Lógica e Computabilidade 2024-2

Hugo Nobrega

Lista de Exercícios 2

As entregas podem ser feitas em duplas, mas lembre-se que
não poderá haver repetição de duplas em listas diferentes!
Entregue todas as questões marcadas com * até
17 de outubro às 20:00

Questão 1. Classifique cada uma das fórmulas abaixo em *válida*, *contradição* ou *contingência*. Justifique cada resposta.

a. `(p ↔ [¬r]) ∧ (r ∧ p)`

```
∧
├ ↔
│ ├ p
│ └ ¬
│   └ r
└ ∧
  ├ r
  └ p
```

b. `(p ↔ [¬r]) ∧ (r ∨ p)`

```
∧
├ ↔
│ ├ p
│ └ ¬
│   └ r
└ ∨
  ├ r
  └ p
```

c. `([{r ∧ q} → {q ↔ q}] ∧ p) ∨ p`

```
∨
├ ∧
│ ├ →
│ │ ├ ∧
│ │ │ ├ r
│ │ │ └ q
│ │ │
```

```
│ │ └ ↔
│ │   ├ q
│ │   └ q
│ └ p
└ p
```

d. `(¬r) ∨ (q ↔ [q ∧ r])`

```
∨
├ ¬
│ └ r
└ ↔
  ├ q
  └ ∧
    ├ q
    └ r
```

Questão 2. A seguinte definição será usada em alguns dos itens abaixo.

> **Definição 1.** Um *literal* é um símbolo proposicional, ou a negação de um símbolo proposicional (i.e., uma fórmula $(\neg p)$ onde p é um símbolo proposicional).
>
> Em notação Backus–Naur:
>
> $$\langle \text{literal} \rangle := \langle \text{símbolo proposicional} \rangle \mid (\neg \langle \text{símbolo proposicional} \rangle)$$

a. Considere os seguintes conjuntos de fórmulas, definidos por recursão:

> **Definição 2.**
>
> - Uma *cláusula para FNC* é um literal, ou a disjunção entre um literal e uma cláusula para FNC.
> - Uma *fórmula em FNC* é uma cláusula para FNC, ou a conjunção entre uma cláusula para FNC e uma fórmula em FNC.
>
> Em notação Backus–Naur:
>
> $$\begin{aligned} \langle \text{cláusula para FNC} \rangle &:= \langle \text{literal} \rangle \mid (\langle \text{literal} \rangle \vee \langle \text{cláusula para FNC} \rangle) \\ \langle \text{fórmula em FNC} \rangle &:= \langle \text{cláusula para FNC} \rangle \\ &\quad\mid (\langle \text{cláusula para FNC} \rangle \wedge \langle \text{fórmula em FNC} \rangle) \end{aligned}$$

A forma normal conjuntiva também é conhecida como *forma normal clausal*.

Prove que toda fórmula da LC é semanticamente equivalente a alguma fórmula em FNC.

b. Faça (em pseudocódigo ou em sua linguagem favorita) um programa de computador que converta uma fórmula qualquer da LC para uma equivalente em FNC.

c. Considere os seguintes conjuntos de fórmulas, definidos por recursão:

Definição 3.

- Uma *cláusula para FND* é um literal, ou a conjunção entre um literal e uma cláusula para FND.
- Uma *fórmula em FND* é uma cláusula para FND, ou a disjunção entre uma cláusula para FND e uma fórmula em FND.

Em notação Backus–Naur:

$$\begin{aligned} \langle\text{cláusula para FND}\rangle &:= \langle\text{literal}\rangle \mid (\langle\text{literal}\rangle \wedge \langle\text{cláusula para FND}\rangle) \\ \langle\text{fórmula em FND}\rangle &:= \langle\text{cláusula para FND}\rangle \\ &\quad\mid (\langle\text{cláusula para FND}\rangle \vee \langle\text{fórmula em FND}\rangle) \end{aligned}$$

Prove que toda fórmula da LC é semanticamente equivalente a alguma fórmula em FND.

d. Faça (em pseudocódigo ou em sua linguagem favorita) um programa de computador que converta uma fórmula qualquer da LC para uma equivalente em FND.

