1. Fundamentos

1.1. Como raciocinamos?: lógica elementar

A lógica é a base de todo o raciocínio. Portanto se quisermos estudar e fazer matemática precisamos de dominar os princípios básicos da lógica. Citando *Language, Proof and Logic* (de Jon Barwise e John Etchemendy):

> "(...) all rational inquiry depends on logic, on the ability of people to reason correctly most of the time."

> "(...) there is an overwhelming intuition that the laws of logic are somehow more irrefutable than the laws of the land, or even the laws of physics."

Todos nós raciocinamos enunciando factos e tirando conclusões baseadas nesses factos. O início de uma conclusão é habitualmente indicada por uma palavra como

Então, Logo, Portanto, Consequentemente, ...

Para chegarmos a uma conclusão aplicamos uma *regra de inferência* (ou *regra de dedução*). A mais comum é a chamada regra *modus ponens* (modo que afirma): sendo A e B afirmações, se A e "se A então B" são ambas verdadeiras, então podemos concluir que B é verdadeira.

`[Como aprendeu a regra modus ponens em criança?]`

Outra regra muito comum é a *modus tollens* (modo que nega): sendo A e B afirmações, se "se A então B" é verdadeira e B é falsa, então podemos concluir que A é falsa.

Por exemplo:

- Se ele foi a Coimbra então visitou a Universidade de Coimbra.
- Ele não visitou a Universidade de Coimbra.
- Logo não foi a Coimbra.

`[Como aprendeu a regra modus tollens em criança?]`

Quando tiramos uma conclusão que não decorre dos factos estabelecidos previamente, o raciocínio diz-se *non sequitur* (que não segue).

`[Pense num exemplo de non sequitur que já tenha observado.]`

Neste primeiro capítulo começaremos por estudar um pouco de lógica. Algumas definições de *lógica* que podemos encontrar nos dicionários:

- Estudo dos princípios do raciocínio, particularmente da estrutura das afirmações e proposições e dos métodos de determinação da sua validade.

- Sistema de raciocínio.
- Raciocínio válido.

Um *cálculo* é uma linguagem de expressões, onde cada expressão tem um valor lógico e há regras para transformar uma expressão noutra com o mesmo valor. Aqui estudaremos um pouco do cálculo proposicional. O *cálculo proposicional* é a linguagem das *proposições*. Uma *proposição* é uma expressão da qual faz sentido dizer que é verdadeira ou que é falsa. Cada proposição tem um e um só valor lógico, entre dois possíveis: V (verdadeiro) ou F (falso).

Exemplo. "Coimbra é uma cidade portuguesa" é uma proposição com valor lógico verdadeiro. Mas atribuir um valor lógico à afirmação "Hoje está um belo dia!" já não faz sentido, pois trata-se duma expressão subjectiva que exprime um sentimento de alguém, não de uma afirmação objectiva.

Lógica e operações-bit: Os computadores representam informação por meio de bits. Um bit tem dois valores possíveis, 0 e 1. Um bit pode ser usado para representar os valores de verdade F e V, 0 representa F e 1 representa V. Há assim uma relação evidente entre a lógica e o sistema de funcionamento dos computadores.

O cálculo proposicional (tal como outros tipos de lógica) que vamos estudar pressupõe os seguintes princípios:

Princípio da não contradição: Uma proposição não pode ser verdadeira e falsa ao mesmo tempo.

Princípio do terceiro excluído: Uma proposição é verdadeira ou falsa.

A afirmação "Os alunos de Estruturas Discretas são de Engenharia Informática ou de Comunicações e Multimédia" pode decompor-se em duas afirmações: "Os alunos de Estruturas Discretas são de Engenharia Informática" e "Os alunos de Estruturas Discretas são de Comunicações e Multimédia". Estas duas últimas afirmações já não se podem decompor mais. Dizemos então que a proposição "Os alunos de Estruturas Discretas são de Engenharia Informática ou de Comunicações e Multimédia" é *composta* e as afirmações "Os alunos de Estruturas Discretas são de Engenharia Informática" e "Os alunos de Estruturas Discretas são de Comunicações e Multimédia" são atómicas. As proposições compostas são construídas a partir de proposições atómicas ligando-as por *conectivos* (ou *operadores*). No exemplo anterior, esse conectivo é "ou". Se denotarmos por p a proposição "Os alunos de Estruturas Discretas são de Engenharia Informática" e por q a proposição "Os alunos de Estruturas Discretas são de Comunicações e Multimédia" e usarmos o símbolo $\vee$ para representar "ou", a afirmação "Os alunos de Estruturas Discretas são de Engenharia Informática ou de Comunicações e Multimédia" escreve-se simplesmente $p \vee q$. Temos assim a operação $\vee$ de *disjunção*.

A afirmação "Ele não visitou a Universidade de Coimbra" é a negação de "Ele visitou a Universidade de Coimbra". Se denotarmos por p esta última proposição e usarmos o símbolo $\neg$ para representar a operação de negação, a afirmação "Ele não visitou a Universidade de Coimbra" escreve-se $\neg p$.

