \neg p_i & \text{se } v(p_i) = \mathbf{F} \end{cases}$$

Nota que $v(\phi_v) = \mathbf{V}$. Então, basta considerar

$$\phi = \bigvee_{f(v(p_1),\ldots,v(p_n))=\mathbf{V}} \phi_v$$

□

Para a seguinte função de verdade:

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
V	**V**	**V**	**V**
V	**V**	**F**	**V**
V	**F**	**V**	**V**
V	**F**	**F**	**F**
F	**V**	**V**	**V**
F	**V**	**F**	**F**
F	**F**	**V**	**F**
F	**F**	**F**	**F**

uma fórmula em forma normal disjuntiva é:

$$(p_1 \wedge p_2 \wedge p_3) \vee (p_1 \wedge p_2 \wedge \neg p_3) \vee (p_1 \wedge \neg p_2 \wedge p_3) \vee (\neg p_1 \wedge p_2 \wedge p_3)$$

Corolário 1.1. *Qualquer fórmula é semanticamente equivalente a uma fórmula em forma normal disjuntiva.*

Exercício 1.9. *Demonstra o Corolário 1.1.* ◇

Resolução 1.3.1. *Dada uma fórmula, é possível transformá-la numa semanticamente equivalente em forma normal disjuntiva, considerando os seguintes passos:*

1. *obter uma fórmula apenas com as conectivas* $\wedge$ *,* $\vee$ *e* $\neg$
2. *obter uma fórmula em forma normal negativa*

3. *aplicar a distributividade:* $(\phi \lor \psi) \land \theta \equiv (\phi \land \theta) \lor (\psi \land \theta)$

   $\theta \land (\phi \lor \psi) \equiv (\theta \land \phi) \lor (\theta \land \psi)$

Exercício 1.10. *Determina uma forma normal disjuntiva para*

$$(p \lor r) \leftrightarrow (q \land \neg p)$$

$\diamond$

Resolução 1.3.2 (1.10)**.**

$$\begin{aligned}
(p \lor r) \leftrightarrow (q \land \neg p) &\Leftrightarrow ((p \lor r) \to (q \land \neg p)) \land ((q \land \neg p) \to (p \lor r)) \\
&\Leftrightarrow (\neg(p \lor r) \lor (q \land \neg p)) \land (\neg(q \land \neg p) \lor (p \lor r)) \\
&\Leftrightarrow ((\neg p \land \neg r) \lor (q \land \neg p)) \land ((\neg q \lor p) \lor p \lor r) \\
&\Leftrightarrow (((\neg p \land \neg r) \lor (q \land \neg p)) \land (\neg q \lor p)) \lor (((\neg p \land \neg r) \\
&\lor (q \land \neg p)) \land (p \lor r)) \\
&\Leftrightarrow (\neg p \land \neg r \land (\neg q \lor p)) \lor (q \land \neg p \land (\neg q \lor p)) \\
&\lor (\neg p \land \neg r \land (p \lor r)) \lor (q \land \neg p \land (p \lor r)) \\
&\Leftrightarrow (\neg p \land \neg r \land \neg q) \lor (\neg p \land \neg r \land p) \lor (q \land \neg p \land \neg q) \lor (q \land \neg p \land p) \\
&\lor (\neg p \land \neg r \land p) \lor (\neg p \land \neg r \land r) \lor (q \land \neg p \land p) \lor (q \land \neg p \land r) \\
&\Leftrightarrow (\neg p \land \neg r \land \neg q) \lor (q \land \neg p \land r)
\end{aligned}$$

No último passo eliminamos conjunções que não eram satisfazíveis.

Lema 1.2. *Uma conjunção de literais* $l_1 \land \ldots \land l_n$ *é satisfazível se e só se para todo o* $1 \leq i, j \leq n$, l_i *não é* $\neg l_j$.

Corolário 1.2. *Uma fórmula* ϕ *em forma normal disjuntiva é satisfazível se e só se alguma das suas conjunções de literais o for.*

Obtemos assim um método de determinar se uma fórmula é satisfazível. Em particular, se a fórmula já estiver em forma normal disjuntiva o método é linear no seu tamanho.

Exercício 1.11. *Explicita este método.* $\diamond$

1.3.3 Forma normal conjuntiva

Uma fórmula diz-se em *forma normal conjuntiva* se for da forma:

$$(\alpha_{11} \lor \ldots \lor \alpha_{1k_1}) \land \ldots \land (\alpha_{n1} \lor \ldots \lor \alpha_{nk_n})$$

onde cada α_{ij} é um literal.
Por dualidade, temos

Lema 1.3. *Uma disjunção de literais* $l_1 \lor \ldots \lor l_n$ *é uma tautologia se e só se para algum* $1 \leq i, j \leq n$, l_i *é* $\neg l_j$.

Então, é fácil determinar se uma fórmula em forma normal conjuntiva é uma tautologia: basta verificar se todas as disjunções são tautologias, pelo método dado no Lema 1.3.
Mas como obter uma fórmula em forma normal conjuntiva?

1. se tivermos a tabela de verdade, por um método dual ao da forma normal disjuntiva: isto é, escolher as linhas que correspondem a **F**, considerar para cada uma a disjunção de literais tal que se $x_i = \mathbf{V}$ coloca-se $\neg p_i$ e se $x_i = \mathbf{F}$ coloca-se p_i; e finalmente tomar a conjunção dessas disjunções (Verifica a correção!).
2. se tivermos uma fórmula, adaptar o método dado para a forma normal disjuntiva, usando a distributividade para a conjunção...

Exercício 1.12. *Obtém uma fórmula em forma normal conjuntiva correspondente à tabela de verdade dada anteriormente.* $\diamond$

Resolução 1.3.3. *Uma fórmula é:*

$$(\neg p_1 \lor p_2 \lor p_3) \land (p_1 \lor \neg p_2 \lor p_3) \land (p_1 \lor p_2 \lor \neg p_3) \land (p_1 \lor p_2 \lor p_3)$$

Exercício 1.13. *Determina uma forma normal conjuntiva equivalente à fórmula*

$$(p \lor r) \leftrightarrow (q \land \neg p)$$

$\diamond$

1.3.4 Fórmulas de Horn e Satisfazibilidade

Uma *fórmula de Horn* da lógica proposicional é uma fórmula em forma normal conjuntiva em que em cada disjunção existe no máximo um literal positivo. Exemplos de fórmulas de Horn são:

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

Numa fórmula de Horn, as disjunções $\neg p_1 \lor \ldots \lor \neg p_n \lor p$ também se podem escrever como

$$(p_1 \land \ldots \land p_n) \to p$$

ou se p não existe (ou é **F**):

$$(p_1 \land \ldots \land p_n) \to \mathbf{F}$$

ou se os p_i não existem)

$$\mathbf{F} \rightarrow p$$

Nota que nem todas as fórmulas têm uma fórmula de Horn equivalente...basta que as suas formas normais conjuntivas tenham mais que um literal positivo que não possa ser simplificado...

Para determinar se uma fórmula de Horn da lógica proposicional é **satisfazível** podemos usar um algoritmo (mais eficiente que a construção da tabela de verdade correspondente e a verificação se para alguma linha a fórmula tem o valor **V**).

Vamos ilustrar o algoritmo com a fórmula $p \land \neg q \land (q \lor \neg p)$:

- começar por colocar numa linha as variáveis proposicionais que ocorrem na fórmula e colocar a fórmula. Ex:

p	q	$p \land \neg q \land (q \lor \neg p)$

- se alguma das variáveis proposicionais é um dos elementos da conjunção atribuir o valor **V** a essa variável (porquê?). Ex:

p	q	$p \land \neg q \land (q \lor \neg p)$
V		

- Com a informação dessas variáveis preencher a tabela como se tivesse a construir a tabela de verdade (para essa linha), analisando cada disjunção para determinar se, para ela ser verdadeira, se pode determinar mais valores para as variáveis proposicionais:

p	q	p	$\land$	$\neg$	q	$\land$	$(q$	$\lor$	$\neg$	$p)$
V									**F**	

  Neste caso, q tem de ser **V** e então isso pode ser acrescentado:

p	q	p	$\land$	$\neg$	q	$\land$	$(q$	$\lor$	$\neg$	$p)$
V	**V**								**F**	

  E voltando a repetir este passo, obtém-se:

p	q	p	$\land$	$\neg$	q	$\land$	$(q$	$\lor$	$\neg$	$p)$
V	**V**			**F**					**F**	

  Continuar até mais nada poder ser acrescentado.

- se no passo anterior se atribuir **F** a um dos elementos da conjunção, a fórmula também fica com o valor **F** e não é satisfazível. Caso contrário podemos atribuir à fórmula o valor **V** se atribuirmos **F** às restantes variáveis proposicionais. Nota que assim no máximo um literal por disjunção é F (e as disjunções não são unitárias).

  No exemplo que estamos a considerar, a fórmula tem o valor **F** e portanto não é satisfazível.

Exercício 1.14. *1. Justifica a correção do algoritmo, isto é, que atribui o valor verdade se e só se a fórmula de Horn é satisfazível.*

2. Aplica o algoritmo às seguintes fórmulas:

- $(\neg p \lor \neg q) \land (\neg q \lor r) \land q$
- $p \land (\neg p \lor q) \land (\neg q \lor p)$
- $\neg p \land (\neg p \lor q) \land \neg q$
- $p \land (\neg p \lor q) \land \neg r$

◇

1.3.5 Satisfazibilidade

O problema de determinar se uma fórmula da lógica proposicional é satisfazível é um problema com muitas aplicações em Ciência de Computadores uma vez que muitos problemas de combinatória e optimização se podem reduzir a ele (p.e. resolução de puzzles, como o sudoku, etc). Já vimos que este problema se pode resolver através da construção de tabelas de verdade, mas no pior caso isso pode ser exponencial no número de variáveis. Na realidade, não se conhece nenhum algoritmo mais eficiente para resolver este problema para qualquer tipo de fórmulas. Contudo, existem classes de fórmulas para as quais o problema é polinomial (linear): por exemplo se a fórmula estiver em forma normal disjuntiva ou for uma fórmula de Horn. Por outro lado, no caso geral existem diversos algoritmos que na prática se comportam muito melhor que o da construção de tabelas de verdade. Estes algoritmos podem ser aplicados a fórmulas genéricas (com qualquer tipo de conectiva) mas são especialmente simples se as fórmulas estiverem em forma normal conjuntiva (FNC). Este tipo de fórmulas pode ser representado de forma compacta usando a noção de *cláusula*.

1.3.5.1 Cláusulas

Definição 1.8. *Uma* cláusula *é uma disjunção de literais $l_1 \lor l_2 \lor \ldots \lor l_n$, $n \geq 0$. Uma cláusula pode-se representar por um conjunto de literais. Se $n = 0$ dizemos que a cláusula é* vazia *e corresponde a $\boldsymbol{F}$. Se $n = 1$ dizemos que a cláusula é* unitária.

Por exemplo $p \vee \neg q \vee \neg p \vee s$ é uma cláusula e pode representar-se por um conjunto $\{p, \neg q, \neg p, s\}$.
Qualquer fórmula da lógica proposicional em FNC pode-se representar por um conjunto de cláusulas. Por exemplo,

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

pode ser vista como um conjunto de cláusulas:

$$\{\{\neg p\}, \{q, r\}, \{\neg r, \neg s\}, \{p, s\}, \{\neg q, \neg s\}\}$$

E como qualquer fórmula é equivalente a uma em FNC, se tivermos um método para determinar a satisfazibilidade de cláusulas, temos um método que se pode aplicar a qualquer fórmula.

Definição 1.9. *Dado um literal l designamos por* literal complementar *o literal $\tilde{l}$ definido por:*

$$\tilde{l} = \begin{cases} \neg l, & \text{se } l \text{ é uma variável (positivo)} \\ p, & \text{se } l \text{ é da forma } \neg p \text{ (negativo)} \end{cases}$$

Como vimos no Lema 1.3, uma cláusula é uma tautologia se contém um par de literais complementares p e $\neg p$. Estas cláusulas podem ser retiradas do conjunto sem alterar a satisfazibilidade.

1.3.5.2 O algoritmo de Davis-Putnam

O algoritmo que vamos apresentar é uma variante do original de 1960 [DP60, DLL62] e que continua a ser a base de muitos dos algoritmos que actualmente são mais competitivos (estando as diferenças nas estruturas de dados e nas diversas heurísticas que usam).
A ideia base do algoritmo é considerar para cada variável os possíveis valores de verdade e simplificar a fórmula de acordo com essas atribuições até se poder concluir que ela é ou não satisfazível. As simplificações incluem um caso especial para as cláusulas unitárias. Por uma questão de legibilidade vamos continuar a representação de cada cláusula usando disjunções (em vez de um conjunto de literais).

Definição 1.10. *Seja S um conjunto de cláusulas. Um conjunto S' é obtido de S por* propagação unitária *se S' se obtém de S por repetição da seguinte transformação: se S contém uma cláusula unitária (i.e. com um único literal l), então:*

1. *remover de S todas as cláusulas da forma $l \vee C'$*
2. *substituir em S cada cláusula da forma $\tilde{l} \vee C'$ pela cláusula C'.*

Exemplo 1.4. *Considera o seguinte conjunto de cláusulas:*

$$\begin{aligned}&\{p_1, \neg p_1 \lor \neg p_2, p_3 \lor p_2, \neg p_7 \lor p_2, \neg p_3 \lor p_4, \neg p_3 \lor p_5, \neg p_4 \lor \neg p \lor q,\\ &\qquad \neg p_5 \lor \neg p_6 \lor r, \neg p \lor \neg q \lor p_6, p \lor p_7, \neg r \lor p_7\}\end{aligned} \tag{1.1}$$

Aplicando a propagação a p_1 resulta em:

$$\begin{aligned}&\{\neg p_2, p_3 \lor p_2, \neg p_7 \lor p_2, \neg p_3 \lor p_4, \neg p_3 \lor p_5, \neg p_4 \lor \neg p \lor q,\\ &\qquad \neg p_5 \lor \neg p_6 \lor r, \neg p \lor \neg q \lor p_6, p \lor p_7, \neg r \lor p_7\}\end{aligned} \tag{1.2}$$

Aplicando a propagação a $\neg p_2$ resulta em:

$$\begin{aligned}&\{p_3, \neg p_7, \neg p_3 \lor p_4, \neg p_3 \lor p_5, \neg p_4 \lor \neg p \lor q,\\ &\qquad \neg p_5 \lor \neg p_6 \lor r, \neg p \lor \neg q \lor p_6, p \lor p_7, \neg r \lor p_7\}\end{aligned} \tag{1.3}$$

Depois de aplicar a propagação a p_3 e $\neg p_7$, temos:

$$\{p_4, p_5, \neg p_4 \lor \neg p \lor q, \neg p_5 \lor \neg p_6 \lor r, \neg p \lor \neg q \lor p_6, p, \neg r\} \tag{1.4}$$

Propagando para p_4, p_5,$\neg r$ e p vem:

$$\{q, \neg p_6, \neg q \lor p_6\} \tag{1.5}$$

E finalmente propagando para q e $\neg p_6$:

$$\{\boldsymbol{F}\} \tag{1.6}$$

o que determina que o conjunto inicial de cláusulas é não satisfazível (e neste caso usando só propagação unitária).

O algoritmo básico de Davis-Putnam é o seguinte:

```
DLL(S) {
  input: conjunto de clausulas S
  output: satisfazivel ou nao satisfazivel
  S := propaga(S)
  if S vazio then return satisfazivel
  if S contem F then return nao satisfazivel
  l := selecciona_literal(S)
  if DLL(S ∪ {l}) = satisfazivel
     then return satisfazivel
     else return DLL(S ∪ {l̃})
}
```

A função `propaga` implementa a propagação unitária. A função `seleciona_literal` retorna um literal da cláusula. Esta função poderá ser considerada um parâmetro do algoritmo, pois uma escolha adequada do literal pode tornar o algoritmo mais eficiente. Possíveis critérios são: escolher uma variável que ocorre mais vezes; que o produto das ocorrências de l e $\tilde{l}$ é máximo; que ocorre mais vezes em cláusulas de tamanho minimal, etc.
Para verificar a satisfazibilidade de um conjunto de fórmulas será necessário primeiro convertê-lo para um conjunto de cláusulas.

Exemplo 1.5. *Vamos aplicar o algoritmo `DLL` ao seguinte conjunto de cláusulas:*

$$S = \{\neg p \vee \neg q, \neg p \vee q, p \vee \neg q, p \vee q\} \tag{1.7}$$

Como não tem cláusulas unitárias não podemos aplicar a propagação. Temos que seleccionar um literal, por exemplo, $\neg p$. Consideremos, primeiro, o conjunto S aumentado com $\neg p$. Podemos nesta caso aplicar a propagação ($\Rightarrow$) sucessivamente:

$$\begin{aligned}\{\neg p \vee \neg q, \neg p \vee q, p \vee \neg q, p \vee q, \neg p\} \Rightarrow& \\ \{\neg q, q\} \Rightarrow \{\boldsymbol{F}\}&\end{aligned} \tag{1.8}$$

Como $S \cup \{\neg p\}$ é não satisfazível, pelo algoritmo temos que considerar ainda $S \cup \{p\}$. Neste caso a propagação é:

$$\begin{aligned}\{\neg p \vee \neg q, \neg p \vee q, p \vee \neg q, p \vee q, p\} \Rightarrow& \\ \{\neg q, q\} \Rightarrow \{\boldsymbol{F}\}&\end{aligned} \tag{1.9}$$

E o algoritmo retorna `nao satisfazivel`.

Pode-se demonstrar que:

Proposição 1.5. *O algoritmo `DLL` é correcto e completo para a satisfazibilidade de um conjunto de cláusulas S, i.e. o algoritmo retorna `satisfazivel` se S é satisfazível e retorna `nao satisfazivel` se S é não satisfazível.*

Podemos considerar ainda algumas optimizações a este algoritmo. A primeira é a *eliminação de tautologias* que pode ser feita apenas uma vez quando se converte uma fórmula (ou conjunto de fórmulas) para um conjunto de cláusulas (e usando o Lema 1.3), dado que o algoritmo não introduz tautologias. Outra optimização pode ser feita quando uma variável p só aparece positivamente ou só negativamente ($\neg p$) num conjunto de cláusulas.

Definição 1.11. *Um literal l é* puro *num conjunto de cláusulas S, se S não contém cláusulas da forma $\tilde{l} \vee C$. A* eliminação de literais puros *remove do conjunto de cláusulas todas as cláusulas que contêm um literal puro.*

Exemplo 1.6. *Considera a fórmula* $\neg((p \to q) \ \wedge \ (p \ \wedge \ q \to r) \to (\neg p \to r))$. *Uma FNC equivalente em forma clausal é:*

$$\{\neg p \ \vee \ q, \neg p \ \vee \ \neg q \ \vee \ r, \neg p, \neg r\} \tag{1.10}$$

O literal $\neg p$ *é puro neste conjunto, portanto todas as cláusulas em que ele ocorre podem ser eliminadas. Ficámos apenas com* $\{\neg r\}$. *Como também é puro, podemos eliminar e concluir que o conjunto é* satisfazível.

Para implementar esta eliminação, basta termos um contador com o número de ocorrências de cada literal l. Quando esse contador for 0, o literal $\tilde{l}$ é puro.

Exercício 1.15. *Aplica o algoritmo DLL ao seguinte conjunto de cláusulas:*

$$\begin{aligned}\{p \ \vee \ q \ \vee \ r, \neg p \ \vee \ \neg q \ \vee \ \neg r, p \ \vee \ \neg q \ \vee \ \neg r p \ \vee \ q \ \vee \ \neg r,&\\ \neg p \ \vee \ q, \neg p \ \vee \ r, p \ \vee \ \neg q \ \vee \ r\}&\end{aligned} \tag{1.11}$$

◇

Exercício 1.16. *O seguinte conjunto de cláusulas formaliza o princípio de Dirichlet (ou* pigeonhole*) para 3 pombos e 2 poleiros: é impossível colocar 3 pombos em 2 poleiros de modo que cada poleiro só comtém um pombo! Considera o seguinte conjunto de variávies proposicionais* $\{p_{ij} \mid i = 1..3, j = 1, 2\}$ *em que* p_{ij} *denota que pombo* i *foi colocado no poleiro* j. *As cláusulas seguintes indicam que cada pombo é colocado nalgum poleiro:*

$$p_{11} \ \vee \ p_{12}, p_{21} \ \vee \ p_{22}, p_{31} \ \vee \ p_{32}$$

Agora é necessário formalizar que cada poleiro tem no máximo um pombo: para cada par de pombos i_1 *e* i_2 *e qualquer poleiro* j *tem-se que ambos não podem estar em* j, *i.e.* $\neg(p_{i_1 j} \wedge p_{i_2 j})$ *ou equivalentemente* $\neg p_{i_1 j} \ \vee \ \neg p_{i_2 j}$. *Temos então 6 cláusulas:*

$$\begin{gathered}\neg p_{11} \ \vee \ \neg p_{21}, \neg p_{11} \ \vee \ \neg p_{31}, \neg p_{21} \ \vee \ \neg p_{31},\\ \neg p_{12} \ \vee \ \neg p_{22}, \neg p_{12} \ \vee \ \neg p_{32}, \neg p_{22} \ \vee \ \neg p_{32}.\end{gathered}$$

Mostra a não satisfazibilidade o conjunto de 9 cláusulas usando o algoritmo `DLL`. ◇

Leituras suplementares [BE00] (Cap. 3, 4, 7) [HR00] (Cap 1.5) [GLM97] (Cap 2.2, 4.3)

1.4 Sistemas dedutivos

Considera os raciocínios seguintes:

1	Todos os homens são mortais
2	Sócrates é um homem
3	Sócrates é mortal

e

1	Todos os actores ricos são bons actores
2	Brad Pitt é um actor rico
3	Brad Pitt é bom actor

Informalmente, um *raciocínio* é uma sequência de afirmações das quais uma – a conclusão – deve ser consequência das restantes – as premissas. A conclusão é uma consequência lógica das premissas, se for verdadeira sempre que as premissas forem verdadeiras. Neste caso temos uma *raciocínio válido*. Num raciocínio válido se as premissas forem verdadeiras, o raciocínio é *integro*. Nos exemplos anteriores, o primeiro raciocínio é íntegro, mas o segundo não: a primeira premissa é falsa!

1.4.1 Métodos de dedução

Como podemos mostrar que uma conclusão *é uma consequência lógica* das premissas?
Construindo uma *sucessão de passos* em que em cada um a conclusão é inequivocamente consequência das conclusões e premissas anteriores. Formalmente iremos considerar *sistemas de dedução*.
Por outro lado, para mostrar que uma conclusão *não é consequência lógica* das premissas, temos que mostrar que existe uma situação em que as premissas podem ser verdadeiras e a conclusão falsa. Essa situação é designada de *contra-exemplo*.

1.4.2 Sistemas de dedução axiomáticos

Um sistema de dedução axiomático $\mathcal{D}$ é um método sintáctico, constituído por:

axiomas (lógicos): fórmulas base, que caracterizam as propriedades das conectivas

regras de inferência: modos de obter fórmulas a partir de outras

Definição 1.12. *Uma sucessão finita de fórmulas $\phi_1, \ldots \phi_n$ é uma* dedução *de ϕ_n em $\mathcal{D}$ a partir de um conjunto Σ de fórmulas se para cada $1 \leq i \leq n$ se verifica:*

- $\phi_i \in \Sigma$
- ϕ_i *é um axioma*
- ϕ_i *resulta de $\phi_1 \ldots \phi_{i-1}$ por aplicação duma regra de inferência*

Neste caso diz-se também que ϕ_n pode ser deduzido a partir de Σ e escreve-se $\Sigma \vdash_{\mathcal{D}} \phi_n$. A fórmula ϕ_n é um teorema *(de $\mathcal{D}$) se $\Sigma = \emptyset$ e escreve-se $\vdash_{\mathcal{D}} \phi_n$. Neste caso, a dedução $\phi_1, \ldots \phi_n$*

diz-se uma demonstração *de* ϕ_n*. Se* $\Sigma = \{\theta_1, \ldots, \theta_n\}$ *é finito, em vez de* $\Sigma \vdash_{\mathcal{D}} \phi$ *escreve-se*

$$\theta_1, \ldots, \theta_n \vdash_{\mathcal{D}} \phi$$

e $\mathcal{D}$ *será omitido se for explícito no contexto.*

1.4.3 Sistema de dedução natural, DN

Sistema inventado por G. Gentzen (1935) (e também por S. Jaskowski), e cujas regras pretendem reflectir as formas de raciocínio usadas nas demonstrações matemáticas. Não tem axiomas, só regras de inferência. Uma das suas originalidades, em relação a outros sistemas axiomáticos, é possibilidade de introduzir hipóteses no meio da dedução, mas que terão se ser eliminadas antes da dedução perminar. Outra característica é a dualidade das regras. Para cada conectiva lógica existem dois tipos de regras: de *introdução* (da conectiva) e de *eliminação* (da conectiva).

Consideremos um exemplo. Suponhamos que queremos concluir que

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

é uma tautologia (é válida).

Para tal, supomos que $(p \vee (q \wedge r))$ se verifica e tentamos concluir $((p \vee q) \wedge (p \vee r))$. Como temos uma disjunção no antecedente, temos que supor separadamente que p se verifica ou que $(q \wedge r)$ se verifica (*eliminação de* $\vee$). Suponhamos p, então $p \vee q$ também se verifica (pois numa disjunção basta que um se verifique) (*introdução de* $\vee$), e também temos $p \vee r$ (*introdução de* $\vee$). Mas então, também se verifica a sua conjunção $((p \vee q) \wedge (p \vee r))$ (*introdução de* $\wedge$). Agora se $(q \wedge r)$ se verifica, então q e r verificam-se (*eliminação de* $\wedge$). Então $(p \vee q)$ e $(p \vee r)$, também se verifica, assim como a sua conjunção. Temos a seguinte *árvore de demonstração*:

$$\begin{array}{c|c}
p & q \wedge r \\
p \vee q & q \\
p \vee r & r \\
((p \vee q) \wedge (p \vee r)) & p \vee q \\
 & p \vee r \\
 & ((p \vee q) \wedge (p \vee r))
\end{array}$$

Como supondo $(p \vee (q \wedge r))$ se "deduz" $((p \vee q) \wedge (p \vee r))$, podemos concluir que $(p \vee (q \wedge r)) \rightarrow ((p \vee q) \wedge (p \vee r))$ (*introdução de* $\rightarrow$).

Vamos agora formalizar as regras de inferência DN.

Uma regra de inferência é da forma:

de $\phi_1 \ldots \phi_k$ infere-se ϕ_n

e pode ser representada graficamente (em árvore) por:

$$\frac{\phi_1, \ldots, \phi_k}{\phi_n}$$

1.4.3.1 Regras DN para a conjunção

Introdução de $\wedge$

$$\frac{\phi \qquad \psi}{\phi \ \wedge \ \psi} \wedge \mathbf{I}$$

Se já deduzimos ϕ e ψ então podemos deduzir $\phi \ \wedge \ \psi$

Eliminação de $\wedge$

$$\frac{\phi \ \wedge \ \psi}{\phi} \wedge \mathbf{E}_1 \qquad \frac{\phi \ \wedge \ \psi}{\psi} \wedge \mathbf{E}_2$$

Se deduzimos $\phi \ \wedge \ \psi$ podemos deduzir ϕ; e podemos também deduzir ψ.

Exemplo 1.7. *Mostrar que $p \ \wedge \ q, r \vdash q \ \wedge \ r$.*

Resolução 1.7.1. *Podemos construir a dedução numa* árvore*:*

$$\frac{\frac{p \wedge q}{q} \wedge \boldsymbol{E}_2 \qquad r}{q \wedge r} \wedge \boldsymbol{I}$$

em que as folhas são as premissas. Mas estas árvores podem ficar muito grandes, portanto vamos considerar uma representação linear para as deduções: numeram-se os passos, separam-se as premissas e, para cada passo, indica-se qual a regra a aplicar e quais as fórmulas que intervêm. Para a dedução anterior, temos

1	$p \ \wedge \ q$	
2	r	
3	q	$\wedge E$, 1
4	$q \ \wedge \ r$	$\wedge I$, 3, 2

Esta notação para a representação de deduções naturais denomina-se notação de Fitch.