* **e.** Para essa questão, primeiramente vamos...

- expandir o alfabeto da lógica proposicional LC, adicionando um novo símbolo: $\perp$ (usualmente chamado "falso" ou "bottom")
- expandir a definição das fórmulas, permitindo um caso base adicional: "$\perp$ é uma fórmula"
- expandir a definição de semântica das fórmulas, com um caso base adicional: "o valor-verdade de $\perp$ é sempre F."

Agora considere a seguinte definição:

Definição 4.

- Uma *fórmula em FNI* é um símbolo proposicional, ou uma fórmula do tipo $(\varphi \to \perp)$, onde φ é uma fórmula em FNI, ou uma fórmula do tipo $(\varphi \to \psi)$, onde φ, ψ são fórmulas em FNI.

Em notação Backus–Naur:

$$\begin{aligned}\langle\text{fórmula em FNI}\rangle := {} & \langle\text{símbolo proposicional}\rangle \\ & \mid (\langle\text{fórmula em FNI}\rangle \to \bot) \\ & \mid (\langle\text{fórmula em FNI}\rangle \to \langle\text{fórmula em FNI}\rangle)\end{aligned}$$

Prove que toda fórmula da LC é semanticamente equivalente a alguma fórmula em FNI.

f. Faça (em pseudocódigo ou em sua linguagem favorita) um programa de computador que converta uma fórmula qualquer da LC para uma equivalente em FNI.

g (Desafio!). Para essa questão, primeiramente vamos...

- expandir o alfabeto da lógica proposicional LC, adicionando um novo símbolo: $\bot$ (usualmente chamado "falso" ou "bottom")
- expandir a definição das fórmulas, permitindo um caso base adicional: "$\bot$ é uma fórmula"
- expandir a definição de semântica das fórmulas, com um caso base adicional: "o valor-verdade de $\bot$ é sempre F."

Agora considere a seguinte definição:

Definição 5.

- Um *literal para FNI2* é um símbolo proposicional ou uma fórmula do tipo $(p \to \bot)$, onde p é um símbolo proposicional.

Em notação Backus–Naur:

$$\begin{aligned}\langle\text{literal para FNI2}\rangle := {} & \langle\text{símbolo proposicional}\rangle \\ & \mid (\langle\text{símbolo proposicional}\rangle \to \bot)\end{aligned}$$

- Uma *cláusula para FNI2* é uma fórmula do tipo $(\ell \to \bot)$ onde ℓ é um literal para FNI2, ou uma fórmula do tipo $(\ell \to C)$ onde ℓ é um literal para FNI2 e C é uma cláusula para FNI2.

Em notação Backus–Naur:

$$\begin{aligned}\langle\text{cláusula para FNI2}\rangle := {} & (\langle\text{literal para FNI2}\rangle \to \bot) \\ & \mid (\langle\text{literal para FNI2}\rangle \to \langle\text{cláusula para FNI2}\rangle)\end{aligned}$$

- Uma *fórmula em FNI2* é uma fórmula do tipo $(C \to \bot)$ onde C é uma cláusula para FNI2, ou uma fórmula do tipo $(C \to \varphi)$ onde C é uma cláusula para FNI2 e φ é uma fórmula em FNI2.

Em notação Backus–Naur:

$$\begin{aligned}\langle\text{fórmula em FNI2}\rangle := {} & (\langle\text{cláusula para FNI2}\rangle \to \bot) \\ & \mid (\langle\text{cláusula para FNI2}\rangle \to \langle\text{fórmula em FNI2}\rangle)\end{aligned}$$

Prove que toda fórmula da LC é semanticamente equivalente a alguma fórmula em FNI2.