A seguinte tabela contém uma lista destes e doutros conectivos lógicos[1]:

negação	$\neg p$ (não p)
conjunção	$p \wedge q$ (p e q)
disjunção	$p \vee q$ (p ou q)
implicação	$p \rightarrow q$ (se p então q; p só se q; p é condição suficiente para que q; q é condição necessária para que p)
equivalência (formal)	$p \leftrightarrow q$ (p é equivalente a q)
disjunção exclusiva	$p \dot{\vee} q$ (ou p ou q)

As proposições são representadas por fórmulas chamadas *fórmulas bem formadas* que são construídas a partir de um alfabeto constituído por:

- Símbolos de verdade: V e F.
- Variáveis proposicionais: letras do alfabeto $p, q, r, \ldots$
- Conectivos (operadores):

  $\neg$ ("não", negação)

  $\wedge$ ("e", conjunção)

  $\vee$ ("ou", disjunção)

  $\rightarrow$ ("implica", implicação).
- Símbolos de parênteses: (,).

Uma *fórmula bem formada* (abreviadamente, fbf) fica definida da seguinte forma:

- V e F são fbf's; toda a variável proposicional é uma fbf.
- Se A e B são fbf's, as seguintes são também fbf's: $\neg A$, $A \wedge B$, $A \vee B$, $A \rightarrow B$, (A).
- Toda a fbf é construída por aplicação sucessiva das regras anteriores.

Exemplo. A expressão $p \neg q$ não é uma fbf. Mas cada uma das seguintes expressões é uma fbf: $p \wedge q \rightarrow r$, $(p \wedge q) \rightarrow r$, $p \wedge (q \rightarrow r)$.

Os parênteses funcionam como símbolos auxiliares que indicam como é formada a fbf. Para evitar um uso excessivo de parênteses e simplificar a escrita das expressões lógicas convenciona-se que as operações lógicas são consideradas pela seguinte ordem de prioridade: $\neg$, $\wedge$, $\vee$, $\rightarrow$. Convenciona-se ainda que na presença de uma só das três últimas operações, na ausência de parênteses as operações são realizadas da esquerda para a direita.

[1] Em lógica é habitual designar estes conectivos por *conectivos booleanos*, em homenagem ao lógico britânico George Boole (1815-1864), que estudou as leis do pensamento usando métodos matemáticos em [*An investigation into the Laws of Thought*, 1854].

Exemplos.

$$\begin{array}{rll} \neg p \wedge q & \text{significa} & (\neg p) \wedge q \\ p \vee q \wedge r & \text{significa} & p \vee (q \wedge r) \\ p \wedge q \rightarrow r & \text{significa} & (p \wedge q) \rightarrow r \\ p \rightarrow q \rightarrow r & \text{significa} & (p \rightarrow q) \rightarrow r. \end{array}$$

Relativamente a uma dada linguagem lógica podemos sempre estudar dois aspectos: a sintaxe e a semântica. A sintaxe diz respeito às regras de formação das expressões lógicas a utilizar, ou seja, as fórmulas bem formadas. Em cima, acabámos de descrever a sintaxe do cálculo proposicional.

A semântica estuda o significado das expressões.

$$\text{Linguagem (conjunto de símbolos)} \begin{cases} \text{sintaxe (fórmulas bem formadas)} \\ \\ \text{semântica (significado).} \end{cases}$$

Quanto à semântica, dada uma fbf, interpretando cada uma das suas variáveis proposicionais com os valores lógicos V ou F, é possível dar um significado à fórmula através da interpretação dos conectivos lógicos dada pelas respectivas *tabelas de verdade*. Cada conectivo tem uma tabela de verdade (que vai ao encontro da forma corrente do significado das operações "não", "e", "ou", etc.). A tabela de verdade faz corresponder aos possíveis valores lógicos das variáveis o correspondente valor lógico da operação[2]:

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \rightarrow q$
V	V	F	V	V	V
V	F	F	F	V	F
F	V	V	F	V	V
F	F	V	F	F	V

Em conclusão:

- O significado de V é verdade e o de F é falso.
- O significado de qualquer outra fbf é dado pela sua tabela de verdade.

Exemplo. $\neg p \wedge q$ corresponde a $(\neg p) \wedge q$, cuja tabela de verdade é:

p	q	$\neg p$	$(\neg p) \wedge q$
V	V	F	F
V	F	F	F
F	V	V	V
F	F	V	F

[2]Nas aulas teórico-práticas usaremos o software `Boole` para nos ajudar a escrever tabelas de verdade. Consulte o fascículo Usando Boole.

É claro que a cada fbf corresponde uma e uma só tabela de verdade.

Uma fbf diz-se uma *tautologia* se for verdadeira para todos os possíveis valores lógicos das suas variáveis proposicionais. Uma fbf diz-se uma *contradição* se for falsa para todos os possíveis valores lógicos das suas variáveis proposicionais. Uma fbf diz-se uma *contingência* se não for uma tautologia nem uma contradição.

Exemplos. Suponhamos que queremos averiguar se $p \rightarrow p \vee q$ é ou não uma tautologia. Para isso basta construir a respectiva tabela de verdade:

p	q	$p \vee q$	$p \rightarrow p \vee q$
V	V	V	V
V	F	V	V
F	V	V	V
F	F	F	V

Como para quaisquer valores de p e q toma sempre o valor de verdade, concluímos que é uma tautologia.

Mais exemplos: $p \vee \neg p$ é uma tautologia, $p \wedge \neg p$ é uma contradição e $p \rightarrow q$ é uma contingência.