Exemplo 1.8. *Mostrar que $(p \ \wedge \ q) \ \wedge \ r \vdash r \ \wedge \ q$*

Resolução 1.8.1.

1	$(p \wedge q) \wedge r$	
2	$p \wedge q$	$\wedge E$, 1
3	r	$\wedge E$, 1
4	q	$\wedge E$, 2
5	$r \wedge q$	$\wedge I$, 3, 4

1.4.3.2 Regras DN para a disjunção

Introdução de $\vee$

$$\frac{\phi}{\phi \vee \psi} \vee \mathbf{I}_1 \qquad \frac{\psi}{\phi \vee \psi} \vee \mathbf{I}_2$$

Se já deduzimos ϕ podemos deduzir qualquer disjunção que contenha ϕ.

Eliminação de $\vee$

$$\frac{\phi \vee \psi \qquad \begin{array}{c}[\phi]\\ \vdots\\ \gamma\end{array} \quad \begin{array}{c}[\psi]\\ \vdots\\ \gamma\end{array}}{\gamma} \vee \mathbf{E}$$

- se já deduzimos a disjunção $\phi \vee \psi$
- se supusermos ϕ deduzirmos γ (numa sub-dedução)
- e se supusermos ψ deduzirmos γ (numa sub-dedução)
- então podemos deduzir γ

Nestas regras, expressões $[\phi]$ indicam que estamos iniciar uma sub-dedução com premissa ϕ, que só deve ser considerada nessa sub-dedução. As sub-deduções têm de terminar pela ordem em que foram iniciadas. Neste caso as sub-deduções terminam quando γ for deduzido. Na notação de Fitch, cada sub-dedução é iniciada com uma indentação.

Exemplo 1.9. *Mostrar que* $(p \wedge q) \vee (q \wedge r) \vdash q$.

Resolução 1.9.1.

$$
\begin{array}{r|ll}
1 & (p \land q) \lor (q \land r) & \\
\hline
2 & \quad \mid p \land q & \\
3 & \quad \mid q & \land E,\ 2 \\
4 & \quad \mid q \land r & \\
5 & \quad \mid q & \land E,\ 4 \\
6 & q & \lor E,\ 1,\ 2\text{–}3,\ 4\text{–}5
\end{array}
$$

1.4.3.3 Regra DN de Repetição

Numa dedução podemos sempre repetir uma conclusão já obtida. A essa regra chamaremos **repetição**:

$$\frac{\phi}{\phi}\mathbf{R}$$

Exemplo 1.10. *Mostrar que* $(p \land q) \lor q \vdash q$

Resolução 1.10.1.

$$
\begin{array}{r|ll}
1 & (p \land q) \lor q & \\
\hline
2 & \quad \mid p \land q & \\
3 & \quad \mid q & \land E,\ 2 \\
4 & \quad \mid q & \\
5 & \quad \mid q & R,\ 4 \\
6 & q & \lor E,\ 1,\ 2\text{–}3,\ 4\text{–}5
\end{array}
$$

1.4.3.4 Utilização de sub-deduções

Como já foi referido, uma dedução pode ser composta por sub-deduções que introduzem novas premissas. Mas nem essa premissa, nem as fórmulas delas deduzidas podem ser usadas depois da sub-dedução em que ocorrem terminar. Considera a seguinte *dedução*:

1	$(p \wedge q) \vee (q \wedge r)$	
2	$\quad p \wedge q$	
3	$\quad p$	$\wedge$E, 2
4	$\quad q$	$\wedge$E, 2
5	$\quad q \wedge r$	
6	$\quad q$	$\wedge$E, 5
7	q	$\vee$E, 1, 2–4, 5–6
8	$q \wedge p$	$\wedge$I, 7, 3

Esta dedução está **ERRADA**! No passo 8 é usado um passo que ocorre numa sub-dedução que já terminou. Uma sub-dedução é iniciada com a introdução de novas hipóteses (premissas) e as deduções aí feitas dependem delas. Quando termina a sub-dedução, essas hipóteses deixam ser assumidas e portanto não se podem utilizar!

1.4.3.5 Regras DN para a Negação

Eliminação de ¬

Corresponde a uma das partes do princípio da dupla negação.

$$\frac{\neg\neg\phi}{\phi}\,\neg\mathbf{E}$$

Introdução de ¬

Esta regra corresponde a demonstrações por contradição. Representamos por **F** uma contradição (p.e., $\phi \wedge \neg\phi$).

$$\frac{\begin{matrix}[\phi]\\ \vdots \\ \mathbf{F}\end{matrix}}{\neg\phi}\,\neg\mathbf{I}$$

Se supondo ϕ podemos deduzir uma contradição, então podemos deduzir $\neg\phi$ das premissas originais.

1.4.3.6 Regras DN para F

Se não considerarmos **F** como uma abreviatura de $\phi \wedge \neg\phi$, temos de ter uma regra para o introduzir[1]:

[1] caso contrário podemos ignorar...

Introdução de F

$$\frac{\begin{array}{c}\phi \\ \vdots \\ \neg\phi\end{array}}{\mathbf{F}}\,\mathbf{FI}$$

Se deduzimos ϕ e $\neg\phi$ então temos uma contradição.

Exemplo 1.11. *Mostrar que* $\phi \vdash \neg\neg\phi$.

Resolução 1.11.1.

1	ϕ	
2	$\quad \neg\phi$	
3	$\quad \boldsymbol{F}$	$\boldsymbol{FI}$, 1, 2
4	$\neg\neg\phi$	$\neg I$, 2–3

Eliminação de F

$$\frac{\mathbf{F}}{\phi}\,\mathbf{FE}$$

Se deduzimos uma contradição, então podemos deduzir qualquer fórmula.

Definição 1.13. *Um conjunto de fórmulas* Σ *diz-se* inconsistente *se* $\Sigma \vdash \boldsymbol{F}$.

1.4.3.7 Métodos de demonstração

Vamos ilustrar algumas aplicações das regras anteriores, em demonstrações em matemática.

Demonstração por casos A regra da **eliminação da disjunção** corresponde ao método de demonstração por casos. Consideremos o seguinte problema:

Mostrar que existem irracionais b e c tal que b^c é racional

Demonstração. Demonstração por casos: Seja $\sqrt{2}^{\sqrt{2}}$. Este número é racional ou irracional.

- Se $\sqrt{2}^{\sqrt{2}}$ é racional então basta tomar $b = c = \sqrt{2}$
- Se $\sqrt{2}^{\sqrt{2}}$ é irracional, então seja $b = \sqrt{2}^{\sqrt{2}}$ e $c = \sqrt{2}$. Vem $b^c = \sqrt{2}^{\sqrt{2}.\sqrt{2}} = \sqrt{2}^2 = 2$, que é racional.

□

É de referir que a demonstração anterior não é construtiva, uma vez que não foi determinado se $\sqrt{2}^{\sqrt{2}}$ é ou não racional.

Demonstração por contradição A regra da **introdução da negação**, é usada nas demonstrações por contradição, isto é, supõe-se a negação do que se quer provar e chega-se a um absurdo. Suponhamos o seguinte problema:

Mostrar que $\sqrt{2}$ *não é racional.*

Demonstração. Suponhamos que $\sqrt{2}$ é racional. Então existem p e q tal que $\sqrt{2} = p/q$, com um deles ímpar (porquê?). Então $\frac{p^2}{q^2} = 2$. E $p^2 = 2q^2$. Então p^2 é par e p também (verifica!). E $4 \mid p^2$ e $4 \mid 2q^2$. Mas então q^2 também é par! Temos então uma contradição. $\square$

Exemplo 1.12. *Mostrar usando o sistema DN:*

a) $\neg p \lor \neg q \vdash \neg(p \land q)$

b) $\vdash \neg(p \land \neg p)$

c) $\neg(\neg p \lor q) \vdash p$

Resolução 1.12.1. *a)* $\neg p \lor \neg q \vdash \neg(p \land q)$

1	$\mid\ \neg p \lor \neg q$	
2	$\mid\ \mid\ \neg p$	
3	$\mid\ \mid\ \mid\ p \land q$	
4	$\mid\ \mid\ \mid\ p$	$\land E$, 3
5	$\mid\ \mid\ \mid\ \boldsymbol{F}$	$\boldsymbol{FI}$, 2, 4
6	$\mid\ \mid\ \neg(p \land q)$	$\neg I$, 3–5
7	$\mid\ \mid\ \neg q$	
8	$\mid\ \mid\ \mid\ p \land q$	
9	$\mid\ \mid\ \mid\ q$	$\land E$, 8
10	$\mid\ \mid\ \mid\ \boldsymbol{F}$	$\boldsymbol{FI}$, 7, 9
11	$\mid\ \mid\ \neg(p \land q)$	$\neg I$, 8–10
12	$\mid\ \neg(p \land q)$	$\lor E$, 1, 2–11

b) $\vdash \neg(p \land \neg p)$

1	$\mid\ \mid\ p \land \neg p$	
2	$\mid\ \mid\ p$	$\land E$, 1
3	$\mid\ \mid\ \neg p$	$\land E$, 1
4	$\mid\ \mid\ \boldsymbol{F}$	$\boldsymbol{FI}$, 2, 3
5	$\mid\ \neg(p \land \neg p)$	$\neg I$, 1–4

c) $\neg(\neg p \lor q) \vdash p$

$$\begin{array}{r|l|l}
1 & \neg(\neg p \lor q) & \\ \hline
2 & \quad \neg p & \\
3 & \quad \neg p \lor q & \lor I,\ 2 \\
4 & \quad \boldsymbol{F} & \boldsymbol{FI},\ 1,\ 3 \\
5 & \neg\neg p & \neg I,\ 2\text{–}4 \\
6 & p & \neg E,\ 2\text{–}4
\end{array}$$

Leituras suplementares [BE00] (Cap. 2,5,6)

1.4.3.8 Regras *DN* para a implicação

Eliminação de → (*modus ponens*)

Esta regra é habitualmente conhecida por *modus ponens* (em latim, *modo que afirma*) e corresponde a raciocínios condicionais:

$$\frac{\phi \quad \phi \to \psi}{\psi}\to\mathbf{E}$$

Se já deduzimos ϕ e $\phi \to \psi$, então podemos deduzir ψ.

Exemplo 1.13. *Mostrar que* $p, p \to q, p \to (q \to r) \vdash r$.

Resolução 1.13.1.

$$\begin{array}{r|l|l}
1 & p \to (q \to r) & \\
2 & p \to q & \\
3 & p & \\ \hline
4 & q \to r & \to E,\ 1,\ 3 \\
5 & q & \to E,\ 2,\ 3 \\
6 & r & \to E,\ 4,\ 5
\end{array}$$

Introdução de → (regra da dedução)

A regra para introduzir uma implicação necessita duma sub-dedução: supondo ϕ tentamos deduzir ψ. Se tal acontecer, terminamos a sub-dedução (retirando a suposição) e concluímos $\phi \to \psi$:

$$\begin{array}{c} [\phi] \\ \vdots \\ \dfrac{\psi}{\phi \to \psi}\to\mathbf{I} \end{array}$$

Exemplo 1.14. *Mostrar que* $(p \lor q) \to r \vdash p \to r$.

Resolução 1.14.1.

1	$(p \lor q) \to r$	
2	$\quad p$	
3	$\quad p \lor q$	$\lor I$, 2
4	$\quad r$	$\to E$, 1, 3
5	$p \to r$	$\to I$, 2–4

1.4.3.9 Deduções sem premissas

Com a introdução da implicação (regra da dedução) podemos converter qualquer dedução com premissas numa dedução sem premissas:

Exemplo 1.15. *Mostrar que* $\vdash \phi \to \neg\neg\phi$.

Resolução 1.15.1.

1	$\quad \phi$	
2	$\quad\quad \neg\phi$	
3	$\quad\quad \boldsymbol{F}$	***FI***, 1, 2
4	$\quad \neg\neg\phi$	$\neg I$, 2–3
5	$\phi \to \neg\neg\phi$	$\to I$, 1–4

Em geral temos:

Lema 1.4. ***(da dedução)*** $\Sigma \cup \{\phi\} \vdash \psi$ *se e só se* $\Sigma \vdash \phi \to \psi$.

Demonstração. $\Rightarrow$: Se $\Sigma \cup \{\phi\} \vdash \psi$, obtemos $\Sigma \vdash \phi \to \psi$, supondo ϕ, usando a dedução anterior até obter ψ e aplicando a regra $\to$ **I**.

$\Leftarrow$: Se $\Sigma \vdash \phi \to \psi$, para obter $\Sigma \cup \{\phi\} \vdash \psi$, usamos apenas dedução de ψ supondo ϕ. □

1.4.3.10 Algumas regras derivadas de *DN*

A partir das regras base podemos obter regras derivadas que correspondem a teoremas no sistema *DN* (eventualmente usando o lema da dedução).

Modus Tollens (em latim, *modo que nega*)

$$\frac{\phi \to \psi \qquad \neg\psi}{\neg\phi}\,\textbf{MT}$$

Exemplo 1.16. *Mostrar que* $\phi \to \psi, \neg\psi \vdash \neg\phi$.

Resolução 1.16.1.

1	$\phi \to \psi$	
2	$\neg\psi$	
3	$\quad \phi$	
4	$\quad \psi$	$\to E$, 1, 3
5	$\quad \boldsymbol{F}$	$\boldsymbol{FI}$, 2, 4
6	$\neg\phi$	$\neg I$, 3–5

Introdução da dupla negação

$$\frac{\phi}{\neg\neg\phi}\,\neg\neg\mathbf{I}$$

Redução ao absurdo

$$\frac{\begin{array}{c}[\neg\phi]\\ \vdots \\ \mathbf{F}\end{array}}{\phi}\,\mathbf{RA}$$

Se tivermos uma dedução de $\mathbf{F}$ supondo $\neg\phi$ podemos ter uma dedução de $\neg\phi \to \mathbf{F}$. Então basta mostrar $\neg\phi \to \mathbf{F} \vdash \phi$:

1	$\neg\phi \to \mathbf{F}$	
2	$\quad \neg\phi$	
3	$\quad \mathbf{F}$	→E, 1, 2
4	$\neg\neg\phi$	¬I, 2–3
5	ϕ	¬E, 4

Terceiro excluído

$$\frac{}{\phi \lor \neg\phi}\,\mathbf{TE}$$

1	$\neg(\phi \vee \neg\phi)$	
2	ϕ	
3	$\phi \vee \neg\phi$	$\vee$I, 2
4	**F**	**FI**, 1, 3
5	$\neg\phi$	$\neg$I, 2–4
6	$\phi \vee \neg\phi$	$\vee$I, 5
7	**F**	**FI**, 1, 5
8	$\phi \vee \neg\phi$	**RA**, 1–7

Exemplo 1.17. *Mostrar* $\neg q \to \neg p \vdash p \to \neg\neg q$.

Resolução 1.17.1.

1	$\neg q \to \neg p$	
2	p	
3	$\neg\neg p$	$\neg$*I*, 2
4	$\neg\neg q$	***MT***, 1, 3
5	$p \to \neg\neg q$	$\to$*I*, 2–4

1.4.3.11 Equivalência dedutiva

Dadas dumas fórmulas ϕ e ψ, dizemos que ϕ e ψ são *dedutivamente equivalentes* se e só se $\phi \vdash \psi$ e $\psi \vdash \phi$. E denotamos por $\phi \dashv\vdash \psi$.

Exercício 1.17. *Mostra que* $\phi \dashv\vdash \psi$ *se e só se* $\vdash (\phi \to \psi) \wedge (\psi \to \phi)$. $\diamond$

Proposição 1.6 (Contraposição)**.**

$$\phi \to \psi \dashv\vdash \neg\psi \to \neg\phi$$

Demonstração. Vamos mostrar só $\phi \to \psi \vdash \neg\psi \to \neg\phi$:

1	$\phi \to \psi$	
2	$\neg\psi$	
3	$\neg\phi$	**MT**, 1, 2
4	$\neg\psi \to \neg\phi$	$\to$I, 2–3

$\square$

1.4.3.12 Ilustração da aplicação das regras

Demonstração duma afirmação condicional Como já vimos, a regra da **introdução da implicação** corresponde à demonstração de uma condicional. Suponhamos que queremos mostrar que:

Se n^2 é par então n é par

Demonstração. Suponhamos que n^2 é par. Vamos provar que n é par, por contradição. Suponhamos que n é ímpar, então $n = 2m + 1$, para algum m. Então:

$$n^2 = (2m+1)^2 = 4m^2 + 4m + 1 = 2(2m^2 + 2m) + 1$$

que é ímpar! Então n é par. E temos demonstrada a implicação. □

Demonstração por contraposição De igual modo temos as demonstrações por contraposição. Suponhamos que queremos mostrar que:

Se n não é par então n^2 não é par

Demonstração. Suponhamos que n não é par, i.e é da forma $n = 2m + 1$, para algum m. Então:

$$n^2 = (2m+1)^2 = 4m^2 + 4m + 1 = 2(2m^2 + 2m) + 1$$

o que mostra que n^2 é ímpar, e portanto não é par. Então temos demonstrada a implicação. □

Neste caso evita-se a redução ao absurdo...

Exemplo 1.18. *Mostra que:*

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

b) $\neg(\phi \wedge \psi) \vdash \neg\phi \vee \neg\psi$

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

d) $\phi \to \psi \vdash \neg\phi \vee \psi$

e) $\vdash \phi \to (\psi \to \phi)$

f) $\vdash (\phi \to (\psi \to \theta)) \to ((\phi \to \psi) \to (\phi \to \theta))$

g) $\vdash (\neg\psi \to \neg\phi) \to ((\neg\psi \to \phi) \to \psi)$

Resolução 1.18.1. *a)* $\vdash p \vee \neg(p \wedge q)$

*Supondo a regra do terceiro excluído (**TE**)*

1	$\neg(p \wedge q) \vee (p \wedge q)$	***TE***
2	$p \wedge q$	
3	p	$\wedge E$, 2
4	$p \vee \neg(p \wedge q)$	$\vee I$, 3
5	$\neg(p \wedge q)$	
6	$p \vee \neg(p \wedge q)$	$\vee I$, 5
7	$p \vee \neg(p \wedge q)$	$\vee E$, 1, 2–4, 5–6

b) $\neg(\phi \wedge \psi) \vdash \neg\phi \vee \neg\psi$

1	$\neg(\phi \wedge \psi)$	
2	$\neg(\neg\phi \vee \neg\psi)$	
3	$\neg\phi$	
4	$\neg\phi \vee \neg\psi$	$\vee I$, 3
5	$\boldsymbol{F}$	***FI***, 2, 4
6	$\neg\neg\phi$	$\neg I$, 3–5
7	ϕ	$\neg E$, 6
8	$\neg\psi$	
9	$\neg\phi \vee \neg\psi$	$\vee I$, 3
10	$\boldsymbol{F}$	***FI***, 2, 9
11	$\neg\neg\psi$	$\neg I$, 8–10
12	ψ	$\neg E$, 11
13	$\phi \wedge \psi$	$\wedge I$, 7, 12
14	$\boldsymbol{F}$	***FI***, 1, 13
15	$\neg\neg(\neg\phi \vee \neg\psi)$	$\neg I$, 2, 14
16	$\neg\phi \vee \neg\psi$	$\neg E$, 15

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

Usar a regra do terceiro excluído com $(p \wedge q) \vee \neg(p \wedge q)$ *e ver a resolução do exercício anterior* *d)* $\phi \to \psi \vdash \neg\phi \vee \psi$

1	$\phi \to \psi$	
2	$\neg(\neg\phi \vee \psi)$	
3	$\neg\phi$	
4	$\neg\phi \vee \psi$	$\vee I$, 3
5	$\boldsymbol{F}$	$\boldsymbol{FI}$, 2, 4
6	$\neg\neg\phi$	$\neg I$, 3–5
7	ϕ	$\neg E$, 6
8	ψ	$\to E$, 1, 7
9	$\neg\phi \vee \psi$	$\vee I$, 8
10	$\boldsymbol{F}$	$\boldsymbol{FI}$, 2, 9
11	$\neg\neg(\neg\phi \vee \psi)$	$\neg I$, 2–10
12	$\neg\phi \vee \psi$	$\neg E$, 11

e) $\vdash \phi \to (\psi \to \phi)$

1	ϕ	
2	ψ	
3	ϕ	R, 1
4	$\psi \to \phi$	$\to I$, 2–3
5	$\phi \to (\psi \to \phi)$	$\to I$, 1–4

f) $\vdash (\phi \to (\psi \to \theta)) \to ((\phi \to \psi) \to (\phi \to \theta))$

1	$\phi \to (\psi \to \theta)$	
2	$(\phi \to \psi)$	
3	ϕ	
4	ψ	$\to E$, 2, 3
5	$\psi \to \theta$	$\to E$, 1, 3
6	θ	$\to E$, 4, 5
7	$\phi \to \theta$	$\to I$, 3–6
8	$(\phi \to \psi) \to (\phi \to \theta)$	$\to I$, 2–7
9	$\phi \to (\psi \to \theta) \to (\phi \to \psi) \to (\phi \to \theta)$	$\to I$, 1–8

g) $\vdash (\neg\psi \to \neg\phi) \to ((\neg\psi \to \phi) \to \psi)$

1	$\quad \neg\psi \to \neg\phi$	
2	$\quad\quad \neg\psi \to \phi$	
3	$\quad\quad\quad \neg\psi$	
4	$\quad\quad\quad \phi$	$\to E$, 2, 3
5	$\quad\quad\quad \neg\phi$	$\to E$, 1, 2
6	$\quad\quad\quad \boldsymbol{F}$	$\boldsymbol{FI}$, 4, 5
7	$\quad\quad \neg\neg\psi$	$\neg I$, 3–6
8	$\quad\quad \psi$	$\neg E$, 7
9	$\quad (\neg\psi \to \phi) \to \psi$	$\to I$, 2–8
10	$(\neg\psi \to \neg\phi) \to (\neg\psi \to \phi) \to \psi$	$\to I$, 1–9

Leituras suplementares [BE00] (Cap. 8.1-2)

1.5 Integridade e completude de um sistema dedutivo

Um sistema dedutivo é um conjunto de regras, puramente sintácticas. Para que possa ser usado para obter a validade ou consequência lógica de fórmulas é necessário (desejável) que seja **íntegro** e **completo**. Isto é: que o que se deduza seja consequência semântica das premissas, e se uma fórmula for consequência semântica das premissas então existe uma dedução para ela.

Definição 1.14 (Integridade)**.** *Um sistema dedutivo é* íntegro*, se dado um conjunto* Σ *de premissas e uma conclusão* ϕ*, se existe uma dedução de* ϕ *com premissas* Σ*, i.e* $\Sigma \vdash_{\mathcal{D}} \phi$*, então a conclusão* ϕ *é consequência semântica de* Σ*, i.e* $\Sigma \models \phi$*. Em particular, se* $\vdash_{\mathcal{D}} \phi$ *então* ϕ *é uma tautologia* ($\models \phi$) *.*

Definição 1.15 (Completude)**.** *Um sistema dedutivo é* completo*, se dado um conjunto* Σ *de premissas e uma conclusão* ϕ*, se* ϕ *é consequência semântica de* Σ*, i.e* $\Sigma \models \phi$*, então existe uma dedução de* ϕ *com premissas* Σ*, i.e* $\Sigma \vdash_{\mathcal{D}} \phi$*. Em particular, se* ϕ *é uma tautologia* ($\models \phi$) *então* $\vdash_{\mathcal{D}} \phi$*.*

1.5.1 Integridade do sistema de dedução natural DN

Proposição 1.7. *Se* $\Sigma \vdash \phi$ *então* $\Sigma \models \phi$*.*

Demonstração. Suponhamos que temos uma dedução de ϕ, $\phi_1, \ldots, \phi_n = \phi$. Vamos mostrar que em cada passo a fórmula que aí ocorre é consequência semântica das premissas (ou hipóteses) que aí são assumidas.

Provamos por redução ao absurdo: Suponhamos que existe um passo p que contém uma fórmula que não é consequência semântica das premissas assumidas em p. E seja p o primeiro desses passos . Vamos ver que qualquer que seja a regra de DN aplicada em p, temos uma contradição. O que permite concluir que não existe tal passo p. Fazemos a demonstração por casos, considerando cada uma das regras. Basicamente temos dois tipos de casos: os que correspondem a regras com sub-deduções e os a regras sem sub-deduções.

Suponhamos que a regra a aplicar é a eliminação da implicação:

$\to$ **E**

$\dfrac{\phi \quad \phi \to \psi}{\psi}$: Seja θ a fórmula deduzida no passo p por aplicação de $\to$ **E** a $\phi \to \theta$ e ϕ. E sejam $\phi_1, \ldots, \phi_k$ as premissas assumidas em θ, e, por hipótese, θ não é consequência semântica delas. Mas as premissas para $\phi \to \theta$ e ϕ estão entre os $\phi_1, \ldots, \phi_k$ e ambos são consequência semânticas delas:

$$
\begin{array}{r|l|l|l}
1 & \phi_1 & & \\ \cline{2-2}
\vdots & \vdots & & \\
n & \phi \to \theta & & \\
\vdots & \vdots & & \\
 & & \phi_2 & \\ \cline{3-3}
\vdots & & \vdots & \\
l & & \phi & \\
\vdots & & \vdots & \\
 & & & \phi_3 \\ \cline{4-4}
\vdots & & & \vdots \\
p & & & \theta \\
\vdots & & & \vdots
\end{array}
$$

Considera a tabela de verdade para as fórmulas $\phi_1, \ldots, \phi_k$,ϕ,$\phi \to \theta$ e θ. Por hipótese, existe uma valorização v tal que $v(\phi_i) = \mathbf{V}$, $1 \leq i \leq k$ e $v(\theta) = \mathbf{F}$. Mas também tem-se que $v(\phi) = v(\phi \to \theta) = \mathbf{V}$. Mas isto contradiz a tabela de verdade para a implicação. Portanto concluímos que esta regra não pode ser aplicada no passo p (se existir).

Vejamos agora o caso da regra da introdução da implicação:

$\to$ **I**

$$\dfrac{\begin{array}{c}[\phi]\\ \vdots\\ \psi\end{array}}{\phi\to\psi}$$

: Suponhamos que o passo p deduz $\phi \to \theta$ por aplicação da regra $\to$ **I** a uma sub-dedução com premissa ϕ e conclusão θ. Sejam $\phi_1, \ldots, \phi_k$ as premissas assumidas em $\phi \to \theta$. Em θ as premissas são algumas das $\phi_1, \ldots, \phi_k$ e ϕ. E ϕ é consequência semântica dessas premissas.

$$\begin{array}{c|l} \vdots & \vdots \\ & \begin{array}{|l} \phi \\ \hline \vdots \\ \theta \end{array} \\ \vdots & \vdots \\ & \phi \to \theta \\ \vdots & \vdots \end{array}$$

Seja a tabela de verdade para $\phi_1, \ldots, \phi_k$, ϕ, $\phi \to \theta$ e θ. Por hipótese, existe uma valorização v tal que $v(\phi_i) = \mathbf{V}$, $1 \leq i \leq k$ e $v(\phi \to \theta) = \mathbf{F}$. Então temos também que $v(\phi) = \mathbf{V}$ e $v(\theta) = \mathbf{F}$. Mas isto *contradiz* o facto de θ ser consequência semântica de $\phi_1, \ldots, \phi_k$ e ϕ. Logo esta regra também não pode ser aplica no passo p.

