UNIVERSIDADE DE ÉVORA

Departamento de Informática

Lógica e Computação

Lógica de Primeira Ordem — Exercícios

Estes exercícios foram copiados e/ou inspirados nas seguintes obras:

- **Lógica e Aritmética** de *Augusto Franco de Oliveira*.
- `forall`χ de *P. D. Magnus*.
- **Logic in Computer Science** de *Michael Huth* e *Mark Ryan*.
- **Artifical Intelligence, A Modern Approach** de *Stuart Russell* e *Peter Norvig*.
- **Mathematics for Computer Science: Course Textbook** de *Eric Lehman*, *F Thomson Leighton* e *Albert R Meyer*.

1 Formalização de Linguagem Natural

Exercício 1 No domínos dos animais, formalize:

1. O Aníbal, o Bói e a Certa vivem no Jardim Zoológico.
2. Bói é um réptil, mas não é um crocodilo.
3. Se Certa gosta do Bói então Bói é um macaco.
4. Se Bói e Certa são crocodilos, Aníbal gosta de ambos.
5. Alguns répteis vivem no jardim zoológico.
6. Todo o crocodilo é um réptil.
7. Qualquer animal que viva num jardim zoológico é um macaco ou um crocodilo.
8. Alguns répteis não são crocodilos.
9. Certa gosta de um réptil.
10. Bói gosta de todos os macacos que moram no jardim zoológico.
11. Todos os animais de que Aníbal gosta também gostam dele.
12. Se algum animal for um réptil, é o Aníbal.
13. Se algum animal for um crocodilo, também é um réptil.
14. Qualquer animal de que Certa gosta, também Aníbal gosta.
15. Há um animal que gosta do Bói mas, infelizmente, o Bói não corresponde.

Exercício 2

$S(x)$	x sabe a combinação do cofre.
$E(x)$	x é espião.
$V(x)$	x é vegetariano.
$C(x, y)$	x confia em y.
B	Bond, James.
G	Nell, Ma.

Use os símbolos acima para formalizar:

1. Bond é um espião, mas nenhum vegetariano é espião.
2. Ninguém sabe a combinação do cofre, a não ser que Nell a saiba.
3. Nenhum espião sabe a combinação do cofre.
4. Nem Bond nem Nell são vegetarianos.
5. Nell confia num vegetariano.
6. Quem confia em Bond confia num vegetariano.
7. Quem confia em Bond confia em alguém que confia num vegetariano.
8. Só Nell sabe a combinação do cofre.
9. Nell confia em Bond, mas em mais ninguém.
10. A pessoa que sabe a combinação do cofre é vegetariano.
11. A pessoa que sabe a combinação do cofre não é espião.

Exercício 3 Defina linguagens adequadas, formalize e demonstre.

1. Os únicos candidatos são o João e o Alberto. O João e o Alberto são idiotas. Portanto, qualquer candidato é idiota.
2. Todo o barbeiro faz a barba das pessoas que não se barbeiam a si próprias. Nenhum barbeiro faz a barba das pessoas que se barbeiam a si próprias. Portanto, não existem barbeiros.

Exercício 4 Defina uma linguagem adequada e formalize:

1. Ou o Sr. Aas ou o Sr. Hoek foi assassinado.
2. Se o Sr. Aas foi assassinado, o culpado é o cozinheiro.
3. Se o Sr. Hoek foi assassinado, o culpado não é o cozinheiro.
4. Ou o culpado é o mordomo ou a duquesa mente.
5. O culpado é o cozinheiro só se a duquesa mente.
6. Se a arma do crime for uma frigideira, então o crime deve ter sido cometido pelo cozinheiro.
7. Se a arma do crime não for uma frigideira então o culpado é o cozinheiro ou o mordomo.
8. O Sr. Aas foi assassinado se e só se o Sr. Hoek não foi assassinado.
9. A duquesa está a mentir, a não ser que o o Sr. Hoek tenha sido assassinado.
10. Se o Sr. Aas foi assassinado, a arma do crime foi uma frigideira.
11. Como o culpado é o cozinheiro, o mordomo está inocente.
12. Claro que a duquesa está a mentir!

2 Fórmulas Proposicionais

Exercício 5 Tendo em conta as convenções de escrita, desenhe árvores de análise sintática das seguintes fórmulas.