Duas fbf's dizem-se *(logicamente) equivalentes* se tiverem o mesmo significado, isto é, a mesma tabela de verdade. Para indicar que duas fbf's A e B são equivalentes, escrevemos $A \equiv B$. Em vez do símbolo $\equiv$ também se costuma usar $\Leftrightarrow$. Note que dizer que A e B são logicamente equivalentes é o mesmo que dizer que as fórmulas $(A \rightarrow B)$ e $(B \rightarrow A)$ são tautologias (Prova: $A \equiv B$ sse A e B têm os mesmos valores de verdade sse $(A \rightarrow B)$ e $(B \rightarrow A)$ são tautologias).

Exemplos. As seguintes equivalências básicas são de fácil verificação e são fundamentais no cálculo proposicional:

$p \vee \neg p \equiv \mathsf{V}$	Lei do terceiro excluído
$p \wedge \neg p \equiv \mathsf{F}$	Lei da contradição
$p \wedge \mathsf{V} \equiv p$	Leis da identidade
$p \vee \mathsf{F} \equiv p$	
$p \vee \mathsf{V} \equiv \mathsf{V}$	Leis do elemento dominante
$p \wedge \mathsf{F} \equiv \mathsf{F}$	
$p \vee p \equiv p$	Leis da idempotência
$p \wedge p \equiv p$	
$\neg(\neg p) \equiv p$	Lei da dupla negação
$p \vee q \equiv q \vee p$	Leis da comutatividade
$p \wedge q \equiv q \wedge p$	
$p \vee (p \wedge q) \equiv p$	Leis da absorção
$p \wedge (p \vee q) \equiv p$	

$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Leis da associatividade
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Leis da distributividade
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	Leis de De Morgan
$p \rightarrow q \equiv \neg p \vee q$	

Por exemplo, construindo as tabelas de verdade de $\neg(p \wedge q)$ e $\neg p \vee \neg q$

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

concluímos que ambas as fbf's têm o mesmo valor lógico para os mesmos valores das variáveis proposicionais, pelo que são logicamente equivalentes.

É possível provar uma equivalência sem construir as tabelas de verdade por causa dos seguintes factos:

1. Se $A \equiv B$ e $B \equiv C$, então $A \equiv C$.
2. Se $A \equiv B$, então qualquer fbf C que contenha A é equivalente à fbf obtida de C substituindo uma ocorrência de A por B.

Exemplo. Use as equivalências básicas na tabela anterior para provar que $p \vee q \rightarrow p \equiv q \rightarrow p$.

Prova:

$$\begin{aligned} p \vee q \rightarrow p &\equiv \neg(p \vee q) \vee p \\ &\equiv (\neg p \wedge \neg q) \vee p \\ &\equiv (\neg p \vee p) \wedge (\neg q \vee p) \\ &\equiv \mathsf{V} \wedge (\neg q \vee p) \\ &\equiv \neg q \vee p \\ &\equiv q \rightarrow p. \qquad QED. \end{aligned}$$

Teste (1 minuto cada). Use equivalências conhecidas para provar:

1. $p \vee q \rightarrow r \equiv (p \rightarrow r) \wedge (q \rightarrow r)$.
2. $(p \rightarrow q) \vee (\neg p \rightarrow q) \equiv \mathsf{V}$ (isto é, $(p \rightarrow q) \vee (\neg p \rightarrow q)$ é uma tautologia).
3. $p \rightarrow q \equiv (p \wedge \neg q) \rightarrow \mathsf{F}$.

Teste (1 minuto cada). Use as leis da absorção para simplificar:

1. $(p \wedge q \wedge r) \vee (p \wedge r) \vee r$.
2. $(s \rightarrow t) \wedge (u \vee t \vee \neg s)$.

Se p é uma variável proposicional numa fbf A, denotemos por $A(p/\mathsf{V})$ a fbf que se obtém de A substituindo todas as ocorrências de p por V . De modo análogo podemos também definir a fórmula $A(p/\mathsf{F})$. As seguintes propriedades verificam-se:

- A é uma tautologia se e só se $A(p/\mathsf{V})$ e $A(p/\mathsf{F})$ são tautologias.
- A é uma contradição se e só se $A(p/\mathsf{V})$ e $A(p/\mathsf{F})$ são contradições.

O *Método de Quine* usa estas propriedades, conjuntamente com as equivalências básicas, para determinar se uma fbf é uma tautologia, uma contradição ou uma contingência. (Trata-se de um método alternativo à construção das tabelas de verdade.)