Suponhamos que a regra a aplicar é a eliminação da disjunção:

$\vee$ **E**

$$\dfrac{\phi \vee \psi \qquad \begin{array}{c}[\phi]\\ \vdots\\ \gamma\end{array} \quad \begin{array}{c}[\psi]\\ \vdots\\ \gamma\end{array}}{\gamma}$$

: Suponhamos que o passo p deduz θ por aplicação da regra $\vee$ **E** a $\phi \vee \psi$ e duas sub-deduções uma com premissa ϕ outra com premissa ϕ e ambas com conclusão θ. Sejam $\phi_1, \ldots, \phi_k$ as premissas assumidas em θ (no passo p) e por, hipótese, θ não é consequência semântica delas. Mas $\phi \vee \psi$ é consequência semântica de algumas delas.

$$
\begin{array}{r|ll}
\vdots & \vdots & \\
n & \phi \vee \psi & \\
\vdots & \vdots & \\
 & & \multicolumn{1}{|l}{\phi} \\ \cline{3-3}
\vdots & & \multicolumn{1}{|l}{\vdots} \\
l & & \multicolumn{1}{|l}{\theta} \\
\vdots & \vdots & \\
 & & \multicolumn{1}{|l}{\psi} \\ \cline{3-3}
\vdots & & \multicolumn{1}{|l}{\vdots} \\
m & & \multicolumn{1}{|l}{\theta} \\
p & \theta & \\
\vdots & \vdots &
\end{array}
$$

Seja a tabela de verdade para $\phi_1, \ldots, \phi_k$, $\phi \vee \psi$, e θ. Por hipótese, existe uma valorização v tal que $v(\phi_i) = \mathbf{V}$, $1 \leq i \leq k$, $v(\phi \vee \psi) = \mathbf{V}$ e $v(\theta) = \mathbf{F}$. Então também, ou $v(\phi) = \mathbf{V}$ ou $v(\psi) = \mathbf{V}$. Ambos os casos obrigavam a $v(\theta) = \mathbf{V}$, no passo l ou no passo m respectivamente, uma vez que nesses passos θ é consequência semântica de $\phi_1, \ldots, \phi_k$, e ϕ ou ψ. Mas isto *contradiz* a hipótese de se ter $v(\theta) = \mathbf{F}$.

Vamos apenas considerar mais um caso. Suponhamos que se aplica a regra da eliminação de $\mathbf{F}$:

FE : Suponhamos que no passo p se deduz ϕ de $\mathbf{F}$. Sejam $\phi_1, \ldots, \phi_k$ as premissas assumidas em ϕ, que são as premissas assumidas em $\mathbf{F}$ (e das quais $\mathbf{F}$ é consequência semântica). Mas isto só pode ser se $\phi_1, \ldots, \phi_k$ forem todas contradições. E portanto ϕ é (vacuosamente) consequência semântica de $\phi_1, \ldots, \phi_k$.

$\dfrac{\mathbf{F}}{\phi}$

Depois de analisados os restantes casos e em todos obtermos uma contradição, podemos concluir que uma dedução no sistema DN não pode ter passos que não sejam consequência semântica das premissas. □

Corolário 1.3. *Se* $\vdash \phi$ *então* $\models \phi$.

A integridade da dedução natural permite-nos determinar se não existe uma dedução de uma fórmula ψ a partir de premissas $\phi_1, \ldots, \phi_n$ (i.e $\phi_1, \ldots, \phi_n \nvdash \psi$): basta encontrar uma valorização v tal que $v(\phi_i) = \mathbf{V}$, $1 \leq i \leq n$ e $v(\psi) = \mathbf{F}$ (i.e que $\phi_1, \ldots, \phi_n \not\models \psi$). Nota, contudo, que a integridade não permite concluir que se ψ é consequência semântica de $\phi_1, \ldots, \phi_n$ então existe uma dedução...isso é a completude.

Exercício 1.18. *Mostra que* $\neg p \lor (q \to p) \nvdash \neg p \land q$. $\diamond$

Exercício 1.19. *Termina a demonstração da proposição 1.7.* $\diamond$

1.5.2 Completude do sistema de dedução natural DN

Vamos ver que a dedução natural DN é completa para a lógica proposicional: qualquer consequência semântica pode ser deduzida em DN; em particular todas as tautologias são teoremas de DN.

Seja v uma atribuição de valores às variáveis. Para cada fórmula ψ, define-se

$$\psi^v = \begin{cases} \psi & \text{se } v(\psi) = \mathbf{V} \\ \neg\psi & \text{se } v(\psi) = \mathbf{F} \end{cases}$$

Lema 1.5. *Seja ϕ uma fórmula cujas variáveis proposicionais são $q_1, \ldots, q_n$, e seja v uma atribuição de valores às variáveis. Então*

$$q_1^v, \ldots, q_n^v \vdash \phi^v$$

Por exemplo, seja $p \land q$, $v(p) = \mathbf{V}$ e $v(q) = \mathbf{F}$. Temos que $p^v = p$, $q^v = \neg q$ e $(p \land q)^v = \neg(p \land q)$. Então pelo lema, vem que $p, \neg q \vdash \neg(p \land q)$.

Demonstração. Por indução estrutural na fórmula ϕ (= no número de conectivas que ocorrem em ϕ):

$\phi = q_1$ Então é claro que $q_1^v \vdash q_1^v$

$\phi = \neg\phi_1$ e tem-se que $q_1^v, \ldots, q_n^v \vdash \phi_1^v$ por hipótese de indução. Se $v(\phi) = \mathbf{V}$, então $v(\phi_1) = \mathbf{F}$, donde $\phi^v = \neg\phi_1 = \phi_1^v$ e portanto $q_1^v, \ldots, q_n^v \vdash \phi^v$. Caso contrário, $v(\phi) = \mathbf{F}$, então $v(\phi_1) = \mathbf{V}$, então $\phi_1^v = \phi_1$ e $\phi^v = \neg\neg\phi_1$. Podemos estender a dedução de $q_1^v, \ldots, q_n^v \vdash \phi_1$ usando a regra $\neg\neg\mathbf{I}$ e obtemos uma dedução $q_1^v, \ldots, q_n^v \vdash \neg\neg\phi_1 = \phi^v$.

$\phi = \phi_1 \circ \phi_2$ onde $\circ$ pode ser $\land$, $\lor$ ou $\to$. Sejam $p_1, \ldots, p_l$ e $r_1, \ldots, r_k$, respectivamente as variáveis proposicionais que ocorrem em ϕ_1 e ϕ_2, e $\{q_1, \ldots, q_n\} = \{p_1, \ldots, p_l\} \cup \{r_1, \ldots, r_k\}$. De $p_1^v, \ldots, p_l^v \vdash \phi_1^v$ e $r_1^v, \ldots, r_k^v \vdash \phi_2^v$ podemos deduzir, usando a regra $\land\mathbf{I}$:

$$q_1^v, \ldots, q_n^v \vdash \phi_1^v \land \phi_2^v$$

donde temos de deduzir ϕ^v.

$\phi = \phi_1 \to \phi_2$ Se $v(\phi) = \mathbf{F}$, então $v(\phi_1) = \mathbf{V}$ e $v(\phi_2) = \mathbf{F}$. Então $\phi_1^v = \phi_1$ e $\phi_2^v = \neg\phi_2$, e $q_1^v, \ldots, q_n^v \vdash \phi_1 \wedge \neg\phi_2$. Para ter $q_1^v, \ldots, q_n^v \vdash \neg(\phi_1 \to \phi_2)$, basta que $\phi_1 \wedge \neg\phi_2 \vdash \neg(\phi_1 \to \phi_2)$ (mostra!).

Se $v(\phi) = \mathbf{V}$, temos 3 casos:

- $v(\phi_1) = v(\phi_2) = \mathbf{F}$. Então $\phi_1^v = \neg\phi_1$ e $\phi_2^v = \neg\phi_2$, e $q_1^v, \ldots, q_n^v \vdash \neg\phi_1 \wedge \neg\phi_2$. Para ter $q_1^v, \ldots, q_n^v \vdash \phi_1 \to \phi_2$, basta que $\neg\phi_1 \wedge \neg\phi_2 \vdash \phi_1 \to \phi_2$ (mostra!).
- $v(\phi_1) = \mathbf{F}$ e$v(\phi_2) = \mathbf{V}$. Então $\phi_1^v = \neg\phi_1$ e $\phi_2^v = \phi_2$, e $q_1^v, \ldots, q_n^v \vdash \neg\phi_1 \wedge \phi_2$. Para ter $q_1^v, \ldots, q_n^v \vdash \phi_1 \to \phi_2$, basta que $\neg\phi_1 \wedge \phi_2 \vdash \phi_1 \to \phi_2$ (mostra!)
- $v(\phi_1) = v(\phi_2) = \mathbf{V}$. Então $\phi_1^v = \phi_1$ e $\phi_2^v = \phi_2$, e $q_1^v, \ldots, q_n^v \vdash \phi_1 \wedge \phi_2$. Para ter $q_1^v, \ldots, q_n^v \vdash \phi_1 \to \phi_2$, basta que $\phi_1 \wedge \phi_2 \vdash \phi_1 \to \phi_2$ (mostra!)

$\phi = \phi_1 \wedge \phi_2$ Temos 4 casos:

- $v(\phi_1) = v(\phi_2) = \mathbf{V}$. Então $\phi_1^v = \phi_1$ e $\phi_2^v = \phi_2$, e $q_1^v, \ldots, q_n^v \vdash \phi_1 \wedge \phi_2$ e então $q_1^v, \ldots, q_n^v \vdash \phi_1 \wedge \phi_2 = \phi^v$
- $v(\phi_1) = \mathbf{F}$ e$v(\phi_2) = \mathbf{V}$. Então $\phi_1^v = \neg\phi_1$ e $\phi_2^v = \phi_2$, e $q_1^v, \ldots, q_n^v \vdash \neg\phi_1 \wedge \phi_2$. Para que $q_1^v, \ldots, q_n^v \vdash \neg(\phi_1 \wedge \phi_2) = \phi^v$ basta que $\neg\phi_1 \wedge \phi_2 \vdash \neg(\phi_1 \wedge \phi_2)$ (mostra!)
- $v(\phi_1) = \mathbf{V}$ e$v(\phi_2) = \mathbf{F}$. Então $\phi_1^v = \phi_1$ e $\phi_2^v = \neg\phi_2$, e $q_1^v, \ldots, q_n^v \vdash \phi_1 \wedge \neg\phi_2$. Para que $q_1^v, \ldots, q_n^v \vdash \neg(\phi_1 \wedge \phi_2) = \phi^v$ basta que $\phi_1 \wedge \neg\phi_2 \vdash \neg(\phi_1 \wedge \phi_2)$ (mostra!)
- $v(\phi_1) = v(\phi_2) = \mathbf{F}$. Então $\phi_1^v = \neg\phi_1$ e $\phi_2^v = \neg\phi_2$, e $q_1^v, \ldots, q_n^v \vdash \neg\phi_1 \wedge \neg\phi_2$. Para o $q_1^v, \ldots, q_n^v \vdash \neg(\phi_1 \wedge \phi_2) = \phi^v$ basta que $\neg\phi_1 \wedge \neg\phi_2 \vdash \neg(\phi_1 \wedge \phi_2)$ (mostra!)

$\phi = \phi_1 \vee \phi_2$ Temos 4 casos:

- $v(\phi_1) = v(\phi_2) = \mathbf{F}$. Então $\phi_1^v = \neg\phi_1$ e $\phi_2^v = \neg\phi_2$, e $q_1^v, \ldots, q_n^v \vdash \neg\phi_1 \wedge \neg\phi_2$. Para que $q_1^v, \ldots, q_n^v \vdash \neg(\phi_1 \vee \phi_2) = \phi^v$ basta que $\neg\phi_1 \wedge \neg\phi_2 \vdash \neg(\phi_1 \vee \phi_2)$ (mostra!)
- $v(\phi_1) = v(\phi_2) = \mathbf{V}$. Então $\phi_1^v = \phi_1$ e $\phi_2^v = \phi_2$, e $q_1^v, \ldots, q_n^v \vdash \phi_1 \wedge \phi_2$. Para que $q_1^v, \ldots, q_n^v \vdash \phi_1 \vee \phi_2 = \phi^v$ basta que $\phi_1 \wedge \phi_2 \vdash \phi_1 \vee \phi_2$ (mostra!)
- $v(\phi_1) = \mathbf{F}$ e $v(\phi_2) = \mathbf{V}$. Então $\phi_1^v = \neg\phi_1$ e $\phi_2^v = \phi_2$, e $q_1^v, \ldots, q_n^v \vdash \neg\phi_1 \wedge \phi_2$. Para que $q_1^v, \ldots, q_n^v \vdash \phi_1 \vee \phi_2 = \phi^v$ basta que $\neg\phi_1 \wedge \phi_2 \vdash \phi_1 \vee \phi_2$ (mostra!)
- $v(\phi_1) = \mathbf{V}$ e $v(\phi_2) = \mathbf{F}$. Então $\phi_1^v = \phi_1$ e $\phi_2^v = \neg\phi_2$, e $q_1^v, \ldots, q_n^v \vdash \phi_1 \wedge \neg\phi_2$. Para o $q_1^v, \ldots, q_n^v \vdash \phi_1 \vee \phi_2 = \phi^v$ basta que $\neg\phi_1 \wedge \neg\phi_2 \vdash \phi_1 \vee \phi_2$ (mostra!)

$\square$

Proposição 1.8. *Se* $\models \phi$ *então* $\vdash \phi$*, i.e, se* ϕ *é uma tautologia então* ϕ *é um teorema.*

Proposição 1.9. *Se* $\phi_1, \ldots, \phi_n \models \phi$ *então* $\phi_1, \ldots, \phi_n \vdash \phi$*.*

Demonstração. **Prop.1.8**
Seja ϕ uma tautologia e $p_1, \ldots, p_n$ as variáveis proposicionais que nela ocorrem. De $\models \phi$ e pelo lema 1.5, $v(\phi) = \mathbf{V}$ e $p_1^v, \ldots, p_n^v \vdash \phi$, para toda a valorização v. Tem-se então que:

$$p_1^v, \ldots, p_{n-1}^v, p_n \vdash \phi \text{ e } p_1^v, \ldots, p_{n-1}^v, \neg p_n \vdash \phi$$

para toda a valorização v. Usando **TE** para $p_n \vee \neg p_n$ e a regra $\vee$**E**, podemos combinar as duas deduções anteriores numa de:

$$p_1^v, \ldots, p_{n-1}^v \vdash \phi$$

Esquematicamente temos:

$$
\begin{array}{|l}
p_1^v \\
\vdots \\
p_n \\ \hline
\boxed{\text{A}} \\
\phi
\end{array}
\qquad
\begin{array}{|l}
p_1^v \\
\vdots \\
\neg p_n \\ \hline
\boxed{\text{B}} \\
\phi
\end{array}
\qquad
\begin{array}{|ll}
p_1^v & \\
\vdots & \\
p_{n-1}^v & \\ \hline
p_n \vee \neg p_n & \quad \textbf{TE} \\
\quad \begin{array}{|l} p_n \\ \hline \boxed{\text{A}} \\ \phi \end{array} & \\
\quad \begin{array}{|l} \neg p_n \\ \hline \boxed{\text{B}} \\ \phi \end{array} & \\
\phi & \quad \vee\text{E}
\end{array}
$$

$$p_1^v, \ldots, p_{n-1}^v, p_n \vdash \phi \qquad p_1^v, \ldots, p_{n-1}^v, \neg p_n \vdash \phi \qquad p_1^v, \ldots, p_{n-1}^v \vdash \phi$$

Repetindo o processo $n-1$ vezes para $p_1, \ldots, p_{n-1}$ obtemos $\vdash \phi$ $\square$

Lema 1.6. $\phi_1, \ldots, \phi_n \models \psi$ *se e só se* $\models (\phi_1 \to (\phi_2 \to (\ldots(\phi_n \to \psi)\ldots)))$*.*

Demonstração. Por contradição. Para qualquer valorização $v((\phi_1 \to (\phi_2 \to (\ldots(\phi_n \to \psi)\ldots)))) = \mathbf{F}$ se $v(\phi_i) = \mathbf{V}$ para todos $1 \leq i \leq n$ e $v(\psi) = \mathbf{F}$ (mostra!). Mas isto contradiz $\phi_1, \ldots, \phi_n \models \psi$! $\square$

Demonstração. (**Prop. 1.9**) Pelo lema 1.6, $\models (\phi_1 \to (\phi_2 \to (\ldots(\phi_n \to \psi)\ldots)))$. Pela proposição 1.8, $\vdash (\phi_1 \to (\phi_2 \to (\ldots(\phi_n \to \psi)\ldots)))$. E pelo lema 1.4 (da dedução, página 33), $\phi_1, \ldots, \phi_n \vdash \psi$ $\square$

O resultado anterior também se verifica para o caso Σ ser infinito e portanto juntando a integridade e a completude, temos para qualquer conjunto de fórmulas:

$$\Sigma \models \phi \quad \text{se e só se} \quad \Sigma \vdash \phi$$

Leituras suplementares [BE00] (Cap. 8.3)

1.6 Decidibilidade da lógica proposicional

Proposição 1.10. *Dado um conjunto de fórmulas Σ e uma fórmula ϕ é decidível se $\Sigma \vdash \phi$, i.e, existe um algoritmo que determina se ϕ é dedutível de Σ.*

Demonstração. Pela completude e integridade, decidir $\Sigma \vdash \phi$ equivale a decidir se $\Sigma \models \phi$. E podemos supor $\Sigma = \emptyset$ (usando a lema 1.4(da dedução)). Então basta construir a tabela de verdade para ϕ (que é finita) e verificar se ϕ é uma tautologia. □

Corolário 1.4. *É decidível se uma fórmula ϕ é um teorema (= é válida).*

Corolário 1.5. *É decidível se uma fórmula ϕ é satisfazível.*

Demonstração. A fórmula ϕ é satisfazível se e só se $\neg\phi$ não é uma tautologia. □

No entanto, utilizar o método das tabelas de verdade para determinar se uma fórmula é satisfazível é um algoritmo pouco eficiente que tem complexidade temporal exponencial. Mas, embora, existam outros algoritmos, até ao momento não se conhece nenhum com complexidade temporal polinomial (este facto está relacionado com a conjectura $P = NP$, como poderão ver em disciplinas de Complexidade).

1.7 Outros sistemas dedutivos

1.7.1 Sistemas dedutivos de Hilbert, H

Usados inicialmente nas tentativas de mecanização das demonstrações matemáticas (Séc. XIX e início de XX) (também usados por Peano, G. Frege e B. Russel)

Supondo apenas o conjunto completo de conectivas $\{\neg, \rightarrow\}$, pode tomar a forma:

Axiomas

- $\phi \rightarrow (\psi \rightarrow \phi)$
- $(\phi \rightarrow (\psi \rightarrow \theta)) \rightarrow ((\phi \rightarrow \psi) \rightarrow (\phi \rightarrow \theta))$

- $(\neg\psi \to \neg\phi) \to ((\neg\psi \to \phi) \to \psi)$

Regras de inferência

- *Modus ponens*: de ϕ e de $\phi \to \psi$, inferir ψ

Proposição 1.11. *(da dedução) Se* $\Sigma \cup \{\psi\} \vdash_H \theta$ *então* $\Sigma \vdash_H \psi \to \theta$.

Proposição 1.12. $\Sigma \vdash_{ND} \phi$ *se e só se* $\Sigma \vdash_H \phi$

Demonstração. ($\Leftarrow$): Basta ver que os axiomas de H são teoremas de DN (ver exercício 1.18). A regra de inferência corresponde à regra da eliminação de implicação de DN ($\to$ **E**)
($\Rightarrow$): é possível transformar uma dedução em DN, numa dedução em H. (Não o faremos neste curso...) □

Corolário 1.6. *O sistema de dedução H é integro e completo para a lógica proposicional.*

A diferença entre os dois sistemas é a impossibilidade de se fazer novas suposições (assumir novas premissas) no sistema H: isso torna as deduções neste sistema mais difíceis e menos automatizáveis...

1.7.2 Sistemas dedutivos analíticos

Mesmo no sistema DN não é fácil construir um algoritmo para determinar se uma fórmula é um teorema, excepto se se usar a completude (e construir a dedução a partir da tabela de verdade). Mas existem outros sistemas de dedução automatizáveis e em que para determinar se uma fórmula ϕ é um teorema não é necessária a semântica. Estes sistemas dizem-se *analíticos* (ou automatizáveis). A propriedade essencial destes sistemas é que, partindo da fórmula que se pretende deduzir (i.e considerando a dedução da conclusão para as premissas), em cada passo duma dedução, as fórmulas são sub-fórmulas de alguma fórmula de um passo "anterior". É fácil ver que a regra de *Modus ponens* não verifica esta propriedade.
Entre os sistemas analíticos, destacamos:

***Sequentes* de Gentzen** Variante do sistema de dedução natural.

***Tableaux* analíticos (ou semânticos)** Para deduzir ϕ, deduz-se que a fórmula $\neg\phi$ é uma contradição.

Resolução também por contradição, mas necessita de $\neg\phi$ em forma normal conjuntiva.

Um sistema dedutivo diz-se de *refutação* se para deduzir ϕ de Σ, se deduz **F** de $\Sigma \cup \{\neg\phi\}$. A consequência semântica é preservada: se $\Sigma \models \phi$ sse $\Sigma \cup \{\neg\phi\}$ não é satisfazível. Tanto os *tableaux* como a resolução são sistemas de refutação.

1.7.3 *Tableaux* semânticos

Para deduzir uma fórmula ϕ, inicia-se com $\neg\phi$ e produz-se uma contradição. Uma dedução é uma árvore em que os nós são etiquetados por fórmulas, sendo a raíz da árvore etiquetada por $\neg\phi$. Em cada passo, expande-se uma folha da árvore de acordo com as regras de expansão dos *Tableaux* (Tabela 1.2). Nestas regras utiliza-se a *notação uniforme* (de R.M. Smullyan [Smu95]), que permite agrupar as fórmulas da forma $\phi \circ \psi$ e $\neg(\phi \circ \psi)$ em duas categorias: as *conjuntivas*, α, e as *disjuntivas*, β. Para cada fórmula α (ou β), associam-se duas componentes que denotamos por α_1 e α_2 (β_1 ou β_2). Na Tabela 1.1 apresenta-se os valores destas componentes para $\circ \in \{\wedge, \vee, \rightarrow\}$.

α	α_1	α_2	β	β_1	β_2
$\phi \wedge \psi$	ϕ	ψ	$\neg(\phi \wedge \psi)$	$\neg\phi$	$\neg\psi$
$\neg(\phi \vee \psi)$	$\neg\psi$	$\neg\phi$	$\phi \vee \psi$	ϕ	ψ
$\neg(\phi \rightarrow \psi)$	ϕ	$\neg\psi$	$\phi \rightarrow \psi$	$\neg\phi$	ψ

Tabela 1.1: Notação uniforme: fórmulas α e β

É fácil ver que:

Proposição 1.13. *Para toda a atribuição de valores às variáveis v, e para todas as fórmulas α e β:*

$$
\begin{aligned}
v(\alpha) &= v(\alpha_1) \wedge v(\alpha_2) \\
v(\beta) &= v(\beta_1) \vee v(\beta_2).
\end{aligned}
$$

As regras de expansão de *tableaux* permitem transformar uma árvore $\mathbf{T}$ etiquetada por fórmulas, noutra árvore $\mathbf{T}^\star$. Suponhamos que num ramo r da árvore ocorre uma fórmula não literal ψ. Se ψ é $\neg\neg\phi$, podemos expandir a folha desse ramo com mais um nó etiquetado por ϕ. Análogamente se ψ é $\neg\mathbf{F}$, adiciona-se $\mathbf{V}$ ou se é $\neg\mathbf{V}$, adiciona-se $\mathbf{F}$. Se ψ é um α, adicionam-se a r dois nós, um etiquetado com α_1 e outro com α_2 (filho de α_1). Se ψ é um β, adiciona-se à folha de r dois filhos, um etiquetado por β_1 outro por β_2.

$$
\frac{\neg\neg\phi}{\phi} \qquad \frac{\neg\mathbf{F}}{\mathbf{V}} \qquad \frac{\neg\mathbf{V}}{\mathbf{F}} \qquad \frac{\alpha}{\begin{array}{c}\alpha_1 \\ \alpha_2\end{array}} \qquad \frac{\beta}{\beta_1 \mid \beta_2}
$$

Tabela 1.2: Regras de expansão dos *tableaux*

Um *tableau* pode ser definido indutivamente por:

Definição 1.16. *Seja* $\{\phi_1, \ldots, \phi_n\}$ *um conjunto de fórmulas.*

1. *Um* tableau *para* $\{\phi_1, \ldots, \phi_n\}$ *é uma árvore de um só ramo:*

$$\begin{array}{c} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_n \end{array}$$

2. *Se* $\mathbf{T}$ *é um* tableau *para* $\{\phi_1, \ldots, \phi_n\}$ *e* $\mathbf{T}^\star$ *resulta de* `T` *por aplicação duma regra de expansão de* tableaux*, então* $\mathbf{T}^\star$ *é um* tableau *para* $\{\phi_1, \ldots, \phi_n\}$.

Um *tableau* para o conjunto $\{p \wedge (\neg q \vee \neg p)\}$ é:

$$\begin{array}{ccc} & p \wedge (\neg q \vee \neg p) & \\ & p & \\ & \neg q \vee \neg p & \\ \neg q & & \neg p \end{array}$$

Definição 1.17. *Um ramo* r *de um* tableau *diz-se* fechado *se existe uma fórmula* ϕ *tal que* ϕ *e* $\neg\phi$ *ocorrem em* r. *Um* tableau *diz-se* fechado *se todos os seus ramos estão fechados.*

O *tableau* do exemplo anterior tem, um ramo fechado mas não é fechado.

Definição 1.18. *Uma dedução por* tableau *de* ϕ *é um* tableau *fechado para* $\{\neg\phi\}$. *A fórmula* ϕ *é um teorema se* ϕ *tem uma dedução por* tableau.

Definição 1.19. *Um ramo* r *de um* tableau *é* satisfazível *se o conjunto de fórmulas que etiquetam os seus nós é satisfazível. Um* tableau $\mathbf{T}$ *é satisfazível se pelo menos um dos seus ramos é satisfazível.*

A proposição seguinte demonstra-se por análise de casos e usando a Proposição 1.13.

Proposição 1.14. *Pela aplicação de qualquer regra de expansão de* tableau *a um* tableau *satisfazível, obtêm-se um* tableau *satisfazível.*

Proposição 1.15. *Se existe um* tableau *fechado para um conjunto de fórmulas* Γ*, então* Γ *não é satisfazível.*

Temos, então, a integridade do sistema de *tableaux*:

Teorema 1.1. *Se* ϕ *tem uma dedução por* tableau*, então* ϕ *é uma tautologia.*

Demonstração. Uma dedução por *tableau* é um *tableau* fechado para $\{\neg\phi\}$. Então, pela Proposição 1.15, $\{\neg\phi\}$ não é satisfazível, logo ϕ é uma tautologia. □

Apresentamos, sem demonstração, a completude do sistema de *tableaux*:

Teorema 1.2. *Se ϕ é uma tautologia, ϕ tem uma dedução por* tableau.

Exercício 1.20. *Constrói deduções de* tableaux *para as seguintes fórmulas:*

a) $\neg((p \lor q) \land (\neg p \land \neg q))$

b) $((p \to q) \land (q \to r)) \to \neg(\neg r \land p)$

c) $(\neg p \to q) \to ((p \to q) \to q)$

d) $((p \to q) \to p) \to p$

◇

1.7.4 Resolução

Recorda que um *literal* é uma fórmula atómica ou a sua negação: p, $\neg p$. Uma *cláusula* é uma disjunção de literais: $p \lor \neg q \lor \neg p \lor s$ e pode representar-se pelo conjunto $\{p, \neg q, \neg p, s\}$. Então uma fórmula da lógica proposicional em FNC, p.e.

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

pode ser vista como um conjunto de cláusulas:

$$\{\{\neg p\}, \{q, r\}, \{\neg r, \neg s\}, \{p, s\}, \{\neg q, \neg s\}\}$$

Seja Σ um conjunto de cláusulas e representamos por $\mathbf{F}$ a cláusula vazia.
O sistema dedutivo por *resolução* não tem axiomas e apenas uma regra de inferência (de *resolução*):

$$\frac{C \cup \{p\} \quad C' \cup \{\neg p\}}{C \cup C'}$$

e a conclusão diz-se a *resolvente* das premissas.
A dedução de $\mathbf{F}$ a partir de um conjunto $\mathcal{C}$ de cláusulas diz-se uma refutação $\mathcal{C}$.
No caso anterior, tinha-se a seguinte dedução:

$$\dfrac{\dfrac{\{p,s\} \quad \{\neg p\}}{\{s\}} \qquad \dfrac{\dfrac{\{q,r\} \quad \{\neg r,\neg s\}}{\{q,\neg s\}} \quad \{\neg q,\neg s\}}{\{\neg s\}}}{\mathbf{F}}$$

$$\dfrac{\dfrac{\{p,s\}\ \{\neg p\}}{\{s\}} \quad \dfrac{\dfrac{\{q,r\}\ \{\neg r,\neg s\}}{\{q,\neg s\}}\ \{\neg q,\neg s\}}{\{\neg s\}}}{\mathbf{F}}$$

Uma *dedução por resolução de* uma fórmula ϕ é uma refutação de cláusulas que correspondem à FNC de $\neg\phi$.

