

EXERCÍCIOS DE LÓGICA PROPOSICIONAL

FRASES ATÓMICAS

Programa Tarski's World

Significado dos predicados do Mundo de Tarski

Cube(a)	– a é um cubo
Tet(a)	– a é um tetraedro (pirâmide)
Dodec(a)	– a é um dodecaedro (bola de futebol)
Small(a)	– a é pequeno
Medium(a)	– a é médio
Large(a)	– a é grande
Smaller(a, b)	– a é menor que b
Larger(a, b)	– a é maior que b
LeftOf(a, b)	– a está mais próximo da beira esquerda do que b
RightOf(a, b)	– a está mais próximo da beira direita do que b
BackOf(a, b)	– a está mais próximo da beira de trás do que b
FrontOf(a, b)	– a está mais próximo da beira da frente do que b
SameSize(a, b)	– a é do mesmo tamanho que b
SameShape(a, b)	– a é da mesma forma que b
SameRow(a, b)	– a está na mesma linha que b (horizontal)
SameCol(a, b)	– a está na mesma coluna que b (vertical)
Adjoins(a, b)	– a e b estão em quadrados adjacentes (não em diagonal)
$a = b$	– a é o mesmo que b (sinónimo)
Between(a, b, c)	– a, b, c na mesma linha, coluna ou diagonal e a está entre b e c

- 1 {1.1} **Compreender frases atómicas.** Abrir o mundo Wittgenstein's World e as frases Wittgenstein's Sentences (encontra-os no diretório TW Exercise Files) e percorrer a lista de frases. Atribuir mentalmente um valor de verdade a cada uma delas, relativamente àquele mundo. Verificar o resultado com o botão Verify. Alterar as posições dos objetos e verificar de novo as frases. O objetivo é perceber a interpretação de cada predicado da linguagem FOL usada no programa Tarski's World. Por exemplo, no mundo original nenhuma das frases com Adjoins é verdadeira; verifique porquê.
- 2 {1.2-3} **Construir um mundo.** Abra um ficheiro de frases novo e copie para lá as frases seguintes na linguagem FOL, verificando que todas são fórmulas bem formadas (*Evaluable*) e frases (*sentences*):
 1. Tet(a)
 2. Medium(a)

3. Dodec(b)
4. Cube(c)
5. FrontOf(a,b)
6. Between(a,b,c)
7. a=d
8. Larger(a,b)
9. Smaller(a,c)
10. LeftOf(b,c)

Construa um mundo novo em que as frases acima sejam simultaneamente verdadeiras. Guarde os resultados nos ficheiros **Sentences 1.2.sen** e **World 1.3.wld**.

3 {1.5} Nomear objectos. Abra **Lestrade's Sentences** e **Lestrade's world**. Atribua nomes aos objetos existentes no mundo de forma a que todas as frases na lista se tornem verdadeiras e guarde o resultado em **World 1.5.wld** com *Save As*. Obtenha uma segunda solução diferente da primeira e guarde o resultado como **World 1.6.wld**.

4 {1.7} Predicados sensíveis ao contexto. Abra **Austin's Sentences** e **Wittgenstein's World**. Avalie cada frase e escreva o resultado na primeira coluna de uma tabela. Depois rode o mundo 90° para a direita, reavalie e registe os valores numa nova coluna. Repita o processo ainda mais duas vezes.

Suponha que uma das linhas da tabela era: False False True False; o que pode dizer acerca do predicado respetivo? Acrescente um predicado que produza uma linha na tabela com os valores False True False False.

5 {1.19} Linguagem da teoria dos conjuntos. O domínio da teoria dos conjuntos, para este exercício, é constituído por objetos, conjuntos de objetos, conjuntos de conjuntos de objetos, etc.. Chamando **a** a 2, **b** a {2,4,6}, **c** a 6 e **d** a {2,7,{2,4,6}}, qual o valor de verdade das seguintes frases:

1. $a \in c$
2. $a \in d$
3. $b \in c$
4. $b \in d$
5. $c \in d$
6. $c \in b$

6 {1.8} Generalidade dos predicados. Considere as duas linguagens seguintes (ver significado dos nomes e dos predicados abaixo na tabela 1):

	L1	L2
Nomes	claire, max	claire, max, scruffy, carl
Predicados	GaveScruffy(x,y), GaveCarl(x,y)	Gave(x,y,z)

- a) Liste todas as frases atómicas que podem ser expressas em L1.
 - b) Quantas frases atómicas podem ser expressas com L2?
 - c) De quantos nomes e predicados binários necessita L1 de molde a dizer tudo o que pode dizer L2?
- 7 {1.9} Tradução para FOL.** Relativamente à tabela 1, traduza as frases seguintes para FOL. Os instantes são todos do mesmo dia.