Exemplo. Seja A a fórmula $(p \wedge q \rightarrow r) \wedge (p \rightarrow q) \rightarrow (p \rightarrow r)$. Então:

$$\begin{aligned} A(p/\mathsf{F}) &= (\mathsf{F} \wedge q \rightarrow r) \wedge (\mathsf{F} \rightarrow q) \rightarrow (\mathsf{F} \rightarrow r) \\ &\equiv (\mathsf{F} \rightarrow r) \wedge \mathsf{V} \rightarrow \mathsf{V} \\ &\equiv \mathsf{V}. \end{aligned}$$

Portanto $A(p/\mathsf{F})$ é uma tautologia. A seguir olhemos para

$$\begin{aligned} A(p/\mathsf{V}) &= (\mathsf{V} \wedge q \rightarrow r) \wedge (\mathsf{V} \rightarrow q) \rightarrow (\mathsf{V} \rightarrow r) \\ &\equiv (q \rightarrow r) \wedge q \rightarrow r. \end{aligned}$$

Seja $B = (q \rightarrow r) \wedge q \rightarrow r$. Então

$$B(q/\mathsf{V}) = (\mathsf{V} \rightarrow r) \wedge \mathsf{V} \rightarrow r \equiv r \wedge \mathsf{V} \rightarrow r \equiv r \rightarrow r \equiv \mathsf{V}$$

e

$$B(q/\mathsf{F}) = (\mathsf{F} \rightarrow r) \wedge \mathsf{F} \rightarrow r \equiv \mathsf{F} \rightarrow r \equiv \mathsf{V},$$

o que mostra que B é uma tautologia. Portanto, A é uma tautologia.

Teste (2 minutos cada). Use o método de Quine em cada caso:

1. Mostre que $(p \vee q \rightarrow r) \vee p \rightarrow (r \rightarrow q)$ NÃO é uma tautologia.
2. Mostre que $(p \rightarrow q) \rightarrow r$ NÃO é equivalente a $p \rightarrow (q \rightarrow r)$.

No nosso dia a dia raciocinamos e tiramos conclusões usando determinadas regras. A lógica ajuda a compreender essas regras permitindo distinguir entre argumentos correctos e argumentos não correctos. Seguem-se alguns argumentos lógicos, cada um deles com um exemplo e a respectiva formalização.

(1)

1. Se o gato vê o peixe, então o gato apanha o peixe.
2. Se o gato apanha o peixe, então o gato come o peixe.

---

3. Se o gato vê o peixe, então o gato come o peixe.

1. $p \rightarrow q$
2. $q \rightarrow r$

---

3. $p \rightarrow r$

(2)

1. Se o João tem mais de 16 anos, então vai ao cinema.
2. O João tem mais de 16 anos.

---

3. O João vai ao cinema.

1. $p \rightarrow q$
2. p

---

3. q

(3)

1. A Maria vai aos testes ou faz o exame.
2. A Maria não faz o exame.

---

3. A Maria vai aos testes.

1. $p \vee q$
2. $\neg q$

---

3. p

Um argumento da forma "De $A_1, A_2, \ldots,$ e A_n deduz-se B", esquematicamente,

$$\begin{array}{c} A_1 \\ A_2 \\ \vdots \\ A_n \\ \hline B \end{array}$$

diz-se um *argumento correcto* se $A_1 \wedge A_2 \wedge \cdots \wedge A_n \rightarrow B$ for uma tautologia. Neste caso escreve-se

$$A_1, A_2, \ldots, A_n \models B.$$

Para indicar que $A \models B$ também se usa $A \Rightarrow B$, nomenclatura que faz parte do dia a dia da escrita matemática.

Um literal é uma variável proposicional ou a sua negação; por exemplo, p e $\neg p$ são literais (ditos *literais complementares*).

Uma fbf diz-se uma *forma normal disjuntiva* (FND) se for da forma $C_1 \vee C_2 \vee \cdots \vee C_n$, onde cada C_i é uma conjunção de literais (chamada *conjunção fundamental*).

Analogamente, uma fbf diz-se uma *forma normal conjuntiva* (FNC) se for da forma $D_1 \wedge D_2 \wedge \cdots \wedge D_n$, onde cada D_i é uma disjunção de literais (chamada *disjunção fundamental*).

Exemplos de formas normais disjuntivas:

$$p$$

$$\neg p$$

$$p \vee \neg q$$

$$p \wedge \neg q$$

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

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

Exemplos de formas normais conjuntivas:

$$p$$

$$\neg p$$

$$p \wedge \neg q$$

$$\neg p \vee q$$

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

Como já observámos, para qualquer variável proposicional p temos $\mathsf{V} \equiv p \vee \neg p$ e $\mathsf{F} \equiv p \wedge \neg p$. Ambas as formas são FND e FNC. Qualquer fbf tem uma FND e uma FNC. De facto, em qualquer fbf podemos usar equivalências básicas para obter uma forma normal conjuntiva:

1. "Removem-se" todas as $\rightarrow$.
2. Se a expressão contém negações de conjunções ou negações de disjunções, fazem-se desaparecer usando as leis de De Morgan, e simplifica-se onde necessário.
3. Agora basta usar as duas propriedades distributivas

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

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

   e simplificar onde necessário.

Para obter uma forma normal disjuntiva procede-se de forma análoga, usando agora em 3 as propriedades

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

$$(p \vee q) \wedge r \equiv (p \wedge r) \vee (q \wedge r).$$

Teste (2 minutos). Transforme $(p \wedge q) \vee \neg(r \rightarrow s)$ numa FND e numa FNC.