Teorema 1.3. *(Integridade) Seja $\mathcal{C} = \{C_1, \ldots, C_n\}$ um conjunto não vazio de cláusulas da lógica proposicional.*

i. Sejam C_i e C_j cláusulas de $\mathcal{C}$ e R uma resolvente de C_i e C_j. Então $\mathcal{C} \models R$.

ii. Se $\mathcal{C} \vdash_R \boldsymbol{F}$, isto é, existe uma dedução de $\boldsymbol{F}$ a partir de $\mathcal{C}$ usando apenas a regra da Resolução, então $\mathcal{C}$ não é satisfazível.

Exercício 1.21. *Mostra o teorema 1.3.* $\diamond$

Teorema 1.4. *(Completude) Se um conjunto de cláusulas C é não satisfazível então existe uma dedução $\mathcal{C} \vdash_R \boldsymbol{F}$.*

Exercício 1.22. *Considera a fórmula seguinte em forma normal conjuntiva (FNC):*

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

i. Converte a fórmula para um conjunto de cláusulas.

ii. A partir desse conjunto constrói uma dedução de $\boldsymbol{F}$,q usando apenas a regra da resolução.

$\diamond$

Exercício 1.23. *Encontra uma dedução de $\boldsymbol{F}$ (uma refutação) para:*

$$\{\{\neg p, \neg q, r\}, \{p, r\}, \{q, r\}, \{\neg r\}\}$$

$\diamond$

Exercício 1.24. *Encontra uma dedução por resolução de:* $((p \to q) \ \wedge \ (q \to r)) \to \neg(\neg r \ \wedge \ p)$

$\diamond$

Exercício 1.25. *Constrói deduções de* tableau *e por* ***resolução*** *para as seguintes fórmulas:*

a) $(\psi \to \phi) \to ((\neg\psi \to \phi) \to \phi)$

b) $\phi \to (\psi \to (\phi \wedge \psi))$

c) $\phi \to (\neg\psi \to \neg(\phi \to \psi))$

$\diamond$

Leituras suplementares [Bro00], [Fit90], [Smu95],[GLM97], [BA01].

Capítulo 2

Lógica de primeira ordem

Na lógica proposicional não é possível representar adequadamente frases como:

- Todos os pássaros têm penas
- Alguns pássaros cantam
- Nem todos os pássaros voam
- Todos os homens são mortais
- Todas as crianças são mais novas que os seus pais
- Um inteiro ou é par ou ímpar
- Todos os pássaros são mais leves que algum mamífero

Por outro lado, proposições como as seguintes podem-se representar usando relações n-árias entre objectos:

Proposição	Representação
O Carlos é irmão da Joana	`irmao(carlos,joana)`
A Joana deu um livro ao Carlos	`dar(joana,livro,carlos)`
2 é par	`par(2)`
$2 = 1 + 1$	`igual(2,soma(1,1))`

Na lógica de primeira ordem vai ser possível representar:

Objectos: Termos que podem ser simples (constantes ou variáveis) ou complexos. Ex: `joana, x, mãe(joana), soma(1,1)`

Propriedades ou relações entre objectos: Fórmulas que podem ser predicados simples ou complexos (usando conectivas). Ex: `dar(joana,livro,carlos)`,$\texttt{par}(2) \wedge \neg\texttt{par}(3)$

Propriedades ou relações entre conjuntos de objectos Fórmulas quantificadas sobre objectos. Os quantificadores permitem representar as palavras `todos`, `alguns`, etc:

$\forall$ (universal) e $\exists$ (existêncial).

Ex:

$\forall \texttt{x}\, (\texttt{passaro}(\texttt{x}) \rightarrow \texttt{penas}(\texttt{x}))$

$\exists \texttt{x}\, (\texttt{passaro}(\texttt{x}) \wedge \texttt{canta}(\texttt{x}))$

$\exists \texttt{x}\, (\texttt{passaro}(\texttt{x}) \wedge \neg\texttt{voa}(\texttt{x}))$

$\forall \texttt{x}\, (\texttt{passaro}(\texttt{x}) \rightarrow (\exists \texttt{y}(\texttt{mamifero}(\texttt{y}) \wedge \texttt{mais\_leve}(\texttt{x}, \texttt{y}))))$

2.1 Linguagens da lógica de primeira ordem

Uma linguagem $\mathcal{L}$ de lógica de 1ª ordem é caracterizada pelos seguintes conjuntos de símbolos:

Símbolos lógicos que estão presentes em qualquer linguagem:

- um conjunto numerável de *variáveis*, $Var = \{x, y, \ldots, x_0, y_0, \ldots\}$;
- as *conectivas lógicas* $\wedge$, $\vee$, $\neg$ e $\rightarrow$;
- os *quantificadores* $\forall$ (universal) e $\exists$ (existencial);
- os parêntesis (e);
- possivelmente o *símbolo de igualdade*, $=$.

Símbolos não lógicos que caracterizam a linguagem:

- um conjunto, possivelmente vazio, de *símbolos funcionais* n-ários para cada $n \geq 0$, $\mathcal{F}_n$; os elementos de $\mathcal{F}_0$ chamam-se *constantes* e representam-se por a, b, c,.... Os restantes símbolos por f, g, h, ...;
- um conjunto, possivelmente vazio, de *símbolos de predicado* (ou relacionais) n-ários para cada $n \geq 0$, $\mathcal{R}_n$. Os símbolos relacionais representam-se por P, R, Q, etc...

Definição 2.1 (Termos). *Seja $\mathcal{L}$ uma linguagem de 1ª ordem. Um* termo*, é uma sequência de símbolos de $\mathcal{L}$, representado por $t, s, \ldots t_1, s_1, \ldots$ e definido indutivamente por:*

- *Uma variável x é um termo;*
- *Uma constante a é um termo;*

- *Se* $t_1, \ \ldots, \ t_n$ *são termos e* $f \in \mathcal{F}_n$ *um símbolo funcional de aridade* $n > 0$*, então* $f(t_1, \ldots, t_n)$ *é um termo .*

O conjunto dos termos $\mathcal{T}_\mathcal{L}$ também pode ser definido pela seguinte gramática independente de contexto, em notação BNF:

$$t ::= x \mid a \mid f(t, \ldots, t)$$

onde $x \in Var$, $a \in \mathcal{F}_0$ e $f \in \mathcal{F}_n$.
Um termo diz-se *fechado* se não tiver ocorrências de variáveis e denotamos por $\mathcal{T}_O$ o seu conjunto. Ex: a, $f(a, g(b, c))$ e $f(f(a, a)))$

Exemplo 2.1. *Supondo* $\mathcal{F}_0 = \{a, d\}$*,* $\mathcal{F}_2 = \{f, h\}$*,* $\mathcal{F}_1 = \mathcal{F}_3 = \{g\}$*, indica quais das seguintes sequências de símbolos são termos e quais são termos fechados:*

- $f(a, g(x, g(a), a))$
- $g(d, h(y, z), d, f(a)))$
- $h(d, f(a, h(d)))$
- $f(f(x, x), f(y, y))$
- $x(d, g(y))$
- $f(a, a(x))$

Definição 2.2 (Fórmulas)**.** *O conjunto de* fórmulas atómicas *duma linguagem* $\mathcal{L}$ *é dado por:*

- *se* $t_1, \ldots, t_n$ *são termos e* $R \in \mathcal{R}_n$ *é um símbolo de predicado n-ário, então* $R(t_1, \ldots, t_n)$ *é uma* fórmula atómica*;*

- *se* $\mathcal{L}$ *tiver a igualdade e se* t_1 *e* t_2 *são termos então* $t_1 = t_2$ *é uma* fórmula atómica*;*

O conjunto das fórmulas *é definido indutivamente por:*

- *uma fórmula atómica é uma* fórmula*;*
- *se* ϕ *é uma fórmula* $\neg\phi$ *é uma* fórmula*;*
- *se* ϕ *e* ψ *são fórmulas então* $(\phi \ \wedge \ \psi)$*,* $(\phi \ \vee \ \psi)$ *e* $(\phi \rightarrow \psi)$ *são* fórmulas
- *se* ϕ *é uma fórmula e* x *uma variável, então* $\forall x \ \phi$ *e* $\exists x \ \phi$ *são* fórmulas.

O conjunto das fórmulas pode também ser definido por uma gramática, em notação BNF:

$$\phi ::= R(t_1, \ldots, t_n) \mid t_1 = t_2 \mid \neg\phi \mid (\phi \ \wedge \ \phi) \mid (\phi \ \vee \ \phi) \mid (\phi \rightarrow \phi) \mid \forall x\phi \mid \exists x\phi$$

onde $R \in \mathcal{R}_n$, $t_i \in \mathcal{T}$ e $x \in Var$.

Exemplo 2.2. *Sendo* $\mathcal{F}_0 = \{m, d\}$, $\mathcal{F}_1 = \{f\}$, $\mathcal{F}_2 = \{g\}$ *e* $\mathcal{R}_2 = \{R, S\}$, *determina quais das seguintes expressões são fórmulas:*

- $S(m, x)$
- $f(m)$
- $R(R(m, x))$
- $(R(x, y) \rightarrow (\exists z\ S(z, f(y))))$
- $R(x, y) \rightarrow R(\exists z R(z, z)))$
- $\forall x\ R(f(d), g(f(m), y))$
- $\forall x \forall y (g(x, y) \rightarrow S(y, x))$

Nas fórmulas da lógica de $1^{\underline{a}}$ ordem, podem-se omitir os parêntesis, supondo que as convenções para a lógica proposicional e que os quantificadores têm maior precedência que qualquer operador lógico. Por exemplo, as seguintes fórmulas são diferentes: $\exists x P(x) \ \vee \ \neg P(x)$ e $\exists x (P(x) \ \vee \ \neg P(x))$.

Exemplo 2.3 (Linguagem para a aritmética)**.** *Pretende-se uma linguagem para exprimir ir fórmulas sobre os números naturais e as operações de adição e multiplicação. Seja* $\mathcal{A}$ *a linguagem com* $=$ *caracterizada por:*

$$\mathcal{F}_0 = \{0, 1\},\ \mathcal{F}_2 = \{+, \times\}\ e\ \mathcal{R}_2 = \{<\}$$

Os termos seguintes representam os naturais:
$0, 1, +(1, 1), +(1, +(1, 1)), +(1, +(+(1, 1), 1)) \dots$
Outros termos são: $(\times(\times(1, 0), 1),\ (+(0, +(0, 1))),\ \dots$
E exemplos de fórmulas são:
$< (\times(1, 1), +(1, 1))$
$\forall x\ (+(0, x) = x)$
$\forall x \forall y (+(x, 1) = +(y, 1) \rightarrow x = y)$
Podemos ainda adoptar a notação usual infixa para os símbolos funcionais e relacionais, e escrever: $(1 \times 1) < (1 + 1)$, $(0 + x)$, *etc...*

Exemplo 2.4 (Tradução de frases para fórmulas)**.** *Na tabela seguinte apresentam-se algumas frases em linguagem natural e a sua representação como fórmulas da lógica de* $1^{\underline{a}}$ *ordem.*

Frases do tipo	*Representação*
Todos os P's são Q's	$\forall x (P(x) \rightarrow Q(x))$
Alguns P's são Q's	$\exists x (P(x) \ \wedge \ Q(x))$
Nenhum P é Q	$\forall x (P(x) \rightarrow \neg Q(x))$
Nem todos os P's são Q's	$\exists x (P(x) \ \wedge \ \neg Q(x))$

Considerando predicados adicionais, exprime na linguagem da aritmética:

- *Todo o número par é primo*
- *Nem todos os números primos são pares*
- *Alguns primos não são pares*
- *Nenhum primo é par*
- *Todo o primo é não par ou igual a 2*

Definição 2.3 (Variáveis livres e ligadas)**.** *Uma variável x tem uma ocorrência* não livre *(ou ligada) numa fórmula ϕ se ϕ tem uma sub-fórmula da forma $\forall x\psi$ ou $\exists x\psi$ e x ocorre em ψ. Caso contrário, diz-se que x* ocorre livre *em ϕ.*

Uma variável pode ocorrer livre e ligada numa mesma fórmula. Em

$$\forall x((P(x) \to Q(x))) \to (\neg P(x) \lor Q(y))$$

x tem duas ocorrências ligadas e uma ocorrência livre e y tem apenas uma ocorrência livre.

Definição 2.4 (Proposição)**.** *Uma fórmula sem variáveis livres diz-se uma* proposição*.*

Por exemplo, $R(a, f(a, b)) \land \forall x(R(x, x) \to \exists y P(y) \lor P(x))$ é uma proposição.

Exemplo 2.5. *Indica quais as ocorrências livres e ligadas de cada uma das variáveis das fórmulas seguintes. Indica quais as fórmulas que são proposições.*

- $\exists x(P(y, z) \land \forall y(\neg Q(x, y) \lor P(y, z)))$;
- $\neg(\forall x \exists y P(x, y, z) \land \forall z P(x, y, z))$;
- $\forall x((P(x) \land C(x)) \to \exists y L(x, y))$;
- $P(a, f(a, b))$;
- $\exists x(P(x) \to \neg Q(x))$.

2.2 Semântica da lógica de 1ª ordem

Na lógica proposicional a semântica duma fórmula complexa é determinada a partir da semântica dos seus constituintes básicos, i.e, as proposições $p, q \ldots$ cujo valor pode ser **V** ou **F**. Por exemplo, se o valor de p for **V** e o de q, **F**, então a fórmula seguinte tem o valor **V** (verifica!):

$$(p \lor \neg q) \to (q \to p)$$

Como avaliar uma fórmula de primeira ordem? Por exemplo, como avaliar:

$$\forall x \exists y (P(x) \ \vee \ \neg Q(y)) \rightarrow (Q(x) \rightarrow P(y))$$

Temos que determinar qual o significado

- das variáveis (livres e ligadas)
- dos quantificadores
- dos símbolos funcionais
- dos símbolos relacionais

num dado *universo* ou *domínio de discurso*, por exemplo: os números inteiros, números reais, conjuntos, grafos, objectos geométricos, etc,etc.
Informalmente para os quantificadores e para um dado universo:

$\forall x P(x)$ será verdadeira se e só se $P(x)$ satisfaz *todos* os objectos do universo

$\exists x P(x)$ será verdadeira se e só se $P(x)$ satisfaz *algum* objecto do universo

Exemplo 2.6 (*Mundo dos Blocos*)**.** *Pretende-se descrever um conjunto de blocos com diferentes formas geométricas e tamanhos, colocados num tabuleiro de xadrez...*

Uma linguagem para definir este conjunto pode ser a seguinte:

$$
\begin{aligned}
\mathcal{F}_1 &= \{maisafrente, maisaesquerda, maisadireita\} \\
\mathcal{R}_1 &= \{Cubo, Tetra, Dodec, Pequeno, Medio, Grande\} \\
\mathcal{R}_2 &= \{Menor, Maior, Esquerda, Direita, MesmoT, \\
&\quad MesmaF, MesmaL, MesmaC\} \\
\mathcal{R}_3 &= \{Entre\}
\end{aligned}
$$

Algumas fórmulas:

- $\exists y\ (Cubo(y)\ \wedge\ Medio(y))$
- $\forall x\ (Cubo(x) \rightarrow \forall y(\neg Cubo(y) \rightarrow Maior(x,y)))$
- $\exists y\ maisaesquerda(y) = y$
- $\forall x\ (Pequeno(x) \rightarrow maisafrente(x) = x)$
- $\forall x\ (maisaesquerda(x) \neq x \rightarrow Cubo(maisaesquerda(x)))$
- $Cubo(x)$
- $MesmoT(z,y)\ \wedge\ MesmaF(z,y)$

Um domínio (conjunto de objectos) irá permitir determinar quais as fórmulas que são satisfeitas nesse "mundo"(estrutura). Para além do domínio é necessário indicar também qual o significado de cada símbolo funcional e cada símbolo relacional.
No mundo (estrutura) $\mathcal{A}$ apresentado na figura:

- *o domínio tem 8 objectos $\{o_1, \ldots o_8\}$ (ordenados de cima para baixo e da esquerda pra a direita...)*
- *às constantes não associamos nenhum objecto*
- *a $maisafrente$ associamos uma função $maisafrente^{\mathcal{A}}$ que para cada objecto indica qual o objecto mais à frente na mesma coluna: $maisafrente^{\mathcal{A}}(o_1) = o_6$,$maisafrente^{\mathcal{A}}(o_4) = o_7$, e para os restantes $maisafrente^{\mathcal{A}}(o) = o$. Analogamente associa-se a $maisaesquerda$ e $maisadireita$, a função que para cada objecto indica qual o objecto mais à esquerda, respectivamente, mais à direita.*
- *a $Cubo$ associamos uma relação unária que corresponde aos objectos do domínio que são cubos: $Cubo^{\mathcal{A}} = \{o_4, o_5\}$. Analogamente,*

$Tetra^{\mathcal{A}} = \{o_1, o_2, o_3\}$

$Dodec^{\mathcal{A}} = \{o_6, o_7, o_8\}$.

$Pequeno^{\mathcal{A}} = \{o_6, o_7, o_8\}$

$Grande^{\mathcal{A}} = \{o_2\}$ *e para os restantes* $o \in Medio^{\mathcal{A}}$

$Esquerda^{\mathcal{A}} = \{(o_1, o_2), (o_4, o_5), (o_6, o_7), (o_7, o_8)\}$

$MesmaL^{\mathcal{A}} = (Esquerda^{\mathcal{A}})^{\star}$ *(fecho de reflexivo e transitivo)*

e de modo análogo se definem as restantes relações binárias.

$Entre^{\mathcal{A}} = \{(o_7, o_6, o_8)\}$

Como determinar se uma fórmula é satisfazível nesta estrutura?
$\exists y(Cubo(y) \wedge Medio(y))$ *é satisfazível em* $\mathcal{A}$ *pois* $o_4 \in Cubo^{\mathcal{A}}$ *e* $o_4 \in Medio^{\mathcal{A}}$
$\forall x(Dodec(x) \rightarrow Pequeno(x))$ *é satisfazível em* $\mathcal{A}$ *pois para todo* $o \in Dodec^{\mathcal{A}}$, $o \in Pequeno^{\mathcal{A}}$
Mas, $\forall x(Medio(x) \rightarrow Cubo(x))$ *não é satisfazível em* $\mathcal{A}$ *porque* $o_2 \in Medio^{\mathcal{A}}$ *e* $o_2 \notin Cubo^{\mathcal{A}}$
E, $\forall x\ (Pequeno(x) \rightarrow maisafrente(x) = x)$ *é satisfazível em* $\mathcal{A}$*. Porquê?*
E para as fórmulas como $Cubo(x)$ *ou* $MesmoT(z, y) \wedge MesmaF(z, y)$*?*
Neste caso temos ainda de ter primeiro uma interpretação s *para as variáveis livres...*
Para $s(x) = o_4$*,* $s(y) = o_7$ *e* $s(z) = o_6$*, as duas fórmulas são satisfazíveis, pois* $o_4 \in Cubo^{\mathcal{A}}$ *e* $o_7, o_6 \in MesmoT^{\mathcal{A}} \cap MesmaF^{\mathcal{A}}$

Definição 2.5 (Estrutura duma linguagem)**.** *Uma* estrutura *duma uma linguagem de primeira ordem* $\mathcal{L}$ *é: um par* $\mathcal{A} = (A, \cdot^{\mathcal{A}})$ *onde* A *é um conjunto não vazio, e que se diz o domínio (ou o universo)* $\mathcal{A}$ *e* $.^{\mathcal{A}}$ *uma função tal que:*

- *associa a cada constante* c *um elemento* $c^{\mathcal{A}} \in A$
- *associa a cada símbolo funcional* $f \in \mathcal{F}_n$*,* $n > 0$ *uma função n-ária* $f^{\mathcal{A}}$ *de* A^n *em* A
- *associa a cada símbolo relacional n-ário* $R \in \mathcal{R}_n$*, uma relação* $R^{\mathcal{A}} \subseteq A^n$

Definição 2.6 (Interpretação de variáveis)**.** *Dada uma linguagem* $\mathcal{L}$ *e uma estrutura* $\mathcal{A} = (A, \cdot^{\mathcal{A}})$ *de* $\mathcal{L}$*.*
Uma interpretação *das variáveis é uma função* $s : Var \longrightarrow A$*.*
Podemos estender uma interpretação s *ao conjunto dos termos de* $\mathcal{L}$*,* $s : \mathcal{T} \longrightarrow A$*:*

- *para* $x \in Var$ *o valor de* $s(x)$ *já está definido*
- *para* $c \in \mathcal{F}_0$*,* $s(c) = c^{\mathcal{A}}$
- *se* $t_1, \ldots, t_n$ *são termos e* $f \in \mathcal{F}_n$*,* $n \geq 1$*,*

$$s(f(t_1, \ldots, t_n)) = f^{\mathcal{A}}(s(t_1) \ldots, s(t_n))$$

Se $t \in \mathcal{T}_O$ então $s(t)$ não depende de s e escrevemos $t^{\mathcal{A}}$.

Definição 2.7 (Relação de satisfazibilidade)**.** *Dada uma linguagem $\mathcal{L}$, uma estrutura $\mathcal{A} = (A, \cdot^{\mathcal{A}})$ de $\mathcal{L}$ e uma interpretação $s : Var \longrightarrow A$. A relação $\mathcal{A}$ satisfaz uma fórmula ϕ para a interpretação s, denota-se por $\mathcal{A} \models_s \phi$ é definida por indução na estrutura de ϕ:*

1. $\mathcal{A} \models_s t_1 = t_2$ *se* $s(t_1) = s(t_2)$
2. $\mathcal{A} \models_s R(t_1, \ldots, t_n)$ *se* $(s(t_1), \ldots, s(t_n)) \in R^{\mathcal{A}}$
3. $\mathcal{A} \models_s \neg\phi$ *se* $\mathcal{A} \not\models_s \phi$ *(i.e não é verdade que* $\mathcal{A} \models_s \phi$*)*
4. $\mathcal{A} \models_s \phi \wedge \psi$ *se* $\mathcal{A} \models_s \phi$ *e* $\mathcal{A} \models_s \psi$
5. $\mathcal{A} \models_s \phi \vee \psi$ *se* $\mathcal{A} \models_s \phi$ *ou* $\mathcal{A} \models_s \psi$
6. $\mathcal{A} \models_s \phi \rightarrow \psi$ *se* $\mathcal{A} \not\models_s \phi$ *ou* $\mathcal{A} \models_s \psi$
7. $\mathcal{A} \models_s \forall x\, \phi$ *se para todo o* $a \in A$ *se tem* $\mathcal{A} \models_{s[a/x]} \phi$ *onde:*

$$s[a/x](y) = \begin{cases} s(y) & \textit{se } y \neq x \\ a & \textit{se } y = x \end{cases}$$

8. $\mathcal{A} \models_s \exists x\, \phi$ *se existe um* $a \in A$ *tal que tem* $\mathcal{A} \models_{s[a/x]} \phi$

Exemplo 2.7. *Seja $\mathcal{L}_N$ a linguagem de $1^{\underline{a}}$ ordem com igualdade para os números naturais já dada: $\mathcal{F}_0 = \{0, 1\}$, $\mathcal{F}_2 = \{+, \times\}$ e $\mathcal{R}_2 = \{<\}$*
Seja $\mathcal{N} = (\mathbb{N}, \cdot^{\mathcal{N}})$ uma estrutura de $\mathcal{L}_N$, onde $\cdot^{\mathcal{N}}$ é definido por:

- $0^{\mathcal{N}} = 0$, $1^{\mathcal{N}} = 1$
- $+^{\mathcal{N}}(n, m) = n + m$, $\times^{\mathcal{N}}(n, m) = n \times m$
- $<^{\mathcal{N}} = \{(n, m) \in \mathbb{N}^2 \mid n < m\}$

a) Verificar que, para qualquer interpretação $s : Var \longrightarrow \mathbb{N}$:

$\mathcal{N} \models_s \forall x\ < (x, +(x, 1))$

Por 7 da Definição 2.7, temos que ver se para todo o $n \in \mathbb{N}$ se tem

$$\mathcal{N} \models_{s[n/x]} < (x, +(x, 1))$$

Por 2 temos:

$$(s[n/x](x), s[n/x](+(x, 1)) \in <^{\mathcal{N}}$$

e usando a definição de interpretação (2.6) vem,

$$(n, n+1) \in <^{\mathcal{N}}$$

i.e, $n < n+1$, o que realmente é verdade para todo o $n \in \mathbb{N}$.

b) *Se $s(x) = 2$, verificar que $\mathcal{N} \models_s x = +(1,1)$.*

Neste caso, por 1 da Definição 2.7, temos que ver se

$$s(x) = s(+(1,1))$$

Isto é:

$$2 = +^{\mathcal{N}}(s(1), s(1))$$

ou seja, $2 = 1+1$ o que é verdade em $\mathbb{N}$.

2.2.1 Satisfazibilidade, validade e consequência

Uma fórmula ϕ duma linguagem de 1ª ordem é:

satisfazível: se existir uma estrutura $\mathcal{A}$ e uma interpretação s, tal que $\mathcal{A} \models_s \phi$

válida: se para toda a estrutura $\mathcal{A}$ e toda a interpretação s, se tem $\mathcal{A} \models_s \phi$. E escreve-se $\models \phi$

Definição 2.8. *Uma estrutura $\mathcal{A}$* satisfaz *um conjunto de fórmulas Γ para uma interpretação s, se satisfizer todas as fórmulas de Γ, e escreve-se $\mathcal{A} \models_s \Gamma$.*
Uma fórmula ϕ é consequência semântica *de Γ se para toda a estrutura $\mathcal{A}$ de $\mathcal{L}$ e toda a interpretação s tal que $\mathcal{A} \models_s \Gamma$, se tem $\mathcal{A} \models_s \phi$. E escreve-se $\Gamma \models \phi$.*

Para que uma estrutura satisfaça uma fórmula ϕ para uma interpretação s, *apenas interessa* o valor que s atribui às variáveis que ocorrem livres em ϕ. Em particular, temos a seguinte proposição:

Proposição 2.1. *Seja ϕ uma fórmula duma linguagem $\mathcal{L}$ e $\mathcal{A} = (A, \cdot^{\mathcal{A}})$ uma estrutura de $\mathcal{L}$. Sejam duas interpretações s_1 e s_2 tal que $s_1(x) = s_2(x)$ se x ocorre livre em ϕ. Então $\mathcal{A} \models_{s_1} \phi$ se e só se $\mathcal{A} \models_{s_2} \phi$*

Demonstração. Ideia: mostrar por indução na estrutura dos termos que então $s_1(t) = s_2(t)$ se t é um termo de ϕ e em t só ocorrem variáveis livres de ϕ. Mostrar o resultado por indução na estrutura de ϕ. □

2.2.1.1 Proposições e modelos

A satisfazibilidade duma proposição numa estrutura não depende das interpretações.

Proposição 2.2. *Se ϕ é uma proposição de $\mathcal{L}$ e $\mathcal{A}$ uma estrutura de $\mathcal{L}$ então verifica-se uma das seguintes condições:*

1. $\mathcal{A} \models_s \phi$ *para toda a interpretação s*
2. $\mathcal{A} \not\models_s \phi$ *para toda a interpretação s*

Demonstração. Consequência imediata da proposição 2.1. □

Se $\mathcal{A} \models \phi$ diz-se que ϕ é *verdadeira* em $\mathcal{A}$ ou $\mathcal{A}$ é um *modelo* de ϕ. Se $\mathcal{A} \not\models \phi$ diz-se que ϕ é *falsa* em $\mathcal{A}$.
Se Σ for um conjunto de proposições e $\mathcal{A} \models \psi$ para todo $\psi \in \Sigma$ então A é um *modelo* de Σ e escreve-se $\mathcal{A} \models \Sigma$.