Questão 3. Mostre que os seguintes conjuntos de conectivos são completos:

a. $\{\neg, \wedge\}$

b. $\{\neg, \vee\}$

c. $\{\neg, \to\}$

d. $\{\neg, \leftrightarrow\}$

e. $\{\to, \bot\}$, com $\bot$ conforme descrito na Questão 2(e) acima (note que $\bot$ deve ser visto como um conectivo "nulário" ou "0-ário", i.e., um conectivo que se aplica a zero outras fórmulas, e **não** como um símbolo proposicional).

f. $\{\text{NAND}\}$, sendo NAND o conectivo binário "não ambos", i.e., com a semântica dada pela tabela abaixo:

φ	ψ	$(\varphi \text{ NAND } \psi)$
V	V	F
V	F	V
F	V	V
F	F	V

Por motivos históricos, o conectivo NAND também é conhecido como "Sheffer stroke".

g. $\{\text{NOR}\}$, sendo NOR o conectivo binário "nenhum dos dois", i.e., com a semântica dada pela tabela abaixo:

φ	ψ	$(\varphi \text{ NOR } \psi)$
V	V	F
V	F	F
F	V	F
F	F	V

h. $\{\bot, \top, \text{IF-THEN-ELSE}\}$, onde $\bot$ é o "conectivo 0-ário" sempre falso ("bottom"), $\top$ é o "conectivo 0-ário" sempre verdadeiro ("top"), e IF-THEN-ELSE é o conectivo "condicional ternário", presente em algumas linguagens de programação, cuja semântica é dada pela tabela abaixo.

φ	ψ	β	(IF φ THEN ψ ELSE β)
V	V	V	V
V	V	F	V
V	F	V	F
V	F	F	F
F	V	V	V
F	V	F	F
F	F	V	V
F	F	F	F

Em outras palavras: se φ é verdadeiro, copie ψ; senão, copie β.

Questão 4. Seja X o conjunto das fórmulas construídas apenas usando os conectivos $\wedge, \vee, \rightarrow, \leftrightarrow$ (além dos símbolos proposicional usuais).

a. Escreva a definição recursiva de X.

*** b.** Prove o seguinte teorema:

Teorema. *Seja $\varphi \in X$ e sejam $p_0, p_1, \ldots, p_n$ as subfórmulas atômicas de φ.*

Então em qualquer contexto no qual $p_0, p_1, \ldots, p_n$ são todos verdadeiros, temos que φ também é verdadeiro.

Em outras palavras, temos

$$p_0,\ p_1,\ \ldots p_{n-1},\ p_n \vDash \varphi$$

*** c.** Prove que o conjunto $C = \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ de conectivos não é completo.

Questão 5. Considere a seguinte sequência infinita de fórmulas definidas por recursão (aqui o parâmetro n é sempre natural):

$$\varphi_n = \begin{cases} (p \rightarrow q), & \text{se } n = 0 \\ (\varphi_{n-1} \rightarrow p), & \text{se } n > 0. \end{cases}$$

Para quais valores de $n \in \mathbb{N}$ temos que φ_n é uma tautologia (i.e., uma fórmula válida)? Justifique sua resposta.

Questão 6. Nesta questão você não pode usar os teoremas de corretude e completude vistos em sala.

Sejam Σ, Δ conjuntos de fórmulas da LC e sejam φ, ψ fórmulas da LC.

Prove cada um dos itens abaixo.

a. Se $\varphi \in \Sigma$, então $\Sigma \vdash \varphi$.

b. Se $\varphi : \star$ e $\varphi : \bar{\star}$ ambos estão em Σ para algum φ, então $\Sigma \vdash \psi$ para qualquer ψ.

c. Se $\Sigma \vdash \varphi$, então também $\Sigma \cup \Delta \vdash \varphi$.

* **d.** $\Sigma, \varphi \vdash \psi$ se, e somente se, $\Sigma \vdash \varphi \to \psi$.

Questão 7. Prove os itens abaixo usando árvores de avaliação.