Exemplo. $(p \to q \vee r) \to (p \wedge s)$

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

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

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

$\equiv ((p \wedge \neg q \wedge \neg r) \vee p) \wedge ((p \wedge \neg q \wedge \neg r) \vee s)$ ($\vee$ é distributiva relativamente a $\wedge$)

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

$\equiv p \wedge (p \vee s) \wedge (\neg q \vee s) \wedge (\neg r \vee s)$ ($\vee$ é distributiva relativamente a $\wedge$) (FNC)

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

Uma *função de verdade* (ou *função lógica*) é uma função que só pode tomar os valores lógicos V ou F e cujos argumentos também só podem tomar esses valores. Por exemplo,

$$f(p,q) = \begin{cases} \mathsf{V} \text{ se } p \text{ é } \mathsf{V} \\ \mathsf{V} \text{ se } p \text{ e } q \text{ são ambas } \mathsf{F} \\ \mathsf{F} \text{ se } p \text{ é } \mathsf{F} \text{ e } q \text{ é } \mathsf{V} \end{cases}$$

é uma função de verdade.

É claro que toda a função de verdade pode ser representada por uma tabela de verdade. No exemplo anterior:

p	q	$f(p,q)$
V	V	V
V	F	V
F	V	F
F	F	V

De seguida vamos ver que

Toda a função de verdade é equivalente a uma fbf.

A metodologia a seguir será encontrar uma fbf com a mesma tabela de verdade (podemos construir quer uma FND quer uma FNC).

Técnica. Para construir uma FND, tendo em conta que a disjunção de um número finito de fbf's é V se e só se uma delas o for, basta tomar cada linha da tabela que tenha valor V e construir uma conjunção fundamental que só seja verdadeira nessa linha. De modo análogo, para construir uma FNC, basta considerar cada linha que tenha valor F e construir uma disjunção fundamental que só seja falsa nessa linha.

Exemplo. Na função f acima,

p	q	$f(p,q)$	Partes FND	Partes FNC
V	V	V	$p \wedge q$	
V	F	V	$p \wedge \neg q$	
F	V	F		$p \vee \neg q$
F	F	V	$\neg p \wedge \neg q$	

Assim, $f(p,q)$ pode ser escrita nas formas:

$$f(p,q) \equiv (p \wedge q) \vee (p \wedge \neg q) \vee (\neg p \wedge \neg q) \ \text{(FND)}$$

$$f(p,q) \equiv p \vee \neg q \ \text{(FNC)}.$$

Uma FND para uma fbf A é uma *FND plena* (*forma normal disjuntiva plena*) se cada conjunção fundamental contém o mesmo número de literais, um por cada variável proposicional de A. Uma FNC para uma fbf A é uma *FNC plena* (*forma normal conjuntiva plena*) se cada disjunção fundamental contém o mesmo número de literais, um por cada variável proposicional de A.

Exemplo. No exemplo anterior obtivemos uma FND plena e uma FNC plena.

Podemos usar a técnica das funções de verdade para determinar uma FND plena ou uma FNC plena de qualquer fbf com a excepção das tautologias (não têm uma FNC plena) e das contradições (não têm uma FND). Por exemplo:

$\mathsf{V} \equiv p \vee \neg p$, que é uma FND plena e uma FNC, mas não é uma FNC plena,

$\mathsf{F} \equiv p \wedge \neg p$, que é uma FNC plena e uma FND, mas não é uma FND plena.

Os conectivos lógicos que usámos para definir as fbf's do cálculo proposicional são $\neg$, $\wedge$, $\vee$ e $\rightarrow$. É evidente que o símbolo $\rightarrow$ não é absolutamente necessário (pela última lei da tabela das equivalências básicas): qualquer fbf pode ser substituída por outra logicamente equivalente e onde não figura o símbolo $\rightarrow$.

Um conjunto de conectivos lógicos diz-se *completo* se toda a fbf do cálculo proposicional é equivalente a uma fbf onde figuram apenas conectivos desse conjunto. É claro que

$$\{\neg, \wedge, \vee, \rightarrow\}$$

é completo, por definição.

Exemplos. Cada um dos seguintes conjuntos é um conjunto completo de conectivos:

$$\{\neg, \wedge, \vee\}, \{\neg, \wedge\}, \{\neg, \vee\}, \{\neg, \rightarrow\}, \{\mathsf{F}, \rightarrow\}.$$

Teste (2 minutos cada).

1. Mostre que $\{\neg, \rightarrow\}$ é completo.
2. Mostre que $\{if\text{-}then\text{-}else, \mathsf{V}, \mathsf{F}\}$ é completo.

Observação final. Como vimos, as tabelas de verdade são suficientes para determinar quando uma fbf é uma tautologia. Contudo, quando uma proposição tem mais do que duas variáveis e contém vários conectivos, a tabela de verdade pode começar a ficar muito complicada. Nesses casos, o método alternativo que vimos de encontrar uma prova de equivalência usando as leis de equivalência básicas ou ainda uma combinação dos dois (por exemplo, o método de Quine) pode ser mais prático. Quando usamos uma prova de equivalência, em vez de uma tabela de verdade, para verificar se duas fbf's são equivalentes, isso parece de certo modo mais parecido com o modo como comunicamos habitualmente. Embora não seja necessário raciocinar formalmente desse modo no cálculo proposicional, há outros tipos de sistemas lógicos onde isso já é necessário para averiguar da validade das fbf's pois aí as tabelas de verdade não funcionam. Para esses casos existe uma ferramenta: os sistemas de raciocínio formal. Quais são as ideais básicas destes sistemas?

Um *sistema formal* consiste em três partes:

(1) Um conjunto (numerável) de símbolos.