Corolário 2.1. *Seja $\Sigma \cup \{\phi\}$ um conjunto de proposições. $\Sigma \models \phi$ se e só se todo o modelo de Σ for um modelo de ϕ.*

Exercício 2.1. *Verifica o corolário anterior.* ◇

Exercício 2.2. *Seja $\mathcal{L}$ uma linguagem de 1$^{\underline{a}}$ ordem com igualdade e tal que $\mathcal{F}_0 = \{a, b\}$, $\mathcal{F}_1 = \{g\}$, $\mathcal{F}_2 = \{f, h\}$, $\mathcal{R}_1 = \{R, S\}$ e $\mathcal{R}_2 = \{P, Q\}$.*

i. Para cada uma das seguintes proposições diz, ***justificando*** *se é verdadeira ou falsa em $\mathcal{A}$*

a) $\forall x \exists y\ f(x, y) = a$

b) $\exists x \exists y\ f(x, y) = a$

c) $\forall x \exists y\ f(x, y) = a \rightarrow \exists y \forall x f(x, y) = a$

d) $\exists x R(x) \rightarrow \forall x R(x)$

ii. Considera as seguintes interpretações $s_i : Var \longrightarrow \mathbb{N}$ para $i = 1, 2, 3$ e onde:

- $s_1(x) = 2$, *para todo* $x \in Var$*;*
- $s_2(x) = 0$, *para todo* $x \in Var$*.*
- $s_3(x) = 3$, $s_3(y) = 1$ *e* $s_3(z) = 5$*.*

Para cada uma das fórmulas ϕ seguintes e cada uma das interpretações s_i, diz se $\mathcal{A} \models_{s_i} \phi$, para $i = 1, 2, 3$:

a) $\exists x \exists y\ f(x, y) = z$

b) $\exists z\, f(x,y) = z \to \forall y\, (S(y) \lor R(y))$

$\diamond$

Resolução 2.2.i

a) Pretende-se que para qualquer interpretação $s : Var \longrightarrow \mathbb{N}$,

$$\mathcal{A} \models_s \forall x \exists y\, f(x,y) = a \tag{2.1}$$

Iremos usar indutivamente a definição de $\models_s$.

Por `(vii)`, (2.1) é verdade, se para todo o $n \in \mathbb{N}$ se tem

$$\mathcal{A} \models_{s[n/x]} \exists y\, f(x,y) = a \tag{2.2}$$

Por `(viii)`, (2.2) é verdade, se existe um $m \in \mathbb{N}$ tal que

$$\mathcal{A} \models_{s[n/x][m/y]} f(x,y) = a \tag{2.3}$$

Por `(i)`, (2.3) é verdade, se

$$s[n/x][m/y](f(x,y)) = s[n/x][m/y](a) \tag{2.4}$$

Mas, pela definição de interpretação estendida a termos e de $s[a/x]$ para a pertencente ao domínio de $\mathcal{A}$,

$$s[n/x][m/y](f(x,y)) = f^{\mathcal{A}}(s[n/x][m/y](x), s[n/x][m/y](y)) = f^{\mathcal{A}}(n,m) = n + m$$

e $s[n/x][m/y](a) = a^{\mathcal{A}} = 1$

Em resumo, (2.1) é verdade se

Para todo o $n \in \mathbb{N}$, existe um $m \in \mathbb{N}$ tal que $n + m = 1$.

o que é **Falso**. Por exemplo, para $n = 3$ ter-se-ia $m = -2 \notin \mathbb{N}$.

b) Analogamente se obteria, para qualquer interpretação s

$$\mathcal{A} \models_s \exists x \exists y\, f(x,y) = a \tag{2.5}$$

se e só se

Existe um $n \in \mathbb{N}$ e existe um $m \in \mathbb{N}$ tal que $n + m = 1$.

o que é **Verdade**, por exemplo para $n = 0$ e $m = 1$.

c) Pretende-se que, para qualquer interpretação s

$$\mathcal{A} \models_s \forall x \exists y\ f(x,y) = a \rightarrow \exists y \forall x f(x,y) = a \tag{2.6}$$

Por (vi) da definição de $\models_s$, (2.6) é verdade se

$$\mathcal{A} \not\models_s \forall x \exists y\ f(x,y) = a \text{ ou } \mathcal{A} \models_s \exists y \forall x f(x,y) = a$$

Pela alínea (a), vimos que não é verdade que $\mathcal{A} \models_s \forall x \exists y\ f(x,y) = a$, logo $\mathcal{A} \not\models_s \forall x \exists y\ f(x,y) = a$ é verdade. E também (2.6).

d) Pretende-se que, para qualquer interpretação s

$$\mathcal{A} \models_s \exists x R(x) \rightarrow \forall x R(x) \tag{2.7}$$

Por (vi) da definição de $\models_s$, (2.7) é verdade se

$$\mathcal{A} \not\models_s \exists x R(x) \text{ ou } \mathcal{A} \models_s \forall x R(x)$$

Vamos determinar se $\mathcal{A} \models_s \exists x R(x)$. Isto é verdade, se existe $n \in \mathbb{N}$ tal que

$$\mathcal{A} \models_{s[n/x]} R(x) \tag{2.8}$$

Por (ii), (2.8) é verdade se

$$s[n/x](x) \in R^{\mathcal{A}} \tag{2.9}$$

Como $s[n/x](x) = n$, vem que (2.8) é verdade se

Existe um $n \in \mathbb{N}$ tal que n é ímpar

o que é **Verdade**. Então $\mathcal{A} \not\models_s \exists x R(x)$ é **Falso**.

Temos que analisar a veracidade de $\mathcal{A} \models_s \forall x R(x)$. Analogamente obtemos que é verdade se

Para todo o $n \in \mathbb{N}$, n é ímpar

o que também é **Falso**. Donde (2.7) é **Falsa**.

Resolução 2.2i.ii

a) Sendo s_1 uma interpretação que atribuí o valor 2 a qualquer variável, pretende-se que

$$\mathcal{A} \models_{s_1} \exists x \exists y\ f(x,y) = z \tag{2.10}$$

Isto é verdade se, existe um $n \in \mathbb{N}$ e existe um $m \in \mathbb{N}$ tal que

$$\mathcal{A} \models_{s_1[n/x][m/y]} f(x,y) = z \tag{2.11}$$

Isto é se,

$$s_1[n/x][m/y](f(x,y) = s_1[n/x][m/y](z) \tag{2.12}$$

Mas $s_1[n/x][m/y](f(x,y) = f^{\mathcal{A}}(n,m) = n + m$ e $s_1[n/x][m/y](z) = s_1(z) = 2$

Então (2.10) é verdade se

Existe um $n \in \mathbb{N}$ e existe um $m \in \mathbb{N}$ tal que $m + n = 2$

o que é **Verdade**. Basta tomar $n = m = 1$.

b) Sendo s_1 uma interpretação que atribuí o valor 2 a qualquer variável, pretende-se que

$$\mathcal{A} \models_{s_1} \exists z\ f(x,y) = z \rightarrow \forall y\ (S(y)\ \vee\ R(y)) \tag{2.13}$$

Mais uma vez isto é verdade se $\mathcal{A} \not\models_{s_1} \exists z\ f(x,y) = z$ ou se $\mathcal{A} \models_{s_1} \forall y\ (S(y)\ \vee\ R(y))$

Temos que:

$$\mathcal{A} \models_{s_1} \exists z\ f(x,y) = z \tag{2.14}$$

se existe $n \in \mathbb{N}$ tal que

$$\mathcal{A} \models_{s_1[n/z]} f(x,y) = z \tag{2.15}$$

Isto é se,

$$s_1[n/z](f(x,y)) = s_1[n/z](z) \tag{2.16}$$

e $s_1[n/z](f(x,y)) = f^{\mathcal{A}}(s_1[n/z](x), s_1[n/z](y)) = f^{\mathcal{A}}(s_1(x), s_1(y)) = 2+2$ e $s_1[n/z](z) = n$

Então (2.14) é verdade se

Existe um $n \in \mathbb{N}$ tal que $2 + 2 = n$

o que é **Verdade**. Tomar $n = 4$.

Mas então $\mathcal{A} \not\models_{s_1} \exists z\ f(x,y) = z$ é **Falso**.

Temos que

$$\mathcal{A} \models_{s_1} \forall y\ (S(y)\ \vee\ R(y)) \tag{2.17}$$

se, para todo o $n \in \mathbb{N}$

$$\mathcal{A} \models_{s_1[n/y]} (S(y) \ \vee \ R(y)) \tag{2.18}$$

Por `(v)` da definição de $\models_s$, (2.18) é verdade se $\mathcal{A} \models_{s_1[n/y]} S(y)$ ou $\mathcal{A} \models_{s_1[n/y]} R(y)$

$$\mathcal{A} \models_{s_1[n/y]} S(y) \tag{2.19}$$

se

$$s_1[n/y](y) \in S^{\mathcal{A}} \tag{2.20}$$

isto é se n é par.

Analogamente $\mathcal{A} \models_{s_1[n/y]} R(y)$, se n é ímpar. Então, (2.17) é verdade se

Para todo o $n \in \mathbb{N}$, n é par ou n é ímpar

o que é **Verdade**. Nota que é importante a disjunção estar no âmbito do quantificador e não o contrário: é falso que *para todo* $n \in \mathbb{N}$, n *é par ou para todo* $n \in \mathbb{N}$, n *é ímpar.*

Finalmente, temos que (2.13) é **Verdade**.

Exercício 2.3. *Seja $\mathcal{L}$ uma linguagem de primeira ordem com igualdade e tal que $\mathcal{F}_0 = \{a\}$, $\mathcal{F}_1 = \{g\}$, $\mathcal{R}_1 = \{R\}$ e $\mathcal{R}_2 = \{P, Q\}$. Seja $\mathcal{A}$ a estrutura de $\mathcal{L}$ definida por:*

- *o universo de $\mathcal{A}$ é o conjunto de números naturais $\mathbb{N} = \{0, 1, \ldots\}$;*
- $a^{\mathcal{A}} = 2$*;*
- $g^{\mathcal{A}}(n) = n + 1$*, para* $n \in \mathbb{N}$*;*
- $R^{\mathcal{A}} = \{n \in \mathbb{N} \,|\, n \text{ é par}\}$*;*
- $P^{\mathcal{A}} = \{(n, m) \in \mathbb{N}^2 \,|\, n \leq m\}$*;*
- $Q^{\mathcal{A}} = \emptyset$*.*

Para cada uma das proposição indica se é verdadeira ou falsa em $\mathcal{A}$.

1. $\exists x g(x) = a \wedge \forall x g(x) \neq x$
2. $\exists x \exists y Q(x, y)$
3. $\neg R(a) \rightarrow \forall x \forall y \, Q(x, y)$
4. $\forall x (R(x) \rightarrow \forall x \, R(g(g(x))) \,)$
5. $\forall x \exists y P(x, y)$
6. $\exists y \forall x P(x, y)$
7. $\exists y \forall x P(y, x)$

8. $\forall x(R(x) \lor \exists y(y = g(x) \land R(y))$

$\diamond$

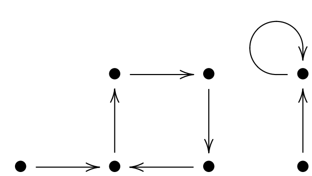
Exercício 2.4. *Seja $\mathcal{L}$ a mesma linguagem do exercício 2.3. Para cada uma das proposição seguintes indique uma estrutura de $\mathcal{L}$ onde ela é verdadeira e outra onde ela é falsa. Pode concluir que nenhuma das proposição e também nenhuma negação de uma das proposição é uma fórmula válida?*

1. $\forall x \forall y\, x = y$
2. $\forall x \exists y P(x, y) \to \exists y \forall x P(x, y)$
3. $\forall x\, x = a$
4. $\forall x \forall y(R(x) \to R(y))$
5. $\exists z(g(z) \neq a) \to \exists z(g(z) = a)$
6. $\forall x\, Q(x, g(x)) \leftrightarrow \exists x\, P(a, x)$
7. $\forall y(\forall x(R(x) \to P(x, x)) \to (R(y) \to \forall x P(x, x)))$

$\diamond$

Uma proposição pode ser vista como uma caracterização de um conjunto de modelos (que a satisfazem).

Estruturas para a teoria dos grafos Vamos ilustrar isso considerando que as estruturas são grafos dirigidos e considerar proposições que caracterizam propriedades desses grafos.
Seja $\mathcal{L}_G$ uma linguagem de $1^{\underline{a}}$ ordem com igualdade, sem símbolos funcionais e apenas com um símbolo relacional binário $R_2 = \{G\}$. Algumas fórmulas são: $G(x, x)$, $\exists x \forall y\ G(y, x)$, $\forall x \forall y(G(x, y) \to G(y, x))$
Qualquer estrutura $\mathcal{G}$ para $\mathcal{L}_G$ é um grafo (dirigido)!
Seja ϕ_1 a fórmula $\forall x \exists y G(x, y)\ \land\ \forall x \forall y \forall z((G(x, y)\ \land\ G(x, z)) \to y = z)$
Considera a seguinte estrutura $\mathcal{G}_1$:



Tem-se que $\mathcal{G}_1 \models \phi_1$. E que outros grafos satisfazem ϕ_1? Os que cada nó tem exactamente grau de partida 1 (isto é representam funções).

Exercício 2.5. *Descreve os grafos que são os modelos de cada uma das seguintes fórmulas:*

1. $\forall x \forall y (G(x,y) \rightarrow G(y,x)))$

2. $\forall x \forall y \forall z ((G(x,z) \ \wedge \ G(z,y)) \rightarrow G(x,y))$

$\diamond$

Inversamente, dado um grafo $\mathcal{G}$ podemos estar interessados em saber se ele tem uma dada propriedade. Se essa propriedade for expressa por uma fórmula ϕ de $\mathcal{L}_{\mathcal{G}}$ (não necessariamente uma proposição) então o que queremos é saber se $\mathcal{G} \models \phi$.

Exercício 2.6. *Mostra que dado ϕ de $\mathcal{L}_{\mathcal{G}}$ e $\mathcal{G}$ de domínio finito, existe um algoritmo que determina se $\mathcal{G} \models \phi$.* $\diamond$

Resolução 2.6
Por indução na estrutura de ϕ.

Estruturas para autómatos finitos Seja $\mathcal{L}_A$ uma linguagem de $1^{\underline{a}}$ ordem com igualdade e $R_1 = \{I, F\}$, $R_2 = \{R_a, R_b\}$. Uma estrutura $\mathcal{A}$ para $\mathcal{L}_A$ é um autómato finito não determinístico sobre $\{a, b\}$ se e só se $|I^A| = 1$.

Exemplo 2.8. *a) Define uma proposição ϕ de $\mathcal{L}_A$ tal que uma estrutura $\mathcal{A}$ de $\mathcal{L}_A$ é um modelo de ϕ se e só se $\mathcal{A}$ é um autómato finito determinístico.*

Basta considerar ϕ a fórmula:

$$\forall x \forall y \forall y_1 (((R_a(x,y) \ \wedge \ R_a(x,y_1)) \rightarrow y = y_1) \ \wedge \ ((R_b(x,y) \ \wedge \ R_b(x,y_1)) \rightarrow y = y_1))$$

*b) Indica, **justificando**, duas estruturas de $\mathcal{L}_A$, $\mathcal{A}_1$ e $\mathcal{A}_2$ tal que $\mathcal{A}_1 \models \phi$ e $\mathcal{A}_2 \models \neg\phi$.*

Seja a estrutura $\mathcal{A}_1$ de $\mathcal{L}_A$:

- *o universo é $Q = \{q_0, q_1\}$*
- $I^{\mathcal{A}_1} = \{q_0\}$
- $F^{\mathcal{A}_1} = \{q_1\}$
- $R_a^{\mathcal{A}_1} = \{(q_0, q_0)\}$
- $R_b^{\mathcal{A}_1} = \{(q_0, q_1), (q_1, q_1)\}$

Verifica que $\mathcal{A}_1 \models \phi$.

Considera agora $\mathcal{A}_2$ de $\mathcal{L}_A$ igual a $\mathcal{A}_1$, excepto que $R_a^{\mathcal{A}_1} = \{(q_0, q_0), (q_0, q_1)\}$.

Verifica que $\mathcal{A}_2 \models \neg\phi$.

Exercício 2.7. *a) Define uma proposição ϕ de $\mathcal{L}_A$ tal que uma estrutura $\mathcal{A}$ de $\mathcal{L}_A$ é um modelo de ϕ se e só se $\mathcal{A}$ é um autómato finito que reconhece alguma palavra de $\Sigma = \{a, b\}$ de comprimento 2.*

b) Indica, ***justificando****, duas estruturas de $\mathcal{L}_A$, $\mathcal{A}_1$ e $\mathcal{A}_2$ tal que $\mathcal{A}_1 \models \phi$ e $\mathcal{A}_2 \models \neg\phi$*

$\diamond$

2.2.1.2 Validade e satisfazibilidade

Proposição 2.3. *Seja ϕ uma fórmula duma linguagem de $1^{\underline{a}}$ ordem $\mathcal{L}$.*

1. *ϕ é válida se e só se $\neg\phi$ não é satisfazível*

2. *ϕ é satisfazível se e só se $\neg\phi$ não é válida*

3. *ϕ é válida se e só se $\forall x\ \phi$ é válida*

4. *ϕ é satisfazível se e só se $\exists x\ \phi$ é satisfazível*

Demonstração. Consequência directa das definições. $\square$

Seja ϕ uma fórmula duma linguagem de $1^{\underline{a}}$ ordem $\mathcal{L}$. Uma fórmula ϕ de $\mathcal{L}$ é válida se a correspondente forma booleana é uma tautologia.

Definição 2.9. *Dada uma fórmula ϕ o* conjunto das suas sub-fórmulas principais*, $SP(\phi)$ é definido indutivamente na estrutura de ϕ por:*

- *se ϕ é uma fórmula atómica, $SP(\phi) = \{\phi\}$*
- *se ϕ é $\forall x\psi$, $SP(\phi) = \{\phi\}$*
- *se ϕ é $\exists x\psi$, $SP(\phi) = \{\forall x\neg\psi\}$*
- *se ϕ é $\neg\psi$, $SP(\phi) = SP(\psi)$*
- *se ϕ é $\psi_1 \circ \psi_2$, onde $\circ$ pode ser $\wedge$, $\vee$ ou $\rightarrow$, $SP(\phi) = SP(\psi_1) \cup SP(\psi_2)$*

Sendo ϕ a fórmula $\forall x P(x, y) \wedge \exists x P(x, y) \wedge (P(z, x) \vee \forall x P(x, y))$

$$SP(\phi) = \{\forall x P(x, y), \forall x \neg P(x, y), P(z, x)\}$$

Definição 2.10. *Associando a cada sub-fórmula principal duma fórmula ϕ, uma variável proposicional diferente e negando a fórmula correspondente a um quantificador existêncial, obtemos a* forma booleana *de ϕ.*

Para o exemplo anterior,

$$p_1 \wedge \neg p_2 \wedge (p_3 \vee p_1)$$

onde $p_1 = \forall x P(x,y)$, $p_2 = \forall x \neg P(x,y)$ e $p_3 = P(z,x)$

Proposição 2.4. *Se a forma booleana duma fórmula ϕ é uma tautologia, então ϕ é válida.*

Demonstração. Seja $\mathcal{A}$ uma estrutura de $\mathcal{L}$ e s uma interpretação . Para cada $\psi \in SP(\phi)$, se $\mathcal{A} \models_s \psi$ atribui-se à correspondente variável proposicional o valor **V**, se $\mathcal{A} \not\models_s \psi$, atribui-se o valor **F**. Como a forma booleana é satisfeita para essa valorização, também se tem que $\mathcal{A} \models_s \phi$. □

2.2.2 Equivalência semântica

Definição 2.11. *Sejam ϕ e ψ duas fórmulas duma linguagem de $1^{\underline{a}}$ ordem $\mathcal{L}$. Se $\phi \models \psi$ e $\psi \models \phi$ então ϕ e ψ são* semanticamente equivalentes *e escreve-se $\phi \equiv \psi$.*

As *Leis de DeMorgan* para quantificadores são fórmulas semanticamente equivalentes são as

Proposição 2.5. *(Leis de DeMorgan)*

$$\begin{aligned} \neg\forall x\phi &\equiv \exists x\neg\phi \\ \neg\exists x\phi &\equiv \forall x\neg\phi \end{aligned}$$

Demonstração. Suponhamos que para toda a estrutura $\mathcal{A}$, e interpretação s se tem $\mathcal{A} \models_s \neg\forall x\phi$. Isto equivale a dizer que $\mathcal{A} \not\models_s \forall x\phi$, isto é, não se verifica que para todo o $a \in A$, $\mathcal{A} \models_{s[a/x]} \phi$. Isto é, existe pelo menos um $a \in A$ tal que $\mathcal{A} \not\models_{s[a/x]} \phi$, i.e, $\mathcal{A} \models_{s[a/x]} \neg\phi$, que é a definição de $\mathcal{A} \models_s \exists x\neg\phi$. O que prova a primeira equivalência (dado termos usado só as definições). Analogamente se prova a segunda lei de DeMorgan. □

Temos também:

Proposição 2.6.

$$\begin{aligned} \forall x(\phi \wedge \psi) &\equiv \forall x\phi \wedge \forall x\psi \quad \textit{(mas não com } \vee \textit{)} \\ \exists x\phi \vee \exists x\psi &\equiv \exists x(\phi \vee \psi) \quad \textit{(mas não com } \wedge \textit{)} \\ \forall x\forall y\phi &\equiv \forall y\forall x\phi \\ \exists x\exists y\phi &\equiv \exists y\exists x\phi \end{aligned}$$

Se x não ocorre livre em ψ

$$\begin{aligned}
\forall x(\phi \wedge \psi) &\equiv \forall x\phi \wedge \psi \\
\forall x(\phi \vee \psi) &\equiv \forall x\phi \vee \psi \\
\exists x(\phi \wedge \psi) &\equiv \exists x\phi \wedge \psi \\
\exists x(\phi \vee \psi) &\equiv \exists x\phi \vee \psi \\
\forall x(\psi \rightarrow \phi) &\equiv \psi \rightarrow \forall x\phi \\
\exists x(\phi \rightarrow \psi) &\equiv \forall x\phi \rightarrow \psi \\
\exists x(\psi \rightarrow \phi) &\equiv \psi \rightarrow \exists x\phi \\
\forall x(\phi \rightarrow \psi) &\equiv \exists x\phi \rightarrow \psi
\end{aligned}$$

Exercício 2.8. *Mostra as equivalências anteriores.* $\diamond$

2.2.3 Substituição de variáveis

Seja ϕ uma fórmula duma linguagem de 1ª ordem $\mathcal{L}$, t um termo de $\mathcal{L}$ e $x \in Var$ uma variável. Denota-se por $\phi[t/x]$ (ou ϕ_x^t) a fórmula que se obtém de ϕ *substituindo* todas as ocorrências livres de x em ϕ por t.

Seja ϕ a fórmula $\forall x(P(x) \wedge Q(x)) \rightarrow (\neg P(x) \vee Q(y))$.
$\phi[f(x,y)/x]$ é a fórmula $\forall x(P(x) \wedge Q(x)) \rightarrow (\neg P(f(x,y)) \vee Q(y))$
E se ϕ for $\forall y(\neg P(x) \vee Q(y))$, qual a fórmula $\phi[f(x,y)/x]$? Fica

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

mas aqui a variável y do termo $f(x,y)$ passou a estar ligada!
Para evitar isto, define-se que:

Definição 2.12 (Variável substituível)**.** *Uma variável x é* substituível *por t em ϕ se nenhuma ocorrência livre de x em ϕ for numa sub-fórmula $\forall y\psi$ ou $\exists y\psi$, para qualquer variável y que ocorra em t. Também se pode dizer que t é* livre *para x em ϕ.*

Exercício 2.9. *Seja ϕ a fórmula $\exists x(P(y,z) \wedge \forall y(\neg Q(x,y) \vee P(y,z)))$*
Determina:

1. *$\phi[w/x]$, $\phi[w/y]$, $\phi[f(x)/y]$ e $\phi[g(y,z)/z]$*

2. *Para quais dos termos w, $f(x)$ e $g(y,z)$ é x substituível em ϕ? e y?*

$\diamond$

Proposição 2.7. *1. Se x é substituível por t em ϕ, são válidas as fórmulas:*

a) $\forall x\, \phi \rightarrow \phi[t/x]$

b) $\phi[t/x] \rightarrow \exists x\phi$

2. *Se* y *não ocorre em* ϕ, *então* $\forall x\phi \equiv \forall y\phi[y/x]$

Demonstração. Vamos apenas considerar 2.71a. Seja $\mathcal{A}$ uma estrutura e s uma interpretação. Temos que mostrar que se $\mathcal{A} \models_s \forall x\phi$ então $\mathcal{A} \models \phi[t/x]$. Para todo o $a \in A$, temos $\mathcal{A} \models_{s[a/x]} \phi$. Seja $a' = s(t)$, e como t é livre para x em ϕ sabemos que esse será o valor da interpretação de todas as ocorrências de t em $\phi[t/x]$. Mas então temos também $\mathcal{A} \models_s \phi[t/x]$. $\square$

Exercício 2.10. *Termina a demonstra a Proposição 2.7.* $\diamond$

2.2.4 Forma normal prenexa

Uma fórmula duma linguagem de 1ª ordem $\mathcal{L}$ está em *forma normal prenexa* se ou:

- não contém quantificadores
- é da forma
$$Q_1x_1Q_2x_2\ldots Q_nx_n\phi$$
onde cada Q_i é ou $\forall$ ou $\exists$, $1 \leq i \leq n$, e ϕ é uma fórmula sem quantificadores.

Por exemplo,

$$\forall x\forall y\forall z\neg((\neg P(x,x) \lor (\neg P(x,y) \land P(z,z))) \lor \neg P(w,0))$$

Proposição 2.8. *Qualquer fórmula duma linguagem de 1ª ordem é semanticamente equivalente a uma fórmula em forma normal prenexa.*

Demonstração. **Ideia:**
Transformar ϕ de modo a que:

- só ocorram as conectivas $\land$, $\lor$ e $\neg$
- nenhuma variável ocorra livre e ligada
- uma mesma variável só aparece quantificada no máximo uma vez.

Seja ϕ_1 a fórmula resultante, que é semanticamente equivalente a ϕ(mostra!). Aplicar às subfórmulas de ϕ_1, as seguintes transformações, até que nenhuma se aplique (onde $\circ$ é $\land$ ou $\lor$):

$$\begin{array}{lllllll}
\neg\forall x\phi & \longrightarrow & \exists x\neg\phi & \quad & \forall x\psi\circ\phi & \longrightarrow & \forall x(\psi\circ\phi)\\
\neg\exists x\phi & \longrightarrow & \forall x\neg\phi & & \phi\circ\exists x\psi & \longrightarrow & \exists x(\phi\circ\psi)\\
\phi\circ\forall x\psi & \longrightarrow & \forall x(\phi\circ\psi) & & \exists x\psi\circ\phi & \longrightarrow & \exists x(\psi\circ\phi)
\end{array}$$

Por indução no número de quantificadores que estão no âmbito de conectivas ϕ_1 mostra-se que a fórmula resultante está em forma normal prenexa e é semanticamente equivalente a ϕ. □

Exercício 2.11. *Obtém uma forma normal prenexa para*

$$\forall x(P(x,x) \wedge (\forall y P(x,y) \vee \exists y \neg P(y,y))) \wedge G(w,0)$$

◇

Resolução 2.11

1. Substituir as variáveis ligadas por variáveis novas (Prop.2.7:2)

$$\forall x(P(x,x) \wedge (\forall y P(x,y) \vee \exists z \neg P(z,z))) \wedge G(w,0)$$

2. Mover $\forall x$ para fora:

$$\forall x((P(x,x) \wedge (\forall y P(x,y) \vee \exists z \neg P(z,z))) \wedge G(w,0))$$

3. Mover $\forall y$ para fora:

$$\forall x \forall y((P(x,x) \wedge (P(x,y) \vee \exists z \neg P(z,z))) \wedge G(w,0))$$

4. Mover $\exists z$ para fora:

$$\forall x \forall y \exists z((P(x,x) \wedge (P(x,y) \vee \neg P(z,z))) \wedge G(w,0))$$

Leituras suplementares [BE00] (Cap. 10,11)

2.3 Sistema de dedução natural para a lógica de 1ª ordem

As regras para as conectivas são as mesmas que para a lógica proposicional.
Haverá novas regras de **introdução** e **eliminação** para:

- a igualdade (=)
- para os quantificadores a introdução corresponderá a uma **generalização** e a eliminação uma **instanciação**:

  (∀-E) de $\forall x P(x)$ deduzir $P(t)$

  (∀-I) pode-se deduzir $\forall x P(x)$ se para qualquer y se deduzir $P(y)$

  (∃-I) de $P(t)$ deduzir $\exists x P(x)$

  (∃-E) se se deduziu $\exists x P(x)$ podemos supor $P(t)$

2.3.1 Regras de inferência DN:igualdade

Introdução de =

$$\frac{}{t = t}{=}\mathbf{I}$$

Podemos deduzir sempre que um termo t é igual a si próprio...a igualdade é reflexiva

Eliminação de =

$$\frac{t_1 = t_2 \qquad \phi[t_1/x]}{\phi[t_2/x]}{=}\mathbf{E} \quad \text{e } x \text{ é substituível por } t_1 \text{ e por } t_2 \text{ em } \phi$$

Se t_1 é igual a t_2 podemos substituir em ϕ, t_1 por t_2.