1. $p(a)$
2. $p(a) \to q(a)$
3. $\forall x\ p(x) \to q(x)$
4. $p(a) \land \big(\forall x\ p(x) \to q(x) \to q(a)\big)$
5. $\big(p(a) \land \forall x\ p(x) \to q(x)\big) \to q(a)$
6. $\big(p(a) \land \forall x\ p(x)\big) \to \big(q(x) \to q(a)\big)$

3 Dedução Natural

Exercício 6 Faça as seguintes provas, usando acumulativamente as regras e condições indicadas:

Sejam p, q, r símbolos relacionais, f funcional e a, b termos «*adequados*».

Eliminação Universal: $\forall^-$.

1. $p(a)\,, \forall x\ p(x) \to q(x) \vdash q(a)$
2. $\neg q(a)\,, \forall x\ p(x) \to q(x) \vdash \neg p(a)$

Introdução Universal: $\forall^+$.

3. $\forall x\ p(x) \to q(x)\,, \forall x\ q(x) \to r(x)$
   $\vdash \forall x\ p(x) \to r(x)$
4. $\forall x\ p(x)\,, \forall x\ p(x) \to q(x) \vdash \forall x\ q(x)$
5. $\forall x\ p(x) \land q(x) \vdash \forall x\ p(x)$
6. $\forall x\ p(x) \land q(x) \dashv\vdash \forall x\ p(x) \land \forall x\ q(x)$
7. $\forall x\ p(x) \lor q(x)\,, \forall x\ \neg p(x) \vdash \forall x\ q(x)$

Introdução Existencial: $\exists^+$.

8. $p(a)\,, \forall x\ p(x) \to q(x) \vdash \exists x\ q(x)$

Eliminação Existencial: $\exists^-$.

N.B. $\forall xy$ abrevia $\forall x\ \forall y$ e também $\exists xy$ abrevia $\exists x\ \exists y$.

9. $\exists x\ p(x)\,, \forall x\ p(x) \to q(x) \vdash \exists x\ q(x)$
10. $\forall x\ p(x) \dashv\vdash \forall y\ p(y)$
11. $\exists x\ p(x) \dashv\vdash \exists y\ p(y)$
12. $\forall xy\ p(x, y) \dashv\vdash \forall yx\ p(x, y)$
13. $\exists xy\ p(x, y) \dashv\vdash \exists yx\ p(x, y)$
14. $\exists x\ p(x) \dashv\vdash \exists xy\ p(x) \land p(y)$
15. $\exists x \forall y\ p(x, y) \vdash \forall y \exists x\ p(x, y)$
16. $\forall x\ p(x) \dashv\vdash \neg \exists x\ \neg p(x)$
17. $\exists x\ p(x) \dashv\vdash \neg \forall x\ \neg p(x)$

Seja s uma fórmula sem variáveis livres e onde não ocorre x.

18. $\forall x\ s \to p(x) \dashv\vdash s \to \forall x\ p(x)$
19. $\forall x\ s \land p(x) \dashv\vdash s \land \forall x\ p(x)$
20. $\forall x\ s \lor p(x) \dashv\vdash s \lor \forall x\ p(x)$
21. $\exists x\ s \lor p(x) \dashv\vdash s \lor \exists x\ p(x)$
22. $\exists x\ s \land p(x) \dashv\vdash s \land \exists x\ p(x)$
23. $\exists x\ \big(p(x) \to s\big) \dashv\vdash \big(\forall x\ p(x)\big) \to s$
24. $\forall x\ \big(p(x) \to s\big) \dashv\vdash \big(\exists x\ p(x)\big) \to s$

Introdução da Igualdade: $=^+$.

25. Se em t não ocorrem variáveis, $\vdash \exists x\ x = t$
26. $\vdash \exists x\ x = x$

Eliminação da Igualdade: $=^-$.

27. $\vdash \forall x\ x = x$
28. $\vdash \forall xy\ x = y \to y = x$
29. $\vdash \forall xyz\ (x = y \land y = x) \to x = z$
30. $\vdash \forall xyuv\ (x = u \land y = v) \to \big(r(x, y) \leftrightarrow r(u, v)\big)$
31. $\vdash \forall xyuv\ (x = u \land y = v) \to \big(f(x, y) = f(u, v)\big)$
32. $p(a) \dashv\vdash \exists x\ x = a \land p(x)$
33. $\forall x\ p(x) \to (x = a \lor x = b)\,, \exists x\ p(x) \land q(x)$
    $\vdash q(a) \lor q(b)$
34. $p(a) \dashv\vdash \forall x\ x = a \to p(x)$
35. $\vdash \forall xyz\ (x = y \land y = z) \to x = y.$

4 Consequência Semântica

Exercício 7 *Senhor dos Anéis*

$\mathcal{U}$	{Legolas, Aragorn}
a^ν	{Legolas, Aragorn}
b^ν	{Legolas}
n^ν	$\emptyset$
c^ν	Aragorn

Use a interpretação acima para determinar das fórmulas seguintes quais são verdadeiras e quais são falsas.