(2) Um conjunto de sequências finitas destes símbolos que constituem as chamadas *fórmulas bem formadas*.

(3) Um determinado conjunto de fbf's, chamadas *axiomas*, que se assumem ser verdadeiras.

(4) Um conjunto finito de "regras de dedução" chamadas *regras de inferência* que permitem deduzir uma fbf como consequência directa de um conjunto finito de fbf's.

Um sistema formal precisa de algumas regras para ajudarem a obter novas fórmulas, as chamadas *regras de inferência*. Uma regra de inferência aplica uma ou mais fbf's, chamadas *premissas*, *hipóteses* ou *antecedentes*, numa só fórmula, chamada *conclusão* ou *consequente*. Algumas regras de inferência úteis:

MP (modus ponens)	MT (modus tollens)	Conj (conjunção)
$\dfrac{p \to q,\ p}{\therefore q}$	$\dfrac{p \to q,\ \neg q}{\therefore \neg p}$	$\dfrac{p,\ q}{\therefore p \wedge q}$
Ad (adição)	**SD (silogismo disjuntivo)**	**SH (silogismo hipotético)**
$\dfrac{p}{\therefore p \vee q}$	$\dfrac{p \vee q,\ \neg p}{\therefore q}$	$\dfrac{p \to q,\ q \to r}{\therefore p \to q}$

Aqui o conceito crucial é o de dedução. Uma dedução de uma certa conclusão — digamos S — a partir de premissas $P_1, P_2, \ldots, P_n$ é feita passo a passo. Numa dedução, estabelecem-se conclusões intermédias, cada uma delas conclusão imediata das premissas e conclusões intermédias anteriores. Podemos dizer que uma dedução consiste numa sucessão de afirmações, que são premissas ou conclusões intermédias, e que termina, ao fim de um número finito de passos, quando se obtém a conclusão S.

Cada passo de dedução é correcto, i.e., não oferece dúvidas quanto à validade de cada conclusão intermédia, em consequência da validade das premissas e das conclusões intermédias anteriores.

Uma dedução de uma afirmação S a partir de premissas $P_1, P_2, \ldots, P_n$ é uma demonstração passo a passo que permite verificar que S tem que ser verdadeira em todas as circunstâncias em que as premissas sejam verdadeiras. Uma dedução formal assenta num conjunto fixo de regras de dedução e tem uma apresentação rígida — um pouco à semelhança dos programas escritos numa dada linguagem de programação.

Seja $\mathcal{C}$ um conjunto de fbf's e seja P uma fbf em S. Diz-se que P *é dedutível a partir de* C em S, e escreve-se $C \vdash_S P$ (ou apenas $C \vdash P$ se não houver dúvidas sobre o sistema S a que nos referimos) se existir uma sequência finita de fbf's, $P_1, P_2, \ldots, P_n$ tais que:

- $P_n = P$.
- Para cada $i \in \{1, \ldots, n\}$, P_i é um axioma de S ou uma fbf em C ou uma consequência dos P_i's anteriores através de aplicação das regras de inferência.

A sequência dos P_i's diz-se uma *prova formal* de P a partir de C. Se P é dedutível de um conjunto vazio escreve-se $\vdash_S P$. Neste caso, P diz-se um teorema de S.

Exemplo. $A \rightarrow (B \rightarrow C), A \rightarrow B, A \vdash C$.

Prova:	1. $A \rightarrow (B \rightarrow C)$	premissa	
	2. $A \rightarrow B$	premissa	
	3. A	premissa	
	4. B	MP(2,3)	
	5. $B \rightarrow C$	MP(1,3)	
	6. C	MP(4,5)	*QED.*

Leituras suplementares:

- James Hein, *Discrete Structures, Logic and Computability*, Secção 6.3.
- Jon Barwise e John Etchemendy, *Language, Proof and Logic*, Capítulo 6.

Apêndice: linguagens de primeira ordem

O cálculo proposicional providencia ferramentas adequadas para raciocinarmos sobre fbf's que são combinações de proposições. Mas uma proposição é uma sentença tomada como um todo o que faz com que o cálculo proposicional não sirva para todo o tipo de raciocínio que precisamos de fazer no dia a dia. Por exemplo, no argumento seguinte é impossível encontrar no cálculo proposicional um método formal para testar a correcção da dedução sem uma análise mais profunda de cada sentença:

- Todos os alunos de Engenharia Informática têm um computador portátil.
- Sócrates não tem um computador portátil.
- Então Sócrates não é um aluno de Engenharia Informática.

Para discutir e analisar este argumento precisamos de partir as sentenças em partes. As palavras "Todos", "têm" e "não" são relevantes para a compreensão do argumento. A afirmação "x tem um computador portátil" não é uma proposição porque o seu valor de verdade não é absoluto, depende de x. De um ponto de vista gramatical, a propriedade "ter um computador portátil" é um *predicado* (ou seja, a parte da frase que enuncia uma propriedade do sujeito). Do ponto de vista matemático, um predicado é uma relação (unária, binária, ternária, etc.). Por exemplo, "ter um computador portátil" é um predicado unário que podemos designar por CP; então $CP(x)$ significa que x tem um computador portátil (por exemplo, acima afirma-se que $\neg CP(Socrates)$).