Exercício 2.12. *Mostra que* $t_1 = t_2 \vdash t_2 = t_1$ *e* $t_1 = t_2, t_2 = t_3 \vdash t_1 = t_3$*, i.e. , que são deduzíveis as propriedades de simetria e transitividade de* $=$*.* $\diamond$

Resolução 2.12

1	$t_1 = t_2$	
2	$t_1 = t_1$	$=$**I**
3	$t_2 = t_1$	$=$**E** (ϕ é $x = t_1$), 1, 2

1	$t_1 = t_2$	
2	$t_2 = t_3$	
3	$t_1 = t_3$	$=$**E** (ϕ é $t_1 = x$), 2, 2

2.3.2 Regras de inferência DN:quantificador universal

Eliminação de $\forall$

$$\frac{\forall x\ \phi}{\phi[t/x]}\ \forall\mathbf{E} \quad \text{onde } x \text{ é substituível por } t \text{ em } \phi$$

Se se deduziu $\forall x\ \phi$ podemos substituir x em ϕ por qualquer termo t (**instanciar** o x com qualquer valor).

Introdução de $\forall$

$$\frac{\begin{array}{l}[v] \\ \quad\vdots \\ \quad\phi[v/x]\end{array}}{\forall x\ \phi}\forall\mathbf{I} \quad \text{onde } v \text{ é uma variável nova (não ocorre antes)}$$

Se supondo uma variável nova y podemos deduzir $\phi[y/x]$, então podemos deduzir ϕ com qualquer valor (**generalização**). Para supor y temos de começar uma nova sub-dedução mas sem novas premissas. Na notação de Fitch isso é indicando com uma nova identação. Esta sub-dedução termina quando concluímos a regra.

Exercício 2.13. *Mostrar que*

$$\forall x(P(x) \to Q(x)), \forall x P(x) \vdash \forall x Q(x)$$

◇

Resolução 2.13

$$
\begin{array}{r|l|l l}
1 & \multicolumn{2}{l}{\forall x(P(x) \to Q(x))} & \\
2 & \multicolumn{2}{l}{\forall x P(x)} & \\
\hline
3 & v & P(v) \to Q(v) & \forall\text{E, 1} \\
4 & & P(v) & \forall\text{E, 2} \\
5 & & Q(v) & \to\text{E, 3, 4} \\
6 & \multicolumn{2}{l}{\forall x Q(x)} & \forall\text{I, 3–5}
\end{array}
$$

Exercício 2.14. *Mostrar que*

$$P(t), \forall x(P(x) \to \neg Q(x)) \vdash \neg Q(t)$$

◇

Resolução 2.14

$$
\begin{array}{r|l l}
1 & P(t) & \\
2 & \forall x(P(x) \to \neg Q(x)) & \\
\hline
3 & P(t) \to \neg Q(t) & \forall\text{E, 2} \\
4 & \neg Q(t) & \to\text{E, 3, 1}
\end{array}
$$

2.3.3 Regras de inferência DN:quantificador existencial

Introdução de $\exists$

$$\frac{\phi[t/x]}{\exists x\ \phi}\ \exists\text{I} \quad \text{onde } x \text{ é substituível por } t \text{ em } \phi$$

Podemos deduzir $\exists x\ \phi$ se já tivermos deduzido $\phi[t/x]$ para algum termo t.

Eliminação de $\exists$

$$\dfrac{\exists x\ \phi \qquad \begin{array}{c}[v \quad \phi[v/x]]\\ \vdots \\ \psi\end{array}}{\psi}\ \exists\mathbf{E}$$

onde v é uma variável nova que não ocorre antes nem em ψ

Se deduzimos $\exists x \phi$, então existe um valor em que ϕ se verifica. Assim, supondo que v representa esse valor e se supondo $\phi[v/x]$ se deduzir ψ onde v não ocorra, podemos deduzir ψ (para um valor genérico). Neste caso temos então uma nova sub-dedução em que é considerada uma nova variável (v) e onde existe uma premissa nova ($\phi[v/x]$).

Exercício 2.15. *Mostrar que:*

a) $\forall x\ \phi \vdash \exists x\ \phi$

$\forall x\ (P(x) \to Q(x)), \exists x\ P(x) \vdash \exists x\ Q(x)$

$\diamond$

Resolução 2.15

a)

1	$\forall x\ \phi$	
2	$\phi[t/x]$	$\forall$E, 1
3	$\exists x\ \phi$	$\exists$I, 2

b)

1	$\forall x\ (P(x) \to Q(x))$	
2	$\exists x\ P(x)$	
3	v $P(v)$	
4	$P(v) \to Q(v)$	$\forall$E, 1
5	$Q(v)$	$\to$E, 4, 3
6	$\exists x\ Q(x)$	$\exists$I, 5
7	$\exists x\ Q(x)$	$\exists$E, 3–6

Como no caso das regras de introdução e eliminação de conectivas, também aqui se tem de ter em conta o âmbito das sub-deduções e em especial as variáveis (livres) só podem ser usadas nas sub-deduções em que são supostas ou ocorrem em premissas dadas.

Por exemplo, a seguinte é uma dedução está *errada*:

1	$\forall x\ (P(x) \rightarrow Q(x))$	
2	$\exists x\ P(x)$	
3	$v \mid P(v)$	
4	$\quad\mid P(v) \rightarrow Q(v)$	$\forall$E, 1
5	$\quad\mid Q(v)$	$\rightarrow$E, 4, 3
6	$Q(v)$	$\exists$I, 3–5
7	$\exists x\ Q(x)$	$\exists$E, 6

Na fórmula da linha 6, ocorre uma variável que só devia ocorrer na sub-dedução iniciada na linha 3!

Exercício 2.16. *Mostrar que*

$$\forall x(Q(x) \rightarrow R(x)), \exists x\ (P(x)\ \wedge\ Q(x)) \vdash \exists x(P(x)\ \wedge\ R(x))$$

$\diamond$

Resolução 2.16

1	$\forall x(Q(x) \rightarrow R(x))$	
2	$\exists (P(x)\ \wedge\ Q(x))$	
3	$v \mid P(v)\ \wedge\ Q(v)$	
4	$\quad\mid Q(v) \rightarrow R(v)$	$\forall$E, 1
5	$\quad\mid Q(v)$	$\wedge$E, 3
6	$\quad\mid R(v)$	$\rightarrow$E, 4
7	$\quad\mid P(v)$	$\wedge$E, 3
8	$\quad\mid P(v)\ \wedge\ R(v)$	$\wedge$I, 7, 6
9	$\quad\mid \exists x\ (P(x)\ \wedge\ R(x))$	$\exists$I, 8
10	$\exists x\ (P(x)\ \wedge\ R(x))$	$\exists$E, 2, 3–9

Exercício 2.17. *Mostrar que*

$$\exists x\ P(x), \forall x \forall y\ (P(x) \rightarrow Q(y)) \vdash \forall y\ Q(y)$$

$\diamond$

Resolução 2.17

1		$\exists x\ P(x)$	
2		$\forall x \forall y\ (P(x) \rightarrow Q(y))$	
3	v u	$P(u)$	
4		$\forall y\ (P(u) \rightarrow Q(y)$	$\forall$E, 2
5		$(P(u) \rightarrow Q(v)$	$\forall$E, 4
6		$Q(v)$	$\rightarrow$E, 5, 3
7		$Q(v)$	$\exists$E, 1, 3–6
8		$\forall y\ Q(y)$	$\forall$I, 3–7

Todas as regras do sistema DN para a lógica de primeira ordem encontram-se na Figura 2.1

2.4 Equivalência dedutiva

Dadas duas fórmulas ϕ e ψ dizemos que são dedutivamente equivalentes se e só se $\phi \vdash \psi$ e $\psi \vdash \phi$. E denotamos por $\phi \dashv\vdash \psi$.

Proposição 2.9. *Sejam ϕ e ψ dumas fórmulas duma linguagem da lógica de $1^{\underline{a}}$ ordem. Então:*

1. $\neg\forall x\phi \dashv\vdash \exists x\neg\phi$
2. $\neg\exists x\phi \dashv\vdash \forall x\neg\phi$
3. $\forall x\phi \wedge \forall x\psi \dashv\vdash \forall x(\phi \wedge \psi)$
4. $\exists x\phi \vee \exists x\psi \dashv\vdash \exists x(\phi \vee \psi)$
5. *Se x não ocorre livre em ψ, e $\circ$ é $\wedge$ ou $\vee$:*

    (a) $\forall x\phi \circ \psi \dashv\vdash \forall x(\phi \circ \psi)$

    (b) $\exists x\phi \circ \psi \dashv\vdash \exists x(\phi \circ \psi)$

	Introdução	**Eliminação**
$\wedge$	$\dfrac{\phi \quad \psi}{\phi \wedge \psi} \wedge \mathbf{I}$	$\dfrac{\phi \wedge \psi}{\phi} \wedge \mathbf{E}_1 \qquad \dfrac{\phi \wedge \psi}{\psi} \wedge \mathbf{E}_2$
$\vee$	$\dfrac{\phi}{\phi \vee \psi} \vee \mathbf{I}_1 \qquad \dfrac{\psi}{\phi \vee \psi} \vee \mathbf{I}_2$	$\dfrac{\phi \vee \psi \qquad \begin{matrix}[\phi] \\ \vdots \\ \gamma\end{matrix} \quad \begin{matrix}[\psi] \\ \vdots \\ \gamma\end{matrix}}{\gamma} \vee \mathbf{E}$
$\neg$	$\dfrac{\begin{matrix}[\phi] \\ \vdots \\ \mathbf{F}\end{matrix}}{\neg\phi} \neg\mathbf{I}$	$\dfrac{\neg\neg\phi}{\phi} \neg\mathbf{E}$
$\mathbf{F}$	$\dfrac{\begin{matrix}\phi \\ \vdots \\ \neg\phi\end{matrix}}{\mathbf{F}} \mathbf{FI}(*)$	$\dfrac{\mathbf{F}}{\phi} \mathbf{FE}$
$\rightarrow$	$\dfrac{\begin{matrix}[\phi] \\ \vdots \\ \psi\end{matrix}}{\phi \rightarrow \psi} \rightarrow\mathbf{I}$	$\dfrac{\phi \quad \phi \rightarrow \psi}{\psi} \rightarrow\mathbf{E}$
$=$	$\dfrac{}{t=t} =\mathbf{I}$	$\dfrac{t_1 = t_2 \quad \phi[t_1/x]}{\phi[t_2/x]} =\mathbf{E}$ e x é substituível por t_1 e por t_2 em ϕ
$\forall$	$\dfrac{\begin{matrix}[v] \\ \vdots \\ \phi[v/x]\end{matrix}}{\forall x\ \phi} \forall\mathbf{I}$ onde v é uma variável nova (não ocorre antes)	$\dfrac{\forall x\ \phi}{\phi[t/x]} \ \forall\mathbf{E}$ onde x é substituível por t em ϕ
$\exists$	$\dfrac{\phi[t/x]}{\exists x\ \phi} \ \exists\mathbf{I}$ onde x é substituível por t em ϕ	$\dfrac{\exists x\ \phi \qquad \begin{matrix}[v \quad \phi[v/x]] \\ \vdots \\ \psi\end{matrix}}{\psi} \ \exists\mathbf{E}$ onde v é uma variável nova que não ocorre antes, nem em ψ
R	$\dfrac{\phi}{\phi} \mathbf{R}$	

Figura 2.1: As regras do sistema DN para a lógica de primeira ordem

Demonstração. $\neg\forall x\phi \vdash \exists x\neg\phi$ Por redução ao absurdo:

$$
\begin{array}{lll}
1 & \mid \underline{\neg\forall x\phi} & \\
2 & \mid \quad \mid \underline{\neg\exists x\neg\phi} & \\
3 & \mid \quad \mid u \mid \quad \mid \underline{\neg\phi[u/x]} & \\
4 & \mid \quad \mid \quad \mid \quad \mid \exists x\neg\phi & \exists\mathrm{I},\ 3 \\
5 & \mid \quad \mid \quad \mid \quad \mid \mathbf{F} & \mathbf{FI},\ 4,\ 2 \\
6 & \mid \quad \mid \quad \mid \phi[u/x] & \mathbf{RA},\ 3\text{–}5 \\
7 & \mid \quad \mid \forall x\phi & \forall\mathrm{I},\ 3\text{–}6 \\
8 & \mid \quad \mid \mathbf{F} & \mathbf{FI},\ 7,\ 1 \\
9 & \mid \exists x\neg\phi & \mathbf{RA},\ 2\text{–}8
\end{array}
$$

$\exists x\neg\phi \vdash \neg\forall x\phi$

$$
\begin{array}{lll}
1 & \mid \underline{\exists x\neg\phi} & \\
2 & \mid \quad \mid \underline{\forall x\phi} & \\
3 & \mid \quad \mid u \mid \underline{\neg\phi[u/x]} & \\
4 & \mid \quad \mid \quad \mid \phi[u/x] & \forall\mathrm{E},\ 2 \\
5 & \mid \quad \mid \quad \mid \mathbf{F} & \mathbf{FI},\ 3,\ 4 \\
6 & \mid \quad \mid \mathbf{F} & \exists\mathrm{E},\ 1,\ 2\text{–}5 \\
7 & \mid \neg\forall x\phi & \neg\mathrm{I},\ 2\text{–}6
\end{array}
$$

$\forall x\phi \ \wedge\ \psi \vdash \forall x(\phi \ \wedge\ \psi)$ e x não ocorre livre em ψ

$$
\begin{array}{lll}
1 & \mid \underline{\forall x\phi \ \wedge\ \psi} & \\
2 & \mid \forall x\phi & \wedge\mathrm{E},\ 1 \\
3 & \mid \psi & \wedge\mathrm{E},\ 1 \\
4 & \mid v \mid \phi[v/x] & \forall\mathrm{E},\ 2 \\
5 & \mid \quad \mid \phi[v/x] \ \wedge\ \psi & \wedge\mathrm{I},\ 3,\ 4 \\
6 & \mid \quad \mid (\phi \ \wedge\ \psi)[v/x] & \mathrm{R},\ 5 \\
7 & \mid \forall x(\phi \ \wedge\ \psi) & \forall\mathrm{I},\ 4\text{–}6
\end{array}
$$

$\forall x(\phi \wedge \psi) \vdash \forall x\phi \wedge \psi$ e x não ocorre livre em ψ

1		$\forall x(\phi \wedge \psi)$	
2	v	$(\phi \wedge \psi)[v/x]$	$\forall$E, 1
3		$\phi[v/x] \wedge \psi$	R, 2
4		ψ	$\wedge$E, 3
5		$\phi[v/x]$	$\wedge$E, 3
6		$\forall x\phi$	$\forall$I, 2–5
7		$(\forall x\phi) \wedge \psi$	$\wedge$I, 4, 6

$(\exists x\phi) \vee (\exists x\psi) \dashv\vdash \exists x(\phi \vee \psi)$

1		$(\exists x\phi) \vee (\exists x\psi)$	
2		$\exists x\phi$	
3	v	$\phi[v/x]$	
4		$\phi[v/x] \vee \psi[v/x]$	$\vee$I, 3
5		$(\phi \vee \psi)[v/x]$	idêntico, 4
6		$\exists x(\phi \vee \psi)$	$\exists$I, 5
7		$\exists x(\phi \vee \psi)$	$\exists$E, 2, 3–5
8		$\exists x\psi$	
9	v	$\psi[v/x]$	
10		$\phi[v/x] \vee \psi[v/x]$	$\vee$I, 9
11		$(\phi \vee \psi)[v/x]$	idêntico, 10
12		$\exists x(\phi \vee \psi)$	$\exists$I, 11
13		$\exists x(\phi \vee \psi)$	$\exists$E, 8, 9–12
14		$\exists x(\phi \vee \psi)$	$\vee$E, 1, 2–7, 8–13

1	$\exists x(\phi \lor \psi)$	
2	$v \quad (\phi \lor \psi)[v/x]$	
3	$\phi[v/x] \lor \psi[v/x]$	idêntico, 2
4	$\phi[v/x]$	
5	$\exists x\phi$	$\exists$I, 4
6	$\exists x\phi \lor \exists x\psi$	$\lor$I, 5
7	$\psi[v/x]$	
8	$\exists x\psi$	$\exists$I, 7
9	$\exists x\phi \lor \exists x\psi$	$\lor$I, 8
10	$\exists x\phi \lor \exists x\psi$	$\lor$E, 3, 4–6, 7–9
11	$\exists x\phi \lor \exists x\psi$	$\exists$E, 1, 2–10

□

Exercício 2.18. *Mostra que:*

a) $\vdash \forall x(\phi \to \psi) \to (\exists x\phi \to \exists x\psi)$

b) $\vdash \forall x(\phi \to \psi) \to (\forall x\phi \to \forall x\psi)$

c) $\exists xP(x), \forall x\forall y(P(x) \land P(y)) \to x = y) \vdash \exists!xP(x)$

onde $\exists!xP(x)$ *representa* $\exists x(P(x) \land \forall y(P(y) \to y = x))$ *e diz-se* existe um único x tal que $P(x)$

d) $\vdash \forall x\forall yx = y \to f(x) = f(y)$

e) $\vdash x = f(y) \to (\forall zP(x, z) \to P(f(y), z))$

◇

Resolução 2.18

a) $\vdash \forall x(\phi \to \psi) \to (\exists x\phi \to \exists x\psi)$

1	$\forall x(\phi \to \psi)$	
2	$\quad \exists x\phi$	
3	$\quad u \quad \phi[u/x]$	
4	$\quad\quad (\phi \to \psi)[u/x]$	$\forall$E, 1
5	$\quad\quad \phi[u/x] \to \psi[u/x]$	R, 4
6	$\quad\quad \psi[u/x]$	$\to$E, 3, 5
7	$\quad\quad \exists x\psi$	$\exists$I, 6
8	$\quad \exists x\psi$	$\exists$E, 2, 3–7
9	$\exists x\phi \to \exists x\psi$	$\to$I, 2–8

b) $\vdash \forall x(\phi \to \psi) \to (\forall x\phi \to \forall x\psi)$

1	$\forall x(\phi \to \psi)$	
2	$\quad \forall x\phi$	
3	$\quad u \quad \phi[u/x]$	$\forall$E, 2
4	$\quad\quad (\phi \to \psi)[u/x]$	$\forall$E, 1
5	$\quad\quad \phi[u/x] \to \psi[u/x]$	R, 4
6	$\quad\quad \psi[u/x]$	$\to$E, 3, 5
7	$\quad \forall x\psi$	$\forall$I, 4–6
8	$\forall x\phi \to \forall x\psi$	$\to$I, 2–8

c) $\exists xP(x), \forall x\forall y(P(x) \land P(y) \to x = y) \vdash \exists x(P(x) \land \forall y(P(y) \to y = x))$

1	$\exists x P(x)$	
2	$\forall x \forall y (P(x) \wedge P(y) \rightarrow x = y)$	
3	u $P(u)$	
4	v $P(v)$	
5	$P(v) \wedge P(u)$	$\wedge$I, 3, 4
6	$\forall y((P(v) \wedge P(y) \rightarrow v = y)$	$\forall$E, 2
7	$(P(v) \wedge P(u)) \rightarrow v = u)$	$\forall$E, 6
8	$v = u$	$\rightarrow$E, 5, 7
9	$P(v) \rightarrow v = u$	$\rightarrow$I, 4–8
10	$\forall y(P(y) \rightarrow y = u)$	$\rightarrow$I, 4–8
11	$P(u) \wedge \forall y(P(y) \rightarrow y = u)$	$\wedge$I, 3, 10
12	$\exists x(P(x) \wedge \forall y(P(y) \rightarrow y = x))$	$\exists$I, 11
13	$\exists x(P(x) \wedge \forall y(P(y) \rightarrow y = x))$	$\exists$E, 1, 2–12

d) $\vdash \forall x \forall y\ x = y \rightarrow f(x) = f(y)$

1	u v $u = v$	
2	$f(u) = f(u)$	=I
3	$f(u) = f(v)$	=E (ϕ é $f(u) = f(x)$), 1, 3
4	$u = v \rightarrow f(u) = f(v)$	$\rightarrow$I, 1, 3
5	$\forall y\ u = y \rightarrow f(u) = f(y)$	$\forall$E, 1–4
6	$\forall x \forall y\ x = y \rightarrow f(x) = f(y)$	$\forall$E, 1–5

e) $\vdash x = f(y) \rightarrow \forall (z P(x, z) \rightarrow P(f(y), z))$

1	$x = f(y)$	
2	u $P(x, u)$	
3	$P(f(y), u)$	=E, 1, 2
4	$P(x, u) \rightarrow P(f(y), u)$	$\rightarrow$I, 2–3
5	$\forall z(P(x, z) \rightarrow P(f(y), z))$	$\forall$I, 2–4
6	$x = f(y) \rightarrow \forall z(P(x, z) \rightarrow P(f(y), z))$	$\rightarrow$I, 1–5

Leituras suplementares [BE00] (Cap. 2.3,12,13)

2.5 Integridade e completude

Vamos ver que o sistema dedutivo DN para a lógica de 1ª ordem é **integro** e **completo**, i.e: Dado um conjunto de fórmulas Σ (premissas) e uma fórmula ϕ (conclusão):

Integridade se existe uma dedução de ϕ com premissas Σ, $\Sigma \vdash \phi$, então ϕ é consequência semântica de Σ, $\Sigma \models \phi$. Em particular se $\vdash \phi$ (teorema), então ϕ é válida, $\models \phi$.

Completude Se ϕ é consequência semântica de Σ, $\Sigma \models \phi$, então existe uma dedução de ϕ com premissas Σ, $\Sigma \vdash \phi$. Em particular se $\models \phi$ então ϕ é um teorema, $\vdash \phi$.

2.5.1 Integridade do sistema dedutivo DN

Proposição 2.10. *Se* $\Sigma \vdash \phi$ *então* $\Sigma \models \phi$

Demonstração. Dada uma dedução de ϕ com premissas de Σ (onde, podem não ser usadas todas as fórmulas), vamos mostrar que em cada passo p a fórmula que aí ocorre é consequência semântica das premissas (ou hipóteses) que aí são assumidas. Então em particular, ϕ sendo o último passo, será consequência semântica das suas premissas (ou seja Σ). Provamos por indução no número de passos da dedução:

Base. Se só houver um passo de dedução então ϕ é uma das premissas ou uma fórmula da forma $t = t$. Em ambos os casos é fácil ver que são consequências semânticas das premissas (mostra!)

Indução. Suponhamos que estamos no passo n e que todos os anteriores verificam a condição. Fazemos uma demonstração por casos considerando cada uma das regras. Suponhamos que a regra a aplicar é a eliminação da implicação:

$\rightarrow$ **E**

$\frac{\phi \quad \phi\rightarrow\psi}{\psi}$: Seja θ a fórmula deduzida no passo n por aplicação de $\rightarrow$ **E** a $\phi \rightarrow \theta$ e ϕ. E sejam $\phi_1, \ldots, \phi_k$ as premissas assumidas em θ. Mas as premissas para $\phi \rightarrow \theta$ e ϕ estão entre os $\phi_1, \ldots, \phi_k$ e ambos são consequência semânticas delas:

$$
\begin{array}{r|l|l|l}
1 & \multicolumn{3}{l}{\phi_1} \\ \cline{2-2}
\vdots & \multicolumn{3}{l}{\vdots} \\
i & \multicolumn{3}{l}{\phi \to \theta} \\
\vdots & \multicolumn{3}{l}{\vdots} \\
 & & \multicolumn{2}{l}{\phi_2} \\ \cline{3-3}
\vdots & & \multicolumn{2}{l}{\vdots} \\
l & & \multicolumn{2}{l}{\phi} \\
\vdots & & \multicolumn{2}{l}{\vdots} \\
 & & & \phi_3 \\ \cline{4-4}
\vdots & & & \vdots \\
n & & & \theta \\
\vdots & & & \vdots
\end{array}
$$

Vamos ver que θ é consequência semântica de $\phi_1, \dots, \phi_k$. Seja $\mathcal{A}$ uma estrutura e s uma interpretação tal que $\mathcal{A} \models_s \phi_i$, $1 \leq i \leq k$. Então, também $\mathcal{A} \models_s \phi \to \theta$ e $\mathcal{A} \models_s \phi$. Mas então, pela definição `(vi)` de $\models_s$, tem-se que $\mathcal{A} \models_s \theta$. E então $\{\phi_1, \dots, \phi_k\} \models \theta$.

Vejamos agora o caso da regra da eliminação do quantificador existencial:

$$
\dfrac{\exists x\, \phi \qquad \begin{array}{c} [v \quad \phi[v/x]] \\ \vdots \\ \psi \end{array}}{\psi}\ \exists\mathbf{E}:
$$

Seja θ a fórmula deduzida no passo n por aplicação de $\exists\mathbf{E}$ a $\exists x\phi$ e a uma sub-dedução que contêm θ. E sejam $\phi_1, \dots, \phi_k$ as premissas assumidas no passo n, em θ. Por hipótese de indução no passo i, $\exists x\phi$ é consequência semântica de premissas que são um subconjunto de $\phi_1, \dots, \phi_k$ e no passo m, θ é consequência semântica de premissas que são um subconjunto de $\phi_1, \dots, \phi_k$, *mais a premissa* $\phi[v/x]$.

$$
\begin{array}{l|ll}
\vdots & \vdots & \\
i & \exists x\phi & \\
\vdots & \vdots & \\
 & v & \multicolumn{1}{|l}{\phi[v/x]} \\ \cline{3-3}
\vdots & & \multicolumn{1}{|l}{\vdots} \\
m & & \multicolumn{1}{|l}{\theta} \\
\vdots & & \multicolumn{1}{|l}{\vdots} \\
\vdots & \vdots & \\
n & \theta &
\end{array}
$$

Vamos ver que θ é consequência semântica de $\phi_1, \ldots, \phi_k$. Seja $\mathcal{A}$ uma estrutura e s uma interpretação tal que $\mathcal{A} \models_s \phi_i$, $1 \leq i \leq k$. E também $\mathcal{A} \models_s \exists x\phi$. Isto é, existe um $b \in A$ tal que $\mathcal{A} \models_{s[b/x]} \phi$. Nota, ainda, que v não pode ocorrer em $\phi_1, \ldots, \phi_k$, $\exists x\phi$ e θ. Seja $s' = s[b/v]$ e, claramente, $\mathcal{A} \models_{s[b/v]} \phi[v/x]$. Mas também $\mathcal{A} \models_{s'} \phi_i$, $1 \leq i \leq k$ e, então, $\mathcal{A} \models_{s'} \theta$ (porquê?). E como v não ocorre em θ, então $\mathcal{A} \models_s \theta$.