1. $b(c)$	2. $a(c) \leftrightarrow \neg n(c)$
3. $\forall x\ a(x)$	4. $n(c) \to \big(a(c) \lor b(c)\big)$
5. $\forall x\ \neg b(x)$	6. $\exists x\ a(x) \land b(x)$
7. $\exists x\ a(x) \to n(x)$	8. $\exists x\ b(x) \to \forall x\ a(x)$

Exercício 8 *Alien*

$\mathcal{U}$	{Ripley, Dallas, Kane}
h^ν	{Ripley, Dallas, Kane}
w^ν	{Ripley, Dallas}
r^ν	$\big\{$(Ripley, Dallas) , (Dallas, Kane) , (Kane, Ripley)$\big\}$
m^ν	Kane

Use a interpretação acima para determinar das fórmulas seguintes quais são verdadeiras e quais são falsas.

1. $\exists x\ r(x, m) \land r(m, x)$
2. $\forall x\ r(x, m) \lor r(m, x)$

3. $\forall x\ h(x) \leftrightarrow w(x)$
4. $\forall x\ r(x, m) \rightarrow w(x)$
5. $\forall x\ w(x) \rightarrow \big(h(x) \wedge w(x)\big)$
6. $\exists x\ r(x, x)$
7. $\exists xy\ r(x, y)$
8. $\forall xy\ r(x, y)$
9. $\forall xy\ r(x, y) \vee r(y, x)$
10. $\forall xyz\ \big(r(x, y) \wedge r(y, z)\big) \rightarrow r(x, z)$

Exercício 9 Mostre que as seguintes relações $\models$ estão erradas, construindo interpretações onde as hipóteses são verdadeiras e a conclusão falsa:

1. $\forall x\ p(x) \rightarrow q(x) \models \exists x\ q(x)$
2. $\forall x\ p(x) \rightarrow q(x)\, , \forall x\ p(x) \rightarrow r(x) \models \exists x\ q(x) \wedge r(x)$
3. $\exists x\ p(x) \rightarrow q(x) \models \exists x\ p(x)$
4. $p(a)\, , p(b)\, , p(c) \models \forall x\ p(x)$
5. $p(a, b)\, , \exists x\ p(x, a) \models p(b, a)$
6. $\exists x\ p(x) \wedge q(x)\, , \exists x\ q(x) \rightarrow \exists x\ r(x) \models \exists x\ p(x) \wedge r(x)$
7. $\forall x\ p(x, a)\, , \forall x\ p(a, x) \models \forall x\ p(x, x)$
8. $\exists x\ p(x) \wedge q(x)\, , \exists x\ \neg p(x)\, , \exists x\ \neg q(x)$ $\models \exists x\ \neg p(x) \wedge \neg q(x)$
9. $p(a, b) \rightarrow \forall x\ p(x, b)\, , \exists x\ p(x, b) \models p(b, b)$

5 Aplicações

Exercício 10 RFC3157
As afirmações seguintes foram retiradas do *RFC3157 Internet Taskforce Document 'Securely Available Credentials – Requirements.'*
Formalize:

1. Um atacante pode convencer o servidor que ocorreu um login com sucesso, mesmo que não tenha ocorrido.
2. Um atacante pode reescrever as credenciais de qualquer pessoa no servidor.
3. Todos os utilizadores introduzem *passwords* em vez de nomes.
4. A transferência de credenciais de e para um dispositivo DEVE ser suportada.
5. NÃO DEVE ser forçado pelo protocolo que as credenciais estejam presentes em *cleartext* em qualquer dispositivo que não seja o do utilizador.
6. O protocolo DEVE suportar vários algoritmos criptográficos, incluindo algoritmos simétricos e assimétricos, de *hash* ou MAC.
7. As credenciais só DEVEM estar disponíveis para *download* depois da autenticação do utilizador e num formato que, para decifrar, necessite que o utilizador faça a autenticação completa.
8. Diferentes dispositivos de utilizadores finais PODEM ser usados para *upload*, *download* e gestão do mesmo conjunto de credenciais.