Portanto, para analisar argumentos deste tipo precisamos de um cálculo de predicados, que inclua quantificadores (existencial e total). Por exemplo, se quisermos representar formalmente a afirmação

Existe um conjunto de números naturais que não contém o 4

precisamos de um quantificador existencial:

$\exists S(S$ é um subconjunto de $\mathbb{N}$ e $\neg S(4))$.

Podemos continuar com a formalização pois uma afirmação do tipo "A é um subconjunto de B" pode ser formalizada como $\forall x(A(x) \rightarrow B(x))$. Então podemos escrever a afirmação inicial na forma:

$$\exists S(\forall x(S(x) \rightarrow \mathbb{N}(x)) \wedge \neg S(4)).$$

Costuma-se classificar o cálculo proposicional como uma *lógica de ordem zero*. Porquê? Porque não permite que os conjuntos sejam quantificados e que sejam elementos de outros conjuntos. O cálculo de predicados já é uma lógica de ordem superior: permite que os conjuntos sejam quantificados e que sejam elementos de outros conjuntos. De que ordem?

Diz-se que um predicado tem *ordem* 1 se todos os seus argumentos são termos (isto é, constantes, variáveis individuais ou valores de funções). Caso contrário, diz-se que tem *ordem* $n+1$, onde n é a maior ordem entre os seus argumentos que não são termos. Por exemplo, em

$S(x) \wedge T(S)$, o predicado S tem ordem 1 e o predicado T tem ordem 2. Em $p(f(x)) \wedge q(f)$, p tem ordem 1, f tem ordem 1 e q tem ordem 2.

Uma *lógica de ordem n* é uma lógica cujas fbf's têm ordem $\leq n$.

Após termos estudado um pouco de cálculo proposicional, o passo seguinte, que não efectuaremos, seria estudar uma lógica de primeira ordem, nomeadamente o chamado *cálculo de predicados de primeira ordem*. Em vez disso aprenderemos, praticando, uma linguagem simbólica simples de primeira ordem, trabalhando com o `Tarski World` [3].

Uma linguagem de primeira ordem serve para descrever mundos da seguinte maneira:

- Cada nome deve designar um objecto.
- Nenhum nome pode designar mais do que um objecto.
- Um objecto pode ter vários nomes, mas também pode não ter nome.

As constantes são símbolos referentes a objectos previamente fixados.

Língua Portuguesa	LPO
nome	constante (designação, termo)

Os *símbolos de predicado* ou *relacionais* são símbolos que designam propriedades dos objectos ou relações entre objectos.

Por exemplo, podemos usar o símbolo EEI, um símbolo de predicado unário (ou seja, de aridade um), para designar, no universo dos alunos deste curso, `ser Estudante de Engenharia Informática`. Outro exemplo: o símbolo $<$, um símbolo de predicado binário (ou seja, de aridade dois), representa, no universo dos números reais, `ser menor do que`.

Língua Portuguesa	LPO
O Pedro é estudante de Eng. Inf.	$EEI(Pedro)$
2 é menor que 1	$2 < 1$

Portanto, numa linguagem de primeira ordem, a cada símbolo de predicado está associado exactamente um número natural — o número de argumentos que ocorre no predicado, que se designa por *aridade*. Além disso, cada símbolo de predicado ou relacional é interpretado por uma propriedade bem determinada ou uma relação com a mesma aridade que o símbolo.

Uma *sentença atómica* é uma sequência finita de símbolos, escolhidos entre as constantes, os símbolos de predicados, os parênteses "(" e ")" e a vírgula, da forma

$$P(c_1), \quad T(c_1, c_2), \quad R(c_1, c_2, c_3)$$

onde c_1, c_2, c_3 são constantes e P, T, R são símbolos de predicados num vocabulário fixado.

[3] Consulte o fascículo Usando Tarksi.

Exemplos.

Língua Portuguesa	LPO
A Rita é estudante de Comunicações e Multimédia	$CM(Rita)$
O Nuno é mais velho do que o Pedro	$MaisVelho(Nuno, Pedro)$

A notação usual é a *prefixa*: o símbolo de predicado escreve-se à esquerda. Excepções: com o símbolo de igualdade $=$ utiliza-se a notação habitual $a = b$; com os símbolos $<, >$ também se utiliza a notação habitual: $1 < 2$, $2 > 1$.

Portanto, com algumas excepções (predicados $=, <, >$), numa linguagem de primeira ordem as sentenças atómicas são expressões que se obtêm escrevendo um símbolo de predicado de aridade n, seguido de n constantes, delimitadas por parênteses e separadas por vírgulas: $P(c_1, c_2, \ldots, c_n)$. As excepções $(=, <, >)$ podem ser estendidas a outros símbolos. Note que a ordem em que as constantes ocorrem é fundamental.

Especifica-se uma linguagem de primeira ordem fixando as constantes, os símbolos de predicado e os símbolos funcionais. Cada símbolo de predicado e cada símbolo funcional tem uma aridade bem determinada. Uma linguagem de primeira ordem pode não incluir símbolos funcionais, mas necessita sempre de símbolos relacionais. No entanto, em vários exemplos, o único símbolo relacional considerado é o da igualdade $=$.

As linguagens de primeira ordem podem assim distinguir-se entre si através das respectivas constantes, símbolos de predicado e símbolos funcionais. Partilham os conectivos $\neg, \wedge, \vee, \rightarrow$ e $\leftrightarrow$ e os quantificadores $\forall, \exists$.