O caso $\forall$**I** é semelhante e os restantes mais simples.

$\square$

Leituras suplementares [BE00] (Cap. 13,18.3)

2.5.2 Conjuntos consistentes e inconsistentes

Definição 2.13. *Um conjunto de fórmulas Σ diz-se* consistente *se e só se não existe nenhuma dedução de $\Sigma \vdash \boldsymbol{F}$, caso contrário diz-se* inconsistente.

Nota que se Σ é inconsistente, $\Sigma \vdash \psi$ para qualquer fórmula ψ.

Exercício 2.19. *Mostra que um conjunto Σ é inconsistente se e só se existe uma fórmula ϕ tal que $\Sigma \vdash \phi$ e $\Sigma \vdash \neg\phi$.* $\diamond$

Lema 2.1. *Se $\Sigma \cup \{\neg\phi\}$ é inconsistente se e só se $\Sigma \vdash \phi$.*

Demonstração. ($\Rightarrow$) Por aplicação da regra **RA** e ($\Leftarrow$) pela aplicação da regra **FI**. $\square$

Lema 2.2. ***(da dedução)*** *$\Sigma \cup \{\phi\} \vdash \psi$ se e só se $\Sigma \vdash \phi \to \psi$.*

Demonstração. Igual ao da lógica proposicional. $\square$

2.5.3 Completude do sistema dedutivo DN

Vamos considerar apenas o caso de $\mathcal{L}$ ser uma linguagem numerável, isto é, em que o seu alfabeto (de variáveis, símbolos funcionais e de predicado) é numerável (não necessariamente finito).

Teorema 2.1. *(Teorema da completude de Gödel) Se $\Sigma \models \phi$ então $\Sigma \vdash \phi$, ou,* equivalentemente*, qualquer conjunto consistente de fórmulas é satisfazível.*

Começamos por mostrar a equivalência entre essas duas afirmações:
(1$\Leftarrow$ 2) Se $\Sigma \models \phi$ então $\Sigma \cup \{\neg\phi\}$ não é satisfazível. Por 2, $\Sigma \cup \{\neg\phi\}$ é inconsistente, logo $\Sigma \vdash \phi$.
(1$\Rightarrow$ 2) Se $\Sigma \cup \{\phi\}$ é não satisfazível, então $\Sigma \models \neg\phi$. Por 1, $\Sigma \vdash \neg\phi$ e então $\Sigma \cup \{\neg\neg\phi\}$ é inconsistente, mas então também o é $\Sigma \cup \{\phi\}$.
Iremos mostrar a segunda afirmação...
Temos que encontrar uma estrutura e interpretação que satisfaça um qualquer conjunto consistente..o que não parece tarefa fácil. A demonstração a apresentar é baseada na de L. Henkin (e não na de Gödel...).
A ideia é construir uma estrutura cujo domínio seja o conjunto dos termos da linguagem $\mathcal{L}$. Mas a presença de quantificadores traz dificuldades acrescidas. Por exemplo,

$$\Delta = \{\exists x P(x)\} \cup \{\neg P(t) : t \text{ é um termo de} \mathcal{L}\}$$

é consistente mas não é possível satisfazê-lo com um domínio só com termos de $\mathcal{L}$. E também é necessário relacionar a satisfazibilidade de $P(t)$ e a de $\exists x P(x)$ (o que não é possível, p.e, só com a validade booleana das fórmulas...)
Assim iremos introduzir novas constantes em $\mathcal{L}$, que irão servir para construir **testemunhas** (de uma fórmula existencial) p.e.:

$$\exists x\ P(x) \rightarrow P(c_P)$$

E iremos alargar qualquer conjunto consistente de modo a conter todas as *testemunhas* necessárias e a continuar consistente...
Dizemos que um conjunto consistente de fórmulas Σ de uma linguagem $\mathcal{L}$ é **maximal** se para toda a fórmula ϕ de $\mathcal{L}$ se tem ou $\phi \in \Sigma$ ou $\neg\phi \in \Sigma$.

Lema 2.3. *Para qualquer conjunto consistente de fórmulas Σ existe um conjunto $\Delta \supseteq \Sigma$, da linguagem alargada $\mathcal{L}' = \mathcal{L} \cup \{c_0, c_1, \ldots\}$ onde c_i são constantes novas e*

1. *Δ é consistente e maximal em $\mathcal{L}'$*

2. *Para qualquer fórmula* ϕ *de* $\mathcal{L}'$ *e qualquer variável* $x \in Var$ *existe uma constante* c *tal* $\exists x\phi \to \phi[c/x] \in \Delta$

Demonstração. (do lema 2.3)

Σ é consistente em $\mathcal{L}'$ Suponhamos, por contradição que $\Sigma \vdash \mathbf{F}$ (ou alternativamente uma qualquer fórmula da forma $\beta \ \wedge \ \neg\beta$). Mas nessa dedução só pode ocorrer um número finito de constantes novas, $c_1, \ldots c_k$. Então podemos substituir cada uma dessas constantes por uma **variável nova**, sejam $y_1, \ldots, y_k$. A dedução resultante é uma dedução de $\mathbf{F}$ a partir de Σ em $\mathcal{L}$. Absurdo! Porque Σ é consistente em $\mathcal{L}$.

Vamos construir Θ tal que $\Sigma \cup \Theta$ satisfaz 2 Como $\mathcal{L}'$ é numerável, então também o são Var e o conjunto das suas fórmulas. Podemos então enumerar $< \phi_1, x_1 >, < \phi_2, x_2 >, \ldots$ Seja θ_1 a fórmula $\exists x_1\phi_1 \to \phi_1[c_1/x_1]$, onde c_1 é uma constante nova que não ocorre em ϕ_1. Para $n > 1$, θ_n é $\exists x_n\phi_n \to \phi_n[c_n/x_n]$, onde c_n é uma constante nova que não ocorre nem ϕ_n nem em $\theta_1 \ldots \theta_{n-1}$. Seja

$$\Theta = \{\theta_1, \theta_2, \ldots\}$$

$\Sigma \cup \Theta$ é consistente Suponhamos que não, então existe $m \geq 0$ tal que $\Sigma \cup \{\theta_1, \ldots, \theta_{m+1}\}$ é inconsistente e $\Sigma \cup \{\theta_1, \ldots, \theta_m\}$ consistente. Mas, então

$$\Sigma \cup \{\theta_1, \ldots, \theta_m\} \vdash \neg\theta_{m+1}$$

Como θ_{m+1} é da forma $\exists x\phi \to \phi[c/x]$, então também (mostra!)

$$\Sigma \cup \{\theta_1, \ldots, \theta_m\} \vdash \exists x\phi$$
$$\Sigma \cup \{\theta_1, \ldots, \theta_m\} \vdash \neg\phi[c/x]$$

mas como c não ocorre em $\Sigma \cup \{\theta_1, \ldots, \theta_m\}$, podemos substituí-lo por uma variável nova e usando a regra $\forall\mathbf{I}$, concluímos também que

$$\Sigma \cup \{\theta_1, \ldots, \theta_m\} \vdash \forall x\neg\phi$$

mas como $\forall x \ \neg\phi \dashv\vdash \neg\exists x\phi$, vem

$$\Sigma \cup \{\theta_1, \ldots, \theta_m\} \vdash \neg\exists x\phi$$

Absurdo! porque $\Sigma \cup \{\theta_1, \ldots, \theta_m\}$ é consistente.

Estendemos $\Sigma \cup \Theta$ a um conjunto consistente maximal Δ Seja $\phi_1, \phi_2, \ldots$ uma enumeração das fórmulas de $\mathcal{L}'$ e definimos

$$\Delta = \bigcup_n \Delta_n$$

onde

$$\begin{array}{rcl} \Delta_0 & = & \Sigma \cup \Theta \\ \Delta_{n+1} & = & \begin{cases} \Delta_n \cup \{\phi_{n+1}\} & \text{se este conjunto é consistente} \\ \Delta_n \cup \{\neg\phi_{n+1}\} & \text{caso contrário} \end{cases} \end{array}$$

Mostra-se por indução sobre n que os Δ_n são consistentes (mostrar!). Logo Δ também é consistente e maximal.

□

Exercício 2.20. *Se Δ é um conjunto consistente maximal e $\Delta \vdash \phi$ então $\phi \in \Delta$* $\diamond$

Para um conjunto de fórmulas Δ do lema anterior, vamos construir uma estrutura $\mathcal{A}_\Delta = (A, \cdot^{\mathcal{A}_\Delta})$ e uma interpretação s_Δ tal que

$$\mathcal{A}_\Delta \models_{s_\Delta} \psi, \text{ para todo } \psi \in \Delta$$

Considere-se o conjunto de termos $\mathcal{T}$ e seja $\sim$ a relação binária dada por

$$t_1 \sim t_2 \text{ se e só se } t_1 = t_2 \in \Delta$$

Pela definição de Δ o fecho reflexivo de $\sim$ é uma relação de equivalência em $\mathcal{T}$, que designamos por $\sim_\Delta$

A estrutura $\mathcal{A}_\Delta$ é:

- o domínio é o conjunto das classes de equivalência de $\sim_\Delta$
- para cada símbolo relacional $R \in \mathcal{R}_n$, $n > 1$:

$$R^{\mathcal{A}_\Delta} = \{([t_1], \ldots, [t_n]) \mid R(t_1, \ldots, t_n) \in \Delta\}$$

- para cada símbolo funcional $f \in \mathcal{F}_n$, $n > 0$ tem-se:

$$f^{\mathcal{A}_\Delta}([t_1], \ldots, [t_n]) = [f(t_1, \ldots, t_n)]$$

- para cada constante c:

$$c^{\mathcal{A}_\Delta} = [c]$$

Para cada $x \in Var$, $s_\Delta(x) = [x]$

Lema 2.4. *Para qualquer termo $t \in \mathcal{T}$, tem-se que $s_\Delta(t) = [t]$*

Demonstração. Por indução sobre t.

Base. Para as variáveis e constantes por definição de s_Δ.

Indução. Se $f \in \mathcal{F}_n$ e $t_1, \ldots, t_n \in \mathcal{T}$,

$$\begin{aligned} s_\Delta(f(t_1, \ldots, t_n)) &= f^{\mathcal{A}_\Delta}(s_\Delta(t_1), \ldots, s_\Delta(t_n)) \\ &= f^{\mathcal{A}_\Delta}([t_1], \ldots, [t_n]) \\ &= [f(t_1, \ldots, t_n)] \end{aligned}$$

□

Lema 2.5. *Seja Δ um conjunto de fórmulas nas condições do lema 2.3. Então*

$$\mathcal{A}_\Delta \models_{s_\Delta} \psi \quad \textit{se e só se } \psi \in \Delta$$

Demonstração. Por indução sobre ψ.

Base. ψ é atómica. Se ψ é $t_1 = t_2$, então $\mathcal{A}_\Delta \models_{s_\Delta} t_1 = t_2$ sse $s_\Delta(t_1) = s_\Delta(t_2)$ sse $[t_1] = [t_2]$, i.e, $t_1 = t_2 \in \Delta$. Se ψ é $R(t_1, \ldots, t_n)$ então $\mathcal{A}_\Delta \models_{s_\Delta} R(t_1, \ldots, t_n)$ sse $(s_\Delta(t_1), \ldots, s_\Delta(t_n)) \in R^{\mathcal{A}_\Delta}$ i.e $([t_1], \ldots, [t_n]) \in R^{\mathcal{A}_\Delta}$ ou $R(t_1, \ldots, t_n) \in \Delta$.

Indução. Se é ψ é $\neg\phi$, então $\mathcal{A}_\Delta \models_{s_\Delta} \neg\phi$ sse $\mathcal{A}_\Delta \not\models_{s_\Delta} \phi$, e por hipótese de indução $\phi \notin \Delta$, logo por Δ ser consistente maximal, $\neg\phi \in \Delta$. Se ψ é $\phi_1 \lor \phi_2$ então,$\mathcal{A}_\Delta \models_{s_\Delta} \phi_1 \lor \phi_2$ sse $\mathcal{A}_\Delta \models_{s_\Delta} \phi_i$ pelo menos para um i, e por hipótese de indução $\phi_i \in \Delta$, mas então por Δ ser consistente maximal, $\phi_1 \lor \phi_2 \in \Delta$. Analogamente para ψ é $\phi_1 \land \phi_2$. Se ψ é $\phi \to \theta$ então, $\mathcal{A}_\Delta \models_{s_\Delta} \phi \to \theta$ sse $\mathcal{A}_\Delta \not\models_{s_\Delta} \phi$ ou $\mathcal{A}_\Delta \models_{s_\Delta} \theta$, e por hipótese de indução ou $\phi \notin \Delta$ ou $\theta \in \Delta$,mas então por Δ ser consistente maximal, $\phi \to \theta \in \Delta$. Se ψ é $\forall x\phi$, tem-se que $\exists x\neg\phi \to \neg\phi[c/x] \in \Delta$, para algum c. Logo

$$\begin{array}{rrl} \forall x\phi \notin \Delta & \textit{sse} & \exists x\neg\phi \in \Delta \\ & \textit{sse} & \neg\phi[c/x] \in \Delta \\ & \textit{sse} & \phi[c/x] \notin \Delta \\ & \textit{sse} & \mathcal{A}_\Delta \not\models_{s_\Delta} \phi[c/x] \\ & \textit{donde} & \mathcal{A}_\Delta \not\models_{s_\Delta} \forall x\phi \end{array}$$

Reciprocamente, se $\mathcal{A}_\Delta \not\models_{s_\Delta} \forall x\phi$, existe t tal que $\mathcal{A}_\Delta \not\models_{s_\Delta[[t]/x]} \phi$. Existe θ equivalente a ϕ e onde x é substituível por t em θ, logo $\mathcal{A}_\Delta \not\models_{s_\Delta[[t]/x]} \theta$ e $\mathcal{A}_\Delta \not\models_{s_\Delta} \theta[t/x]$. Por hipótese de indução, $\theta[t/x] \notin \Delta$ e portanto $\forall x\theta \notin \Delta$. E como θ e ϕ são equivalentes e Δ maximal, $\forall x\phi \notin \Delta$. Analogamente se ψ é $\exists x\phi$.

□

Demonstração. (**Teorema da completude** 2.1) Mostrar que dado um conjunto consistente Σ ele é satisfazível. Estendemos $\Sigma \in \Delta$ nas condições do lema 2.3 para $\mathcal{L}'$ e sejam $\mathcal{A}_\Delta$ e s_Δ a estrutura e a interpretação que satisfazem Δ em $\mathcal{L}'$, pelo lema 2.5. A restrição $\mathcal{A}_\Sigma$ de $\mathcal{A}_\Delta$ a $\mathcal{L}$ satisfaz Σ. □

2.5.4 Consequências da completude e integridade

Corolário 2.2. ***(Teorema da compacidade)***

1. *$\Sigma \models \phi$ se e só se existe um subconjunto finito $\Sigma_0 \subseteq \Sigma$ tal que $\Sigma_0 \models \phi$.*

2. *Um conjunto de fórmulas duma linguagem de $1^{\underline{a}}$ ordem é satisfazível se e só se todo o seu subconjunto finito o for.*

Demonstração. As duas afirmações são equivalentes (verifica!), portanto basta demonstrar a primeira. Se $\Sigma \models \phi$ então $\Sigma \vdash \phi$. Como qualquer dedução só utiliza um número finito de premissas (hipóteses), existe $\Sigma_0 \subseteq \Sigma$ finito tal que $\Sigma_0 \vdash \phi$ ou seja $\Sigma_0 \models \phi$. □

Vamos ver uma aplicação deste teorema que mostra as limitações da expressividade da lógica de primeira ordem. Em particular, que não existe um conjunto de fórmulas que caracterize na estrutura dos números naturais.

Exemplo 2.9. *Modelos não* standard *para a aritmética*
Seja $\mathcal{L}_N$ a linguagem de $1^{\underline{a}}$ ordem com igualdade para os números naturais já dada: $\mathcal{F}_0 = \{0, 1\}$, $\mathcal{F}_2 = \{+, \times\}$ e $\mathcal{R}_2 = \{<\}$
Seja $\mathcal{N} = (\mathbb{N}, \cdot^{\mathcal{N}})$ uma estrutura de $\mathcal{L}_N$, onde $\cdot^{\mathcal{N}}$ é definido por:

- $0^{\mathcal{N}} = 0$, $1^{\mathcal{N}} = 1$
- $+^{\mathcal{N}}(n, m) = n + m$, $\times^{\mathcal{N}}(n, m) = n \times m$
- $<^{\mathcal{N}} = \{(n, m) \in \mathbb{N}^2 \mid n < m\}$

Podemos provar, por exemplo que, $\mathcal{N} \models \forall x\ x < x + 1$ (verifica!)
A estrutura $\mathcal{N}$ é designada a estrutura standard *de $\mathcal{L}_N$, pois $\mathcal{L}_N$ foi construída para se poder falar das propriedades dos números naturais. Mas podemos escolher outras estruturas para $\mathcal{L}_N$:*

Por exemplo, para $p > 1$, $\mathcal{N}_p = (\{0, \ldots, p-1\}, \cdot_p^{\mathcal{N}})$ *com*

- $0^{\mathcal{N}_p} = 0$, $1^{\mathcal{N}_p} = 1$
- $+^{\mathcal{N}_p}(n, m) = (n + m) \mod p$
- $\times^{\mathcal{N}_p} = (n \times m) \mod p$

E, neste caso $\mathcal{N}_p \not\models \forall x\, x < x + 1$*(verifica!)*
Isto é, existe uma fórmula da lógica de $1^{\underline{a}}$ *ordem que permite distinguir a estrutura* $\mathcal{N}$ *da* $\mathcal{N}_p$. *No entanto, usando o teorema da compacidade podemos demonstrar que nem sempre é possível* ***distinguir*** *por uma fórmula duas estruturas para a mesma linguagem.*
Seja $\mathcal{N}' = (\mathbb{N} \cup \{n + i \mid n \in \mathbb{N}\}, \cdot^{\mathcal{N}'})$ *a estrutura de* $\mathcal{L}_N$, *onde* $i = \sqrt{-1}$ *e, por exemplo, para o símbolo funcional* $+$ *temos:*

- $+^{\mathcal{N}_p}(n, m) = (n + m)$
- $+^{\mathcal{N}_p}(n + i, m + i) = (n + m) + i$

Vamos ver que $\mathcal{N}'$ *não pode ser distinguido de* $\mathcal{N}$ *por uma proposição de* $\mathcal{L}_N$ *(ou doutra linguagem de* $1^{\underline{a}}$ *ordem).*

Corolário 2.3. *Se* Σ *é um conjunto de proposições tal que* $\mathcal{N} \models \Sigma$ *então existe um modelo* $\mathcal{N}'$ *tal que* $\mathcal{N}' \models \Sigma$ *e o domínio de* $\mathcal{N}$ *é um subconjunto próprio de* $\mathcal{N}'$.

Demonstração. Considera as proposições ϕ_n dadas por

$$\exists x((x \neq 0) \wedge (x \neq 1) \wedge \ldots \wedge (x \neq n))$$

Então o conjunto $\Sigma \cup \{\phi_n \mid n \geq 0\}$ é consistente. Porque se não o fosse, teria um subconjunto finito que era inconsistente. Esse conjunto finito conteria um número finito dos ϕ_n. Mas obviamente $\mathcal{N}$ satisfaz Σ e qualquer conjunto finito de ϕ_n. Absurdo!
Portanto o conjunto $\Sigma \cup \{\phi_n \mid n \geq 0\}$ é consistente, logo é satisfazível. Isto é, tem de existir uma estrutura cujo domínio seja um superconjunto do de $\mathcal{N}$ e que seja um modelo desse conjunto, p.e, $\mathcal{N}'$. □

Corolário 2.4. ***(Teorema de Löwenheim-Skolem I)*** *Se um conjunto de fórmulas* Σ *é satisfazível por uma estrutura, então* Σ *é satisfazível por uma estrutura com um domínio numerável.*

Demonstração. Porque o domínio da estrutura construída na demonstração do teorema da completude é numerável. □

Sendo um modelo numerável ele pode ser finito ou infinito. Mas será que todas as proposições têm um modelo numerável infinito?
Sejam, por exemplo, $\forall x \forall y x = y$ ou $\exists x \exists y \forall z (z = x \ \vee \ z = y)$.
Estas proposições não têm modelos infinitos! A primeira não aceita modelos com domínios de cardinalidade maior que 1, e a segunda maior que 2.

Corolário 2.5. ***(Teorema de Löwenheim-Skolem II)*** *Se uma proposição tem um modelo finito de cardinalidade arbitrariamente grande, então tem um modelo infinito.*

Demonstração. Seja a proposição ψ_k, para $k > 1$

$$\exists x_1 \ldots \exists x_k \bigwedge_{1 \leq i \leq j \leq k} x_i \neq x_j$$

ψ_k indica que existem pelo menos k objectos diferentes no domínio; não pode ser satisfeita por uma estrutura com menos de k elementos e todas as estruturas com mais elementos a satisfazem. Suponhamos, por contradição, que existe uma proposição ϕ que tem modelos arbitrariamente grandes, mas nenhum modelo infinito. Seja o conjunto

$$\Sigma = \{\phi\} \cup \{\psi_k \mid k \in \mathbb{N} \setminus \{0, 1\}\}$$

Se Σ tem um modelo M, então M não pode ser finito (seja k, então ψ_{k+1} não era satisfeita), e não pode ser infinito (satisfaria ϕ). Então, Σ não tem modelo.
Pela compacidade, existe um conjunto finito $D \subset \Sigma$ que não tem modelo. D tem de conter ϕ (senão haveria um modelo suficientemente grande que continha todos os ψ_k de D). Seja k o maior inteiro tal que $\psi_k \in D$. Por hipótese, ϕ tem um modelo finito de cardinalidade maior que k. Então esse modelo satisfaz todas as proposições de D. Absurdo! □

Este teorema permite ilustrar mais limitações da expressividade da lógica de 1ª ordem. Em particular, temos:

Corolário 2.6. *Não existe nenhuma fórmula ϕ (com duas variáveis livres) tal que saber se existe uma estrutura $\mathcal{G}$ da linguagem $\mathcal{L}_G$, tal que $\mathcal{G} \models \phi$ equivale a determinar se:*

> *Dado um grafo dirigido (finito) G e dois nós x e y de G, existe um caminho de x para y.*

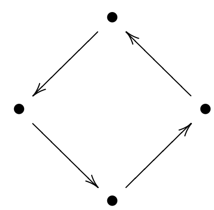
Demonstração. Suponhamos que existe uma tal fórmula ϕ. Seja ψ_0 a fórmula $\forall x \forall y \phi$. A fórmula ψ_0 indica que G é fortemente conexo. Seja ainda ψ_1 a fórmula (que corresponde a *todos os nós têm grau de saída 1*)

$$\forall x \exists y G(x, y) \ \wedge \ \forall x \forall y \forall z ((G(x, y) \ \wedge \ G(x, z)) \rightarrow y = z)$$

e ψ_2 a fórmula (que corresponde a *todos os nós têm grau de entrada 1*)

$$\forall x \exists y G(y,x) \ \wedge \ \forall x \forall y \forall z((G(y,x) \ \wedge \ G(z,x)) \rightarrow y = z)$$

E seja ψ a proposição $\psi_0 \ \wedge \ \psi_1 \ \wedge \ \psi_2$. Os grafos que satisfazem ψ dizem-se *ciclos*.



Obviamente, existem *ciclos* finitos com número arbitrário de nós. Pelo teorema de Löwenheim-Skolem, ψ tem um modelo infinito, seja G_∞. Mas ciclos infinitos (i.e com um número infinito de nós) não existem!: seja n_0 um nó de G_∞ e consideremos todos os nós atingíveis de n_0. Como é fortemente conexo esse conjunto incluí todos os nós do grafo. Mas como o grau de entrada de n_0 é 1, existe um nó, seja n_j tal que (n_j, n_0) é um arco de G_∞. Mas então o ciclo é finito. Absurdo! □

Do mesmo modo, se uma linguagem da lógica de $1^{\underline{a}}$ ordem tivesse um símbolo de predicado que pretendesse significar `é antepassado de`, então não haveria nenhum conjunto de proposições que "capturasse" este conceito pois existiria sempre um modelo que permitiria *antepassados infinitamente distantes....*

Teorema 2.2. ***(da compacidade da lógica proposicional)*** *Um conjunto Σ de fórmulas da lógica proposicional é satisfazível se e só se todo o seu subconjunto finito o for.*

Demonstração. Seja $\mathcal{L}_{Prop}$ uma linguagem de $1^{\underline{a}}$ ordem sem igualdade e um símbolo de predicado unário P, como único símbolo não-lógico. Seja $A = \{P(x_i) \mid i \in \mathbb{N}\}$ o conjunto de fórmulas atómicas de $\mathcal{L}_{Prop}$ e seja $\pi : A \longrightarrow \mathcal{V}_{Prop}$ tal que $\pi(P(x_i)) = p_i$, $i \in \mathbb{N}$. Podemos estender π a uma função bijectiva entre o conjunto das fórmulas de $\mathcal{L}_{Prop}$ sem quantificadores e as fórmulas proposicionais, obtendo a *forma booleana* das primeiras fórmulas. Em particular, um conjunto Σ de fórmulas de $\mathcal{L}_{Prop}$ sem quantificadores é *satisfazível* se e só se o conjunto $\pi(\Sigma)$ das respectivas formas booleanas o for.

($\Leftarrow$) Seja Σ um conjunto de fórmulas proposicionais, tal que todo o subconjunto finito $\Sigma_0 \subseteq \Sigma$ é satisfazível. Então, todo o subconjunto finito de $\pi^{-1}(\Sigma)$ é satisfazível. Pela compacidade, $\pi^{-1}(\Sigma)$ é satisfazível. Mas então também $\pi(\pi^{-1}(\Sigma)) = \Sigma$ é satisfazível

($\Rightarrow$) Trivial. □

Exercício 2.21. *(Aplicação do teorema da compacidade)*

Seja $\mathcal{L}$ uma linguagem de $1^{\underline{a}}$ ordem com igualdade.

1. *Para cada* $n \geq 1$*, construir uma proposição* ϕ_n *de* $\mathcal{L}$ *tal que* $\mathcal{A} \models \phi_n$ *se e só se o domínio de* $\mathcal{A}$ *tem pelo menos* n *elementos.*

   Resolução

   Para cada i*,* ϕ_i *é a fórmula*

   $$\exists x_1 \ldots \exists x_n \bigwedge_{1 \leq i \leq j \leq n} x_i \neq x_j$$

2. *Seja* $\Sigma = \cup_{i \geq 1} \{\phi_i\}$*. Quanto é que uma estrutura de* $\mathcal{L}$ *é um modelo de* Σ*?*

   Resolução

   Quando o seu domínio é infinito (pelo menos numerável)

3. *Suponhamos que* ϕ *é uma proposição tal que* $\mathcal{A}$ *é um modelo de* ϕ *se e só se o domínio de* $\mathcal{A}$ *é infinito. Mostra que* $\Sigma \models \phi$*.*

   Resolução

   Se $\mathcal{A} = (A, \cdot^{\mathcal{A}})$ *satisfaz* Σ *então* A *é um conjunto infinito. Mas então satisfaz* ϕ*.*

4. *Justifica que* ϕ *não pode existir.*

   Resolução

   Pela compacidade, $\Sigma \models \phi$ *se e só se existe* $\Sigma_0 \subseteq \Sigma$ *finito tal que* $\Sigma_0 \models \phi$*. Seja* n_0 *o maior valor tal que* $\phi_{n_0} \in \Sigma_0$*, então qualquer estrutura* $\mathcal{A} = (A, \cdot^{\mathcal{A}})$*, com* $|A| = n_0$ *satisfaz* Σ_0*. Logo,,* $\mathcal{A}$ *satisfaz* ϕ*. Absurdo!*

$\diamond$

Leituras suplementares [BE00] (Cap. 19)

2.6 Axiomatizações e teorias

Normalmente, para além de fórmulas válidas, estamos interessados na satisfazibilidade de fórmulas numa dada estrutura (ou na sua validade numa dada estrutura). Mas para que possamos ter um sistema dedutivo associado, interessa ter um *conjunto de fórmulas* que sejam válidas nessa estrutura e a partir das quais se possam obter todas as consequências semânticas. Essas fórmulas são premissas nos sistemas dedutivos e são designadas de *axiomas não-lógicos*. Para além de válido numa dada estrutura $\mathcal{A}$ (e portanto consistente), um conjunto de axiomas Σ deve ser completo: se uma fórmula ϕ for válida em $\mathcal{A}$ então ϕ é consequência semântica de Σ. Dada a completude e integridade, isso equivale a que para toda a fórmula ϕ, ou ϕ ou $\neg\phi$ seja dedutível de Σ.