Exercício 11 Linguagem das Listas
Considere a Linguagem das Listas. Formalize, classifique (compatível, contingente, válida, contradição) justificando formalmente, dê exemplos e contra-exemplos das seguintes afirmações e afirmações:

1. $\text{First}\Big(\big[x|y\big]\Big) = x$ e $\text{Rest}\Big(\big[x|y\big]\Big) = y$.
2. $\text{Find}\Big(a, \big[x|y\big]\Big)$ se e só se $a = x$ ou $\text{Find}(a, y)$.
3. Duas listas são iguais se têm os mesmos elementos nas mesmas posições.
4. Defina a função $\text{Append}(x, y)$ de acordo com as seguintes regras:

$$\text{Append}\big([\,]\, , z\big) = z$$
$$\text{Append}\Big(\big[x|y\big]\, , z\Big) = \big[x|\text{Append}(y, z)\big]$$

5. Defina a função **comprimento** de uma lista, acrescentando a linguagem dos números naturais.
6. Mostre que duas listas iguais têm o mesmo comprimento.
7. Mostre se existem listas diferentes com o mesmo comprimento (e se no domínio existir só um elemento?).
8. Mostre se duas listas que têm os mesmos elementos também têm o mesmo comprimento.

Exercício 12 Linguagem dos Autómatos Finitos
Considere a Linguagem dos Autómatos Finitos.
Formalize:

1. **Deadlock** Um estado de onde não há saída.
2. **Monótono** Um estado de onde só há um destino.
3. **Concordantes** Dois estados que concordam nos destinos dos símbolos.
4. **Descendente** Um estado é descendente de outro se é possível transitar do segundo para o primeiro. *Sugestão: Adapte a fórmula da Regra 4 da Linguagem dos Conjuntos*.
5. **Determinista**, um autómato onde a transição está definida para todos os pares (p, a) e, se $T(p, a, q_1) \wedge T(p, a, q_2)$ então $q_1 = q_2$, *isto é,* T é **funcional** em (p, a).
6. **Não determinista** Um autómato onde a transição não tem de ser funcional nem estar definida para todos os p, a. **N.B.** que ainda falta tratar das *transições vazias* — resolva este problema.
7. **Bem Preparado**, um autómato em que nenhuma transição vai para o estado inicial, existe apenas um estado final e nenhuma transição sai desse estado final.

8. **Demonstre** que um AFD não pode ser bem-preparado e que, *vice-versa*, um autómato que é bem-preparado não pode ser determinista.
9. **Palavra**, uma lista de símbolos. Aumente a linguagem de forma a incluir a *Linguagem das Listas* onde os elementos são **símbolos**.
10. **Configuração**, um par (p, w) em que p é um estado e w uma palavra.
11. **Sucessor** de uma configuração $\left(p, \left[a|w\right]\right)$, outra configuração (q, w) em que $\mathsf{T}(p, a, q)$. *Defina sucessor como uma relação entre configurações.*
12. **Configuração Completa**, (p, w) se w é a lista [ ].
13. **Computação** da palavra w, como uma lista (!) de configurações em que:
    - A primeira configuração é (I, w).
    - O elemento seguinte é sucessor do anterior.
    - o último elemento é final ou não tem sucessor.
14. **Palavra Aceite**, uma palavra que tem uma computação em que o estado da última configuração é final.

Classifique como compatível, contingente, válida ou contradição, justificando formalmente, cada uma das seguintes afirmações:

1. *O estado inicial não é final.*
2. *Existe só um estado final.*
3. *Existe só um estado inicial.*
4. *Se existe só um estado final o autómato é bem preparado.*
5. *Se existem dois estados finais o autómato não é bem preparado.*
6. *Existe pelo menos um estado final.*
7. *Se não existem estados finais nenhuma palavra é aceite.*
8. *Se todas as palavras são aceites, todos os estados são finais.*
9. *Se todos os estados são finais todas as palavras são aceites.*

Exercício 13 Linguagem das Árvores Genealógicas

- Partindo apenas das relações Descendente, Cônjuge e Feminina, defina as restantes relações e funções (progenitor, pai, mãe, irm@, irmão, irmã, descendente, filha, filho, marido, esposa, av@, net@, ti@, prim@).

- Consulte a árvore genealógica dos deuses gregos nesta página da wikipédia e deduza quem são @s net@s de *Phoebe*, @s cunhad@s de *Hestia*, e @s bisav@s de *Anteros* num sistema lógico adequado (por exemplo, prolog).