Quando se traduz uma frase em Língua Portuguesa para uma sentença numa linguagem de primeira ordem, tem-se em geral uma linguagem previamente definida, em que se conhecem à partida as constantes, os símbolos relacionais e (caso existam) os símbolos funcionais. No entanto, há situações em que há que decidir quais as constantes, os símbolos relacionais e (caso existam) os símbolos funcionais adequados para expressar o que se pretende.

Exemplo. Consideremos a frase O Nuno explicou o Tarski World ao Pedro.

(1) Tomando o símbolo de predicado binário $ExplicouTarskiWorld$ podemos escrever

$$ExplicouTarskiWorld(Nuno, Pedro).$$

(2) Tomando o símbolo de predicado ternário $Explicou$ podemos escrever

$$Explicou(Nuno, TarskiWorld, Pedro).$$

O poder expressivo da linguagem (2) é maior do que o da linguagem (1). De facto, considerando a frase A Rita explicou o Boole ao Miguel, esta pode ser traduzida usando o símbolo de predicado ternário $Explicou$ — teríamos $Explicou(Rita, Boole, Miguel)$ — mas não pode ser traduzida usando o símbolo de predicado $ExplicouTarskiWorld$. O símbolo de predicado $Explicou$ é mais versátil do que os símbolos de predicado $ExplicouTarskiWorld$ ou $ExplicouBoole$.

Para considerar as frases A Rita explicou o Boole ao Miguel no sábado e No domingo, o Miguel explicou o Boole ao João podemos considerar um predicado quaternário $Explicou(x, y, z, w)$ — que se lê "x explicou y a z no w" — e traduzir as duas frases consideradas para LPO:

$$Explicou(Rita, Boole, Miguel, sábado)$$

$$Explicou(Miguel, Boole, João, domingo).$$

Os *símbolos funcionais* são símbolos que permitem obter outras designações para objectos.

Exemplos. (1) Jorge é pai do Nuno. Supondo que a afirmação é verdadeira, $Jorge$ e $pai(Nuno)$ são duas designações diferentes para o mesmo indivíduo; *pai* é um *símbolo funcional unário.*

(2) As expressões 3 e $((1 + 1) + 1)$ são duas designações diferentes do mesmo número natural; $+$ é um *símbolo funcional binário.*

As expressões 1, $(1 + 1)$ e $((1 + 1) + 1)$ são termos. A definição de *termo* é a seguinte:

- Constantes são termos.
- Se F é um símbolo funcional de aridade n e $t_1, t_2, \ldots, t_n$ são n termos, então a expressão $F(t_1, t_2, \ldots, t_n)$ é um termo.

Numa linguagem de primeira ordem com símbolos funcionais

- Termos complexos obtêm-se colocando um símbolo funcional n-ário antes de um n-tuplo de n termos. Excepção: certos símbolos funcionais binários escrevem-se entre termos, como por exemplo $+$ (por exemplo, $(1 + 1)$).
- Termos usam-se como nomes ou designações na formação de sentenças atómicas.

Exemplos de linguagens de primeira ordem.
(1) `A Linguagem de Primeira Ordem da Teoria de Conjuntos`. Na linguagem de primeira ordem da Teoria de Conjuntos tem-se apenas dois símbolos de predicados, ambos binários, $=$ e $\in$. As sentenças atómicas nesta linguagem são da forma $a = b$ (lê-se "a é igual a b") e $a \in b$ (lê-se "(o elemento) a pertence ao (conjunto) b"), sendo a e b constantes individuais.

Por exemplo, supondo que a designa 2, b designa o conjunto $\mathbb{N}$ dos números naturais e c designa o conjunto dos números ímpares, tem-se:

$a \in a$	sentença falsa
$a \in b$	sentença verdadeira
$a \in c$	sentença falsa
$b = c$	sentença falsa.

(2) `A Linguagem de Primeira Ordem da Aritmética`. A linguagem de primeira ordem da Aritmética contém duas constantes 0 e 1, dois símbolos relacionais binários $=$ e $<$ e dois símbolos funcionais binários $+$ e $\times$. São termos desta linguagem $0, 1, (1+1), ((1+1)+1), (0\times(1+1)), \ldots$

Os termos na aritmética de primeira ordem formam-se segundo as regras:

- As constantes 0, 1 são termos.
- Se t_1 e t_2 são termos, também são termos as expressões $(t_1 + t_2)$ e $(t_1 \times t_2)$.
- São termos apenas as expressões que possam ser obtidas por aplicação sucessiva dos passos anteriores um número finito de vezes.

As sentenças atómicas na aritmética de primeira ordem são as expressões que se podem escrever usando os termos (no lugar das constantes) e os símbolos relacionais $=, <$:

- Se t_1 e t_2 são termos, são sentenças atómicas as expressões $t_1 = t_2$ e $t_1 < t_2$.

Com o `Tarski World` trabalharemos com linguagens de primeira ordem simples. Aí podemos construir mundos tridimensionais habitados por blocos geométricos de diversos tipos e tamanhos, e usar uma linguagem de primeira ordem para descrever esses mundos e testar o valor lógico (verdadeiro ou falso) de sentenças de primeira ordem elaboradas sobre esses mundos.