Exemplo 2.10. *No* Mundo dos Blocos*, onde há objectos geométricos de três formas possíveis (cubos, tetraedros e dodecaedros), uma axiomatização da noção de* forma *é dada por:*

1. $\neg\exists x(Cubo(x) \wedge Tetra(x))$
2. $\neg\exists x(Cubo(x) \wedge Dodec(x))$
3. $\neg\exists x(Dodec(x) \wedge Tetra(x))$
4. $\forall x(Cubo(x) \vee Dodec(x) \vee Tetra(x))$
5. $\forall x\forall y((Cubo(x) \wedge Cubo(y)) \rightarrow MesmaF(x, y))$
6. $\forall x\forall y((Tetra(x) \wedge Tetra(y)) \rightarrow MesmaF(x, y))$
7. $\forall x\forall y((Dodec(x) \wedge Dodec(y)) \rightarrow MesmaF(x, y))$
8. $\forall x\forall y((Cubo(x) \wedge MesmaF(x, y)) \rightarrow Cubo(y))$
9. $\forall x\forall y((Tetra(x) \wedge MesmaF(x, y)) \rightarrow Tetra(y))$
10. $\forall x\forall y((Dodec(x) \wedge MesmaF(x, y)) \rightarrow Dodec(y))$

As primeiras 3 proposições indicam que um objecto não pode ter duas formas; a quarta que cada objecto tem uma forma; e as restantes indicam que dois objectos têm a mesma forma se e só se são os dois cubos, tetraedros ou dodecaedros.

Exemplo 2.11. ***Teoria dos grupos*** *Considera uma linguagem de* $1^{\underline{a}}$ *ordem (com =), com um símbolo funcional binário* $\circ$ *e uma constante* 1*. A noção de* Grupo *da teoria dos grupos é axiomatizada completamente pelos seguintes axiomas não-lógicos (dos quais se podem deduzir todas as propriedades de um grupo):*

1. $\forall x\forall y\forall z\ (x \circ y) \circ z = x \circ (y \circ z)$
2. $\forall x\ (x \circ 1) = x$
3. $\forall x\exists y\ (x \circ y) = 1$

Exemplos de outras axiomatizações:

- Axiomas da Geometria de Euclides
- Axiomas de Zermello-Frankel para a teoria dos conjuntos
- Axiomas de Peano para a teoria dos números naturais

2.6.1 Teoria *ingénua* dos conjuntos

Um conjunto é uma colecção de elementos. Seja uma linguagem de 1ª ordem com igualdade que tem um símbolo relacional binário $\in$. Para distinguir entre conjuntos e elementos, usaremos variáveis a, b, c, ... para conjuntos e x, y, z, ... para elementos. Consideremos então os seguintes axiomas:

Axioma da extensionalidade $\forall\, a \forall\, b\, (\forall x\, (x \in a \leftrightarrow x \in b) \rightarrow a = b)$

Um conjunto é completamente determinado pelos seus elementos; assim, conjuntos com os mesmos elementos são iguais.

(Esquema de) Axioma da compreensão $\forall \exists a \forall x (x \in a \leftrightarrow \phi(x))$

Cada propriedade determina um conjunto, ie, cada fórmula determina um conjunto: o dos elementos que têm essa propriedade. A quantificação sem variáveis corresponde a quantificar universalmente todas as variáveis que ocorram em ϕ.

A partir destes axiomas, podem-se também definir (e deduzir) fórmulas correspondentes à relação $\subset$, às operações $\cap$ e $\cup$, conjunto potência, ... Por exemplo,

Inclusão $\forall x (x \in a \rightarrow x \in b)$, $(a \subseteq b)$

União $\forall a \forall b \forall z ((z \in a \cup b) \leftrightarrow (z \in a \ \vee\ z \in b))$, $(a \cup b)$

Intersecção $\forall a \forall b \forall z ((z \in a \cap b) \leftrightarrow (z \in a \ \wedge\ z \in b))$, $(a \cap b)$

Conjunto potência $\forall b \exists c \forall x (x \in c \leftrightarrow x \subseteq b)$, $(\mathcal{P}(b))$ (para qualquer conjunto existe um único conjunto cujos elementos são os subconjuntos desse conjunto)

e também se podem deduzir propriedades destas operações.

Proposição 2.11. *Para qualquer b, é falso que $P(b) \subseteq b$.*

Demonstração. Pretende-se provar que, para qualquer b, $P(b) \not\subseteq b$, isto é, que existe um subconjunto de b que não pertence a b. Seja

$$c = \{x \mid x \in b \ \wedge\ x \notin x\}$$

Pelo axioma da compreensão c é um conjunto e $c \subset b$, logo $c \in P(b)$. Vamos ver que $c \notin b$. Suponhamos que $c \in b$. Então ou $c \in c$ ou $c \notin c$. Mas facilmente se vê que ambas as hipóteses não se podem verificar. $\square$

Proposição 2.12. *Existe um conjunto c, tal que $P(c) \subseteq c$.*

Demonstração. Pelo axioma da compreensão existe um conjunto (universal) que contém tudo:

$$c = \{x \mid x = x\}$$

Mas então qualquer subconjunto de c é um elemento de c, portanto $P(c) \in c$. □

Acabámos de ver que se podem deduzir as fórmulas $\forall b \neg (P(b) \subseteq b)$ e $\exists b P(b) \subseteq b$
Donde os axiomas são **inconsistentes**! (Logo não satisfazíveis...)
Esta contradição está relacionada com o paradoxo de Russell:

$$Z = \{x \mid x \notin x\} \text{ e considerar } Z \in Z$$

Podemos então concluir que o **Axioma da compreensão** não é válido: a propriedade *não pertencer a si próprio* não determina nenhum conjunto.

2.6.2 Teoria de conjuntos de Zermelo-Frankel

Para evitar a inconsistência da teoria ingénua dos conjuntos, os axiomas usualmente adoptados para a teoria dos conjuntos são os de Zermelo-Frankel. Mantém-se o axioma da extensionalidade mas o axioma da compreensão é substituído por vários outros, que evitam a possibilidade de referência a *conjuntos tão grandes* como o *conjunto de todos os conjuntos*.

Axioma da extensionalidade $\forall\, a \forall\, b\ (\forall x\ (x \in a \leftrightarrow x \in b) \rightarrow a = b$

Axioma da separação $\forall \forall a \exists b \forall x\ (x \in b \leftrightarrow (x \in a\ \wedge\ \phi(x))$ Esta é a versão mais fraca do Axioma da compreensão. Só se podem formar subconjuntos de conjuntos já existentes. Mas este axioma, embora torne a teoria consistente, é muito restritivo. Nem sequer permite deduzir que a reunião de dois conjuntos é um conjunto. Assim foi necessário acrescentar mais axiomas.

Axioma dos pares $\forall u \forall v \exists a \forall x (x \in a \leftrightarrow (x = u\ \vee\ x = v))$ (para cada dois elementos existe um conjunto a que os dois pertencem)

Axioma da união $\forall a \exists b \forall x (x \in b \leftrightarrow \exists c (c \in a\ \wedge\ x \in c))$

Axioma da potência $\forall b \exists c \forall x (x \in c \leftrightarrow x \subseteq b)$ (conjunto potência)

Axioma da infinitude Existe o conjunto de todos os números naturais.

Axioma da substituição Se $\forall a (\forall x (x \in a\ \wedge\ (\exists! y \phi(x, y))) \rightarrow \exists b \forall x (x \in b \leftrightarrow \exists t\ (t \in a\ \wedge\ \phi(t, x))))$

Axioma da regularidade $\forall z(z \neq \emptyset \rightarrow \exists y(y \in z \wedge \forall x(x \in z \rightarrow x \notin y)))$ (nenhum conjunto tem uma intersecção não vazia com cada um dos seus elementos)

Axioma da escolha Se f é uma função com um domínio a não vazio e para $x \in a$, $f(x)$ é um conjunto não vazio então existe uma função g de domínio a tal que para cada $x \in a$, $g(x) \in f(x)$. (g escolhe elementos de cada $f(x)$)

2.6.3 Axiomas para a teoria dos números (aritmética)

Seja uma linguagem de 1ª ordem com igualdade, $\mathcal{F}_0 = \{0, 1\}$ e $\mathcal{F}_2 = \{+, \times\}$. Podemos omitir o operador relacional $\leq$ uma vez que essa noção se pode definir a partir da igualdade:
$x \leq y$ é $\exists z x + z = y$
$x < y$ é $\exists z(x + z = y \wedge z \neq 0)$
Já vimos que a esta linguagem permite definir fórmulas que caracterizam propriedades dos números naturais, i.e, da estrutura $\mathcal{N} = (\mathbb{N}, .^{\mathcal{A}})$ onde as interpretações dos símbolos não-lógicos correspodem às operações aritméticas usuais.
Alguns desses conceitos podem-se exprimir pelas seguintes fórmulas:

Divisão inteira: $x = q \times y + r \wedge r < y =_{def} INTDIV(x, y, q, r)$

y **divide** x**:** $\exists q INTDIV(x, y, q, 0) =_{def} DIV(y, x)$

$2 :\ 1 + 1 =_{def} 2$

x **é par:** $DIV(2, x) =_{def} Par(x)$

x **é ímpar:** $\neg Par(x) =_{def} Impar(x)$

x **é primo:** $\neg x < 2 \wedge \forall y(DIV(y, x) \rightarrow (y = 1 \vee y = x)) =_{def} Primo(x)$

x **é potência de 2:** $\forall y((DIV(y, x) \wedge Primo(x)) \rightarrow y = 2) =_{def} P(2, x)$

y **é** 2^k **e o** k**-ésimo bit de** x **é** 1**:** $P(2, y) \wedge \forall q \forall r(INTDIV(x, y, q, r) \rightarrow Impar(q)) =_{def} BIT(x, y)$

Os axiomas de Peano (**PA**) são factos básicos dos números naturais:

1. $\forall x(x + 1 \neq 0)$ (0 não é o sucessor de nenhum natural)

2. $\forall x \forall y(x + 1 = y + 1 \rightarrow x = y)$ (o sucessor é injectivo)

3. $0 + 1 = 1$

4. $\forall x\ x + 0 = x$ (0 é identidade para +)

5. $\forall x \forall y\ x + (y + 1) = (x + y) + 1$ (+ é associativo)

6. $\forall x\ x \times 0 = 0$ (0 é absorvente de $\times$)

7. $\forall x \forall y\ x \times (y + 1) = (x \times y) + x$ (distributividade)

8. (princípio da indução) $(Q(0)\ \wedge\ (\forall x(Q(x) \to Q(x + 1)) \to \forall x Q(x)$

É fácil de verificar que os axiomas de Peano são válidos em $\mathcal{N}$. Assim, pela integridade da dedução natural, todas as proposições ϕ tal que $\mathbf{PA} \vdash \phi$, são válidas em $\mathcal{N}$.

Exercício 2.22. *Mostrar que os axiomas de Peano são válidos em* $\mathcal{N}$. $\diamond$

Exercício 2.23. *Mostrar que* $\mathbf{PA} \vdash \forall x(x + 1 = 1 + x)$. $\diamond$

Resolução 2.23

1	$\forall x(x + 1 \neq 0)$	
2	$\forall x \forall y(x + 1 = y + 1 \to x = y)$	
3	$0 + 1 = 1$	
4	$\forall x\ x + 0 = x$	
5	$\forall x \forall y\ x + (y + 1) = (x + y) + 1$	
6	$\forall x\ x \times 0 = 0$	
7	$\forall x \forall y\ x \times (y + 1) = (x \times y) + x$	
8	$(Q(0)\ \wedge\ (\forall x(Q(x) \to Q(x + 1)))) \to \forall x Q(x)$	
9	$1 + 0 = 1$	$\forall$E, 4
10	$0 + 1 = 1 + 0$	$=$**I**, 3, 9
11	u | | $u + 1 = 1 + u$	
12	| | $\forall y\ 1 + (y + 1) = (1 + y) + 1$	$\forall$E, 7
13	| | $1 + (u + 1) = (1 + u) + 1$	$\forall$E, 7, 12
14	| | $1 + u = u + 1$	$=$ simetria, 11
15	| | $1 + (u + 1) = (u + 1) + 1$	$=$**E**, 14, 13
16	| | $(u + 1) + 1 = 1 + (u + 1)$	$=$ simetria, 15
17	| $u + 1 = 1 + u \to (u + 1) + 1 = 1 + (u + 1)$	$\to$I, 11–16
18	$\forall x(x + 1 = 1 + x \to (x + 1) + 1 = 1 + (x + 1))$	$\forall$I, 11–17
19	$0 + 1 = 1\ \wedge\ \forall x(x + 1 = 1 + x \to (x + 1) + 1 = 1 + (x + 1))$	$\wedge$I, 10, 18
20	$\forall x(x + 1 = 1 + x)$	$\to$E, 8, 19

2.6.4 Teorias da lógica de 1ª ordem

Apresentamos em seguida uma formalização das noções de axiomatização e teoria.

Definição 2.14. *Uma teoria $\mathcal{T}$ é um conjunto de proposições de uma linguagem $\mathcal{L}$. Uma teoria duma linguagem $\mathcal{L}$ diz-se (formalmente)* completa *se para qualquer proposição ϕ de $\mathcal{L}$ ou ϕ ou $\neg\phi$ é deduzível de $\mathcal{T}$.*

Definição 2.15. *(**Teoria duma estrutura**) Dada uma estrutura $\mathcal{A}$ duma linguagem $\mathcal{L}$, o conjunto de todas as proposições válidas em $\mathcal{A}$ denomina-se teoria da estrutura e denota-se por $Th(\mathcal{A})$.*

Definição 2.16. *(**Axiomatização**) Uma* axiomatização *de uma estrutura $\mathcal{A}$ é um conjunto de proposições Σ válido em $\mathcal{A}$, i.e, tal que $\mathcal{A} \models \Sigma$. Uma axiomatização de $\mathcal{A}$ é* completa *se para todo $\phi \in Th(\mathcal{A})$, $\Sigma \models \phi$ (e também $\Sigma \vdash \phi$).*

Para a aritmética, p.e, pretendia-se uma axiomatização completa, isto é, em que fosse possível deduzir todas e só as proposições que eram verdadeiras em $\mathbb{N}$, isto é, na estrutura $\mathcal{N} = (\mathbb{N}, .^{\mathcal{A}})$. Já vimos que a axiomatização de Peano é íntegra.

Leituras suplementares [BE00] (Cap. 15,16.4)

2.7 Outros sistemas dedutivos

2.7.1 Sistemas dedutivos de Hilbert, H

Supondo apenas o conjunto completo de conectivas $\{\neg, \rightarrow\}$ e uma linguagem com igualdade:

Axiomas

- $\phi \rightarrow (\psi \rightarrow \phi)$
- $(\phi \rightarrow (\psi \rightarrow \theta)) \rightarrow ((\phi \rightarrow \psi) \rightarrow (\phi \rightarrow \theta))$
- $(\neg\psi \rightarrow \neg\phi) \rightarrow ((\neg\psi \rightarrow \phi) \rightarrow \psi)$
- $\forall x\phi \rightarrow \phi[t/x]$, onde x é substituível por t em ϕ
- $\forall x(\phi \rightarrow \psi) \rightarrow (\phi \rightarrow \forall x\psi)$, onde x não ocorre livre em ϕ
- $x = x$
- $x = y \rightarrow (\phi \rightarrow \phi[y/x])$ e ϕ é atómica

Regras de inferência

- *Modus ponens*: de ϕ e de $\phi \rightarrow \psi$, inferir ψ

- *generalização*: para $x \in Var$, inferir $\forall x\phi$ a partir de ϕ

Proposição 2.13. $\Sigma \vdash_{ND} \phi$ *se e só se* $\Sigma \vdash_H \phi$

Demonstração. ($\Leftarrow$): Basta ver que os axiomas de H são teoremas de DN. A regra de inferência *modus ponens* corresponde à regra da eliminação de implicação de DN ($\rightarrow$ **E**) e a regra generalização à regra da introdução do quantificador universal de DN ($\forall$**I**)
($\Rightarrow$): é possível transformar uma dedução em DN, numa dedução em H. (Não o faremos neste curso...) □

2.7.2 Tableaux

Recordemos que as regras proposicionais de expansão de *tableaux* são:

$$\frac{\neg\neg\phi}{\phi} \qquad \frac{\neg\mathbf{F}}{\mathbf{V}} \qquad \frac{\neg\mathbf{V}}{\mathbf{F}} \qquad \frac{\alpha}{\begin{array}{c}\alpha_1\\ \alpha_2\end{array}} \qquad \frac{\beta}{\beta_1 \mid \beta_2}$$

onde

α	α_1	α_2	β	β_1	β_2
$\phi \wedge \psi$	ϕ	ψ	$\neg(\phi \wedge \psi)$	$\neg\phi$	$\neg\psi$
$\neg(\phi \vee \psi)$	$\neg\psi$	$\neg\phi$	$\phi \vee \psi$	ϕ	ψ
$\neg(\phi \rightarrow \psi)$	ϕ	$\neg\psi$	$\phi \rightarrow \psi$	$\neg\phi$	ψ

Exemplo 2.12. *Considerar o* tableaux *para a negação duma fórmula válida:*

$$\begin{array}{ccc}
 & \neg(\forall x(p(x) \rightarrow q(x)) \rightarrow (\forall x p(x) \rightarrow \forall x q(x)))) & \\
 & \forall x(p(x) \rightarrow q(x)) & \\
 & \forall x p(x) & \\
 & \neg\forall x q(x) & \\
 & \neg q(a) & \\
 & p(a) & \\
 & p(a) \rightarrow q(a) & \\
\neg p(a) & & q(a) \\
\times & & \times
\end{array}$$

obtemos um tableaux *fechado, onde a é uma constante, que não aparece na fórmula.*
Para cada quantificador existencial é necessário usar uma constante nova. Por exemplo se considerarmos a negação de:

$$\forall x(p(x) \ \vee \ q(x)) \rightarrow (\forall x p(x) \ \vee \ \forall x q(x))$$

que é satisfazível mas não válida, se não tivermos esse cuidado obtemos um ***tableaux*** *fechado. Verifica!*

$$\neg(\forall x(p(x) \ \vee \ q(x)) \rightarrow (\forall x p(x) \ \vee \ \forall x q(x))$$
$$(\forall x(p(x) \ \vee \ q(x))$$
$$\neg(\forall x p(x) \ \vee \ \forall x q(x))$$
$$\neg\forall x p(x)$$
$$\neg\forall x q(x))$$
$$\neg q(a)$$
$$\neg p(\mathbf{b})$$
$$p(a) \ \vee \ q(a)$$
$$p(b) \ \vee \ q(b)$$

$$\begin{array}{ccccccc} & p(b) & & & & q(b) & \\ q(a) & & p(a) & \qquad\qquad & p(a) & & q(a) \\ \times & & \times & & \times & & \end{array}$$

Notar que um ramo não fechado de um *tableaux* define uma estrutura em que a fórmula é válida...

Exemplo 2.13. *Consideremos ainda um **tableaux** para ϕ igual a $\phi_1 \ \wedge \ \phi_2 \ \wedge \ \phi_3$ onde:*

$$\begin{array}{rcc} \phi_1 & = & \forall x \exists y p(x,y)) \\ \phi_2 & = & \forall x \neg p(x,x)) \\ \phi_3 & = & \forall x \forall y \forall z((p(x,y) \ \wedge \ p(y,z)) \rightarrow p(x,z)) \end{array}$$

Depois de aplicar as regras α obtemos

$$\forall x \exists y p(x,y))$$
$$\forall x \neg p(x,x))$$
$$\forall x \forall y \forall z((p(x,y) \ \wedge \ p(y,z)) \rightarrow p(x,z))$$

Agora não temos nenhuma constante (ou outro termo) para instanciar. Podemos escolher qualquer elemento a_1 e obter $\exists y p(a_1, y)$ a partir de ϕ_1. E depois instanciar o $\exists$ com uma constante nova, a_2. E voltar a instanciar ϕ_1 com a_2, ficando $\exists y p(a_2, y)$. E voltar a instanciar este com uma nova constante a_3....e deste modo temos um processo que não termina...
*Pode-se mostrar que ϕ não tem nenhum modelo **finito**!: Suponhamos que existe $\mathcal{A}$ com donínio finito mas não vazio. Por ϕ_1 existe uma sequência de a_i tal que $\mathcal{A} \models_{s[x/a_i][y/a_j]} p(x,y)$, para todos i e $j = i+1$. Por ϕ_3, $j \neq i$. Mas como o domínio é finito, existe k com $a_k = a_i$. O que contradiz ϕ_2 que obrigaria a $\mathcal{A} \not\models_{s[x/a_i/]} [p(x,x)$.*

E podemos concluir que os **tableaux** não são um processo de decisão para a validade das fórmulas de primeira ordem.

2.7.2.1 Regras de expansão para *tableaux*

A notação uniforme para as fórmulas pode ser extendida para os quantificadores (γ **universais** e δ **existenciais**):

γ	$\gamma(t)$	δ	$\delta(t)$
$\forall x \phi$	$\phi[x/t]$	$\exists x \phi$	$\phi[x/t]$
$\neg \exists x \phi$	$\neg \phi[x/t]$	$\neg \forall x \phi$	$\neg \phi[x/t]$

Nesta notação, as novas regras de expansão dos **tableaux** são:

$$\frac{\gamma}{\gamma(t)} \qquad \frac{\delta}{\delta(p)}$$

onde t é um termo fechado e p é uma constante nova.

Apêndice A

Dedução natural para a lógica proposicional

	Introdução	**Eliminação**
$\wedge$	$\dfrac{\phi \quad \psi}{\phi \wedge \psi} \wedge \mathbf{I}$	$\dfrac{\phi \wedge \psi}{\phi} \wedge \mathbf{E}_1$ $\quad$ $\dfrac{\phi \wedge \psi}{\psi} \wedge \mathbf{E}_2$
$\vee$	$\dfrac{\phi}{\phi \vee \psi} \vee \mathbf{I}_1$ $\quad$ $\dfrac{\psi}{\phi \vee \psi} \vee \mathbf{I}_2$	$\dfrac{\phi \vee \psi \qquad \begin{array}{c}[\phi]\\ \vdots\\ \gamma\end{array} \quad \begin{array}{c}[\psi]\\ \vdots\\ \gamma\end{array}}{\gamma} \vee \mathbf{E}$
$\neg$	$\dfrac{\begin{array}{c}[\phi]\\ \vdots\\ \mathbf{F}\end{array}}{\neg\phi} \neg\mathbf{I}$	$\dfrac{\neg\neg\phi}{\phi} \neg\mathbf{E}$
$\mathbf{F}$	$\dfrac{\begin{array}{c}\phi\\ \vdots\\ \neg\phi\end{array}}{\mathbf{F}} \mathbf{FI}(*)$	$\dfrac{\mathbf{F}}{\phi} \mathbf{FE}$
$\rightarrow$	$\dfrac{\begin{array}{c}[\phi]\\ \vdots\\ \psi\end{array}}{\phi \rightarrow \psi} \rightarrow\mathbf{I}$	$\dfrac{\phi \quad \phi \rightarrow \psi}{\psi} \rightarrow\mathbf{E}$

Regra da Repetição

$$\frac{\phi}{\phi}\,\mathbf{R}$$

Algumas regras derivadas:

$$\frac{\phi \to \psi \qquad \neg\psi}{\neg\phi}\,\mathbf{MT}$$

$$\frac{\phi}{\neg\neg\phi}\,\neg\neg\mathbf{I}$$

$$\frac{\begin{array}{c}[\neg\phi]\\ \vdots \\ \mathbf{F}\end{array}}{\phi}\,\mathbf{RA}$$

$$\frac{}{\phi \;\vee\; \neg\phi}\,\mathbf{TE}$$

Apêndice B

Dedução natural para a lógica de 1ª ordem

	Introdução	**Eliminação**
$\wedge$	$\dfrac{\phi \quad \psi}{\phi \wedge \psi} \wedge \mathbf{I}$	$\dfrac{\phi \wedge \psi}{\phi} \wedge \mathbf{E}_1 \qquad \dfrac{\phi \wedge \psi}{\psi} \wedge \mathbf{E}_2$
$\vee$	$\dfrac{\phi}{\phi \vee \psi} \vee \mathbf{I}_1 \qquad \dfrac{\psi}{\phi \vee \psi} \vee \mathbf{I}_2$	$\dfrac{\phi \vee \psi \qquad \begin{array}{c}[\phi]\\ \vdots \\ \gamma\end{array} \quad \begin{array}{c}[\psi]\\ \vdots \\ \gamma\end{array}}{\gamma} \vee \mathbf{E}$
$\neg$	$\dfrac{\begin{array}{c}[\phi]\\ \vdots \\ \mathbf{F}\end{array}}{\neg\phi} \neg\mathbf{I}$	$\dfrac{\neg\neg\phi}{\phi} \neg\mathbf{E}$
$\mathbf{F}$	$\dfrac{\begin{array}{c}\phi\\ \vdots \\ \neg\phi\end{array}}{\mathbf{F}} \mathbf{FI}(*)$	$\dfrac{\mathbf{F}}{\phi} \mathbf{FE}$
$\rightarrow$	$\dfrac{\begin{array}{c}[\phi]\\ \vdots \\ \psi\end{array}}{\phi \rightarrow \psi} \rightarrow\mathbf{I}$	$\dfrac{\phi \quad \phi \rightarrow \psi}{\psi} \rightarrow\mathbf{E}$
$=$	$\dfrac{}{t = t} = \mathbf{I}$	$\dfrac{t_1 = t_2 \quad \phi[t_1/x]}{\phi[t_2/x]} = \mathbf{E}$ e x é substituível por t_1 e por t_2 em ϕ
	$[v]$	

Regra da Repetição

$$\frac{\phi}{\phi}\mathbf{R}$$

Algumas regras derivadas:

$$\frac{\phi\rightarrow\psi \qquad \neg\psi}{\neg\phi}\mathbf{MT} \qquad \frac{\phi}{\neg\neg\phi}\neg\neg\mathbf{I}$$

$$\begin{array}{c}[\neg\phi]\\ \vdots\\ \mathbf{F}\\ \hline \phi\end{array}\mathbf{RA} \qquad \frac{}{\phi\vee\neg\phi}\mathbf{TE}$$

Bibliografia

[And86] Peter B. Andrews. *An Introduction to Mathematical Logic and Type Theory:To Truth Through Proof.* Academic Press, Orlando, Florida, 1986.

[BA01] Mordechai Ben-Ari. *Mathematical Logic for Computer Science.* SV, 2nd edition, 2001.

[BE00] Jon Barwise and John Etchmendy. *Language, Proof, and Logic.* CSLI, 2000.

[Bro00] Sabine Broda. Apontamentos de lógica computacional. Technical report, Departamento de Ciência de Computadores, FCUP, 2000.

[DLL62] Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem proving. *Communications of the ACM*, 5(7):394–397, July 1962.

[DP60] Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *Journal of the ACM*, 7(3):201–215, July 1960.

[Fit90] Melvin Fitting. *First-Order Logic and Automated Theorem Proving.* SV, 1990.

[GLM97] Jean Goubault-Larrecq and Ian Mackie. *Proof Theory and Automated Deduction.* Kluwer Academic Press, 1997.

[HR00] Michael Huth and Mark Ryan. *Logic in Computer Science: Modelling and reasoning about systems.* CUP, 2000. 430 DCCBIB.

[Llo87] J.W. Lloyd. *Foundations of Logic Programming.* SV, 1987.

[Smu95] Raymond M. Smullyan. *Fisrt-order Logic.* Dover Publications, 1995.

Índice