* **a.** $p \lor q, p \to r, q \to r \vdash r$

b. $p \to (q \to r) \vdash (p \land q) \to r$

c. $p \leftrightarrow (q \leftrightarrow r) \nvdash [p \land (q \land r)] \lor [(\neg p) \land \langle(\neg q) \land (\neg r)\rangle]$

* **d.** $p \to (q \to r) \nvdash (p \to q) \to r$

Questão 8. Como vimos, além de $\{\neg, \land, \lor, \to, \leftrightarrow\}$ há outros conjuntos de conectivos completos para a LC. Em cada item abaixo, dê regras de manipulação de árvores de avaliação que correspondam aos conectivos listados e que sejam corretas, i.e., que preservem satisfabilidade das árvores.

a. $\{\text{NAND}\}$

b. $\{\text{NOR}\}$

c. $\{\bot, \top, \text{IF-THEN-ELSE}\}$

Questão 9. Nesta questão vamos provar a completude do nosso sistema de provas para a LC, i.e., vamos provar que para qualquer conjunto Σ de fórmulas da LC e qualquer fórmula φ da LC temos

$$\Sigma \vDash \varphi \quad \Rightarrow \quad \Sigma \vdash \varphi.$$

Na verdade, provaremos a contrapositiva dessa implicação.

Definição. Seja A uma árvore de avaliação e seja r um ramo de A. Dizemos que r é *saturado* se, para qualquer julgamento composto $\varphi : \star$ que ocorra como rótulo de um nó de r, temos algum dos casos abaixo:

- Caso $\neg$. Se $\varphi = (\neg\psi)$, então em r há algum nó com rótulo $\psi : \bar{\star}$.
- Caso $\land : V$. Se $\varphi = (\psi \land \beta)$ e $\star = V$, então em r há nós com rótulos $\psi : V$ e $\beta : V$.
- Caso $\land : F$. Se $\varphi = (\psi \land \beta)$ e $\star = F$, então em r há algum nó com rótulo $\psi : F$ ou algum nó com rótulo $\beta : F$.
- Caso $\lor : V$. Se $\varphi = (\psi \lor \beta)$ e $\star = V$, então em r há algum nó com rótulo $\psi : V$ ou algum nó com rótulo $\beta : V$.

- Caso $\vee : F$. Se $\varphi = (\psi \vee \beta)$ e $\star = F$, então em r há nós com rótulos $\psi : F$ e $\beta : F$.

- Caso $\to$: V. Se $\varphi = (\psi \to \beta)$ e $\star = V$, então em r há algum nó com rótulo $\psi : F$ ou algum nó com rótulo $\beta : V$.

- Caso $\to$: F. Se $\varphi = (\psi \to \beta)$ e $\star = F$, então em r há nós com rótulos $\psi : V$ e $\beta : F$.

- Caso $\leftrightarrow$: V. Se $\varphi = (\psi \leftrightarrow \beta)$ e $\star = V$, então em r há nós com rótulos $\psi : V$ e $\beta : V$, ou nós com rótulos $\psi : F$ e $\beta : F$.

- Caso $\leftrightarrow$: F. Se $\varphi = (\psi \leftrightarrow \beta)$ e $\star = F$, então em r há nós com rótulos $\psi : V$ e $\beta : F$, ou nós com rótulos $\psi : F$ e $\beta : V$.

* **a.** Seja r um ramo aberto e saturado de uma árvore de avaliação A. Seja c_r o contexto definido por:

$$c_r(p) = \begin{cases} V, & \text{se } p : V \text{ é o rótulo de algum nó de } r \\ F, & \text{se } p : F \text{ é o rótulo de algum nó de } r. \end{cases}$$

Mostre que este contexto satisfaz r (i.e., satisfaz todos os julgamentos que aparecem como rótulos dos nós em r). (*Dica*: em "quem" você pode fazer indução?)

b. Por que precisamos exigir que o ramo r seja aberto e saturado, no item anterior?

* **c.** Prove o teorema da completude: se $\Sigma \nvdash \varphi$ (i.e., nenhuma árvore de avaliação para $\Gamma = \{\sigma : V \ ; \ \sigma \in \Sigma\} \cup \{\varphi : F\}$ é fechada), então $\Sigma \nvDash \varphi$ (i.e., algum contexto torna todas as fórmulas em Σ verdadeiras mas φ falsa). Para simplificar, você pode assumir que Σ é finito.