EXERCÍCIOS DE LÓGICA DE 1ª ORDEM

MÉTODOS DE PROVA COM QUANTIFICADORES

1 {12.1} **Prova informal.** Será que $\exists x\ [S(x) \wedge M(x)]$ é uma consequência das premissas seguintes:

1. $\forall x\ [(B(x) \vee T(x)) \rightarrow (M(x) \wedge G(x))]$
2. $\forall y\ [(S(y) \vee M(y)) \rightarrow T(y)]$
3. $\exists x\ S(x)$

Se sim, elabore uma prova informal. Se não, construa um mundo em que as premissas sejam verdadeiras e a conclusão falsa.

2 {12-4} **Validade de argumentos**. Assuma as premissas seguintes:

1. $\forall y\ [Cube(y) \vee Dodec(y)]$
2. $\forall x\ [Cube(x) \rightarrow Large(x)]$
3. $\exists x \neg Large(x)$

Será que se pode concluir que $\exists x\ Dodec(x)$? Se sim, mostre uma prova. Se não construa um mundo em que as premissas sejam verdadeiras e a conclusão falsa.

3 {12.5} Assuma as mesmas premissas do exercício anterior. Conclui-se que $\exists x[Dodec(x) \wedge Small(x)]$? Se sim, mostre uma prova. Se não, construa um mundo em que as premissas sejam verdadeiras e a conclusão falsa.

4 {12.7} Assuma as premissas seguintes:

1. $\forall x\ [Cube(x) \vee Dodec(x)]$
2. $\forall x\ [Cube(x) \rightarrow (Large(x) \wedge LeftOf(c, x))]$
3. $\forall x\ [\neg Small(x) \rightarrow Tet(x)]$

Será que se pode concluir $\exists z\ Dodec(z)$? Se sim, mostre uma prova. Se não construa um mundo em que as premissas sejam verdadeiras e a conclusão falsa.

5 {12.15} **Análise de prova.** Faça uma análise da tentativa de prova seguinte. As premissas são:

1. $\forall x\ \forall y\ \forall z\ [(O(x,y) \wedge O(y,z)) \rightarrow O(x,z)]$
2. $\forall x\ \forall y\ [(O(x,y) \rightarrow O(y,x)]$
3. $\exists x\ \exists y\ O(x,y)$

A conclusão pretendida é $\forall x\ O(x,x)$. A prova é: aplicando instanciação existencial à terceira premissa, sejam b e c objetos arbitrários no domínio de discurso, tais que $O(b,c)$. Pela segunda premissa, temos também $O(c,b)$. Aplicando a primeira premissa (com $x=z=b$ e $y=c$) obtemos $O(b,b)$. Mas b era arbitrário. Assim, por generalização universal, $\forall x\ O(x,x)$.

6 {13.40, 13.41, 13.42} **Quantificadores e conetivas.** Algumas das conclusões seguintes são válidas; outras não. Para as que o forem, elabore uma prova no Fitch. Para as outras, dê um contraexemplo usando o Tarski's.

a) Concluir $\exists x\ (Cube(x) \wedge Small(d))$ da premissa $\exists x\ Cube(x) \wedge Small(d)$.

b) Concluir $\forall x\ Cube(x) \vee \forall x\ Small(x)$ da premissa $\forall x\ (Cube(x) \vee Small(x))$.

c) Concluir $\forall x$ (Cube(x) $\vee$ Small(x)) da premissa $\forall x$ Cube(x) $\vee$ $\forall x$ Small(x).

7 {13.25, 13.26} **Argumentos.** Para os seguintes argumentos, elabore uma prova no Fitch se se tratar de um argumento válido ou um contraexemplo usando o Tarski's no caso contrário. (Pode usar **Taut Con** nas provas.)

1. Concluir $\exists x$ (Cube(x) $\wedge$ Small(x)) a partir de $\exists x$ Cube(x) $\wedge$ $\exists x$ Small(x).
2. Concluir $\forall x$ ((Cube(x) $\vee$ Small(x)) $\rightarrow$ Adjoins(x,b)) a partir de $\forall x$(Cube(x) $\rightarrow$ Small(x)) e $\forall x$ (Adjoins(x,b) $\rightarrow$ Small(x)).

8 {13.29, 13.30} **Argumentos.** Para cada um dos seguintes argumentos, elabore uma prova formal no Fitch. (Pode usar **Taut Con** nas provas, mas não **FO Con**.)

1. Concluir $\exists x$ Cube(x) a partir de $\forall x$(Small(x) $\rightarrow$ Cube(x)) e de $\exists x$ $\neg$Cube(x) $\rightarrow$ $\exists x$ Small(x).
2. Concluir $\exists x$ Likes(x, carl) a partir de Likes(carl, marx) e de $\forall x$ ($\exists y$ (Likes(y,x) $\vee$ Likes(x,y)) $\rightarrow$Likes(x,x)).

9 {12.22, 13.51} **Prova informal e formal**. Obtenha uma prova informal e uma prova formal de $\exists x\ (P(x) \rightarrow \forall y\ P(y))$, sem premissas.

10 Analise o seguinte argumento:

| 1. $\forall x$ (Cube(x) $\rightarrow$ Small(x))

| 2. $\forall x$ ($\neg$Adjoins(x, b) $\rightarrow$ $\neg$Small(x))

____

| 3. $\forall x$ ((Cube(x) $\vee$ Small(x)) $\rightarrow$ Adjoins(x, b))

Se o argumento for válido, efetue a respetiva prova formal num ficheiro Fitch. Se não for, crie um contraexemplo num ficheiro do programa Tarski.