TABELA 1

NOMES E PREDICADOS DE UMA LINGUAGEM

Português	FOL	Comentário
Nomes		
<i>Max</i>	max	Nome de uma pessoa
<i>Claire</i>	claire	Nome de uma pessoa
<i>Folly</i>	folly	Nome de um cão
<i>Carl</i>	carl	Nome de um cão
<i>Scruffy</i>	scruffy	Nome de um gato
<i>Pris</i>	pris	Nome de um gato
<i>2H00, Jan 2, 2001</i>	2:00	Nome de um instante
<i>2:01 pm, Jan 2, 1001</i>	2:01	Um minuto depois
...	...	Semelhante para outros instantes
Predicados		
<i>x é um animal doméstico</i>	Pet(x)	
<i>x é uma pessoa</i>	Person(x)	
<i>x é um estudante</i>	Student(x)	
<i>t é antes de t'</i>	$t < t'$	
<i>x estava com fome no instante t</i>	Hungry(x,t)	
<i>x estava zangado no instante t</i>	Angry(x, t)	
<i>x era o dono de y no instante t</i>	Owned(x,y,t)	
<i>x deu y a z em t</i>	Gave(x,y,z,t)	
<i>x alimentou y no instante t</i>	Fed(x,y,t)	

1. Claire era a dona de Folly às 2H00.
2. Claire deu Pris a Max às 2H05.
3. Max é um estudante.
4. Claire alimentou Carl às 2H00.
5. Folly pertencia a Max às 3H05.
6. 2H00 é mais cedo do que 2H05.

8 {1.13, 1.14} **Símbolos de função.** Considere uma extensão ao mundo de Tarski com os símbolos de função unários fm, bm, rm e lm que significam respetivamente o objeto mais à frente, mais atrás, mais à direita e mais à esquerda na coluna/linha do argumento. As fórmulas seguintes são assim frases da linguagem:

1. Tet(lm(e))
2. fm(c) = c
3. bm(b) = bm(e)
4. FrontOf(fm(e),e)
5. LeftOf(fm(b),b)
6. SameRow(rm(c),c)
7. bm(lm(c)) = lm(bm(c))
8. SameShape(lm(b),bm(rm(e)))
9. d = lm(fm(rm(bm(d))))
10. Between(b,lm(b),rm(b))

Preencha a tabela seguinte com TRUE e FALSE conforme o valor de verdade da frase no mundo respetivo.

	Leibniz	Bolzano	Boole	Witgenstein
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

Note que 3 das frases são verdadeiras nos 4 mundos. Uma delas não se consegue falsificar. Construa um mundo em que as outras duas sejam falsas.

9 {1.15} Linguagem funcional. Considere as duas linguagens seguintes:

	linguagem funcional	linguagem relacional
Nomes	claire, melanie, jon	claire, melanie, jon
símbolos funcionais	father	<i>não tem!</i>
Predicados	=, Taller	=, Taller, FatherOf

Traduza as frases seguintes da linguagem relacional para a funcional:

1. FatherOf(jon, claire)
2. FatherOf(jon, melanie)
3. Taller(claire, melanie)

Para as seguintes frases atómicas da linguagem funcional, quais têm traduções por frases atómicas na linguagem relacional? Indique a tradução e explique o problema das que não a têm.

4. father(melanie) = jon
5. father(melanie) = father(claire)
6. Taller(father(claire), father(jon)).

10 {1.20, 1.22} Linguagem da aritmética. Mostre que as expressões seguintes são termos na linguagem de primeira ordem da aritmética. A que números se referem?

1. $(0 + 0)$
2. $(0 + (1 \times 0))$
3. $((1 + 1) + ((1 + 1) \times (1 + 1)))$
4. $((((1 \times 1) \times 1) \times 1) \times 1)$

5. Mostre que há uma infinidade de termos que se referem ao número 1.

Programa Fitch

11 {Sec 2.3} Provas formais. O sistema dedutivo que utilizamos inclui as seguintes regras.

Introdução da Identidade (= Intro)

▷ | $n = n$

Eliminação da Identidade (= Elim)

▷ | $P(n)$
 ...
 $n = m$
 ...
 $P(m)$

Reiteração (Reit)

▷ | P
 ...
 P

Obtenha uma prova formal, justificando todos os passos, da frase **SameRow(b,a)**, a partir das premissas **SameRow(a,a)** e **b=a**.

12 {Sec 2.4} Abrir Ana Con 1.

- Posicione-se na 1ª conclusão, **SameShape(c,b)**. A regra Ana Con refere-se a **Cube(b)** e **Cube(c)**. Cite estas frases e faça a verificação do passo (check step).
- Posicione-se em **SameRow(b,a)**. Visto que a relação **SameRow** é simétrica e transitiva pode-se concluir **SameRow(b,a)** a partir de **SameRow(b,c)** e **SameRow(a,c)**. Cite estas duas frases e verifique o passo.
- Quais são as premissas que permitem concluir **BackOf(e,c)**? (Cite-as e verifique).
- Quais são as citações necessárias para as 4ª e 5ª regras?
- A última conclusão, **SameCol(b,b)** não precisa de citar nenhuma premissa. É simplesmente uma verdade analítica ou seja, verdade em virtude do seu significado. Especifique a regra e verifique o passo.
- Verifique a prova.

13 Construa a prova formal de que a conclusão é consequência das premissas:

a)

	SameCol(a,b)
	b=c
	c=d
—	SameCol(a,d)

b)

	Between(a,d,b)
	a=c
	e=b
—	Between(c,d,e)

c)

	Smaller(a,b)
	Smaller(b,c)
—	Smaller(a,c)

d)

	RightOf(b,c)
	LeftOf(d,e)
	b=d
—	LeftOf(c,e)