

Cálculo de Sistemas de Informação
Perfil: MÉTODOS FORMAIS DE PROGRAMAÇÃO

1.º Ano de MEI e MMC, Universidade do Minho
Ano Lectivo de 2024/25

2º Teste — 22 de Maio de 2025, 09h00–11h00
Sala E2-2.12

*Esta prova consta de 8 questões que valem, cada uma, 2.5 valores. O tempo **médio** estimado para resolução de cada questão é de 15 min.*

Recomenda-se que os alunos leiam a prova antes de decidirem por que ordem querem responder às questões que são colocadas.

PROVA COM CONSULTA (2 horas)

Questão 1 Seja dado um sistema de ficheiros

$$FS : Path \rightarrow File \quad (F1)$$

entendido como uma relação que define o conteúdo (*File*) associado a cada *Path* existente em *FS*. Uma vez que uma *path* não pode referir-se a dois ficheiros diferentes ao mesmo tempo, *FS* é simples.

Aplicações como, por exemplo, a *Time machine* do MacOS permitem guardar, num disco externo, sucessivos *backups* de sistemas de ficheiros, em que cada *backup* é identificado pela data e hora em que o *backup* foi feito.

Suponha que alguém instala uma *Time machine* numa máquina com *FS* (F1) no seu disco. Esse *software* de imediato vai criar uma relação $Bkp : Path \times Time \rightarrow File$ no disco de *backup*, inicializando-a com

$$Bkp = FS \cdot \langle id, t_0 \rangle^\circ \quad (F2)$$

onde $t_0 \in Time$ é o instante em que a Time Machine fez esse primeiro *backup*. Quer dizer, todas as *paths* que *FS* tinha ficaram associadas ao instante t_0 .

Que garantias temos que *Bkp* seja também uma relação simples? Apresente um contra-exemplo ou cálculos relacionais que justifiquem a sua resposta.

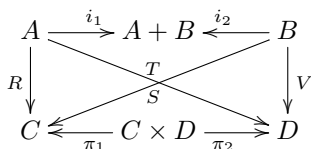
Questão 2 Prove as seguintes afirmações ou apresente contra-exemplos:

1. Toda a relação injectiva é difuncional
 2. A relação $\mathbb{N}_0 \xleftarrow{\leq} \mathbb{N}_0$ é difuncional
-

Questão 3 (1 alínea) Apresente justificações para o seguinte cálculo da generalização da *lei da troca* a relações:

$$[\langle R, S \rangle, \langle T, V \rangle] = \langle [R, T], [S, V] \rangle \quad (F3)$$

Diagrama:



Prova:

$$\begin{aligned}
& [\langle R, S \rangle, \langle T, V \rangle] = \langle [R, T], [S, V] \rangle \\
\equiv & \{ \dots \} \\
& \langle R, S \rangle = \langle [R, T], [S, V] \rangle \cdot i_1 \wedge \langle T, V \rangle = \langle [R, T], [S, V] \rangle \cdot i_2 \\
\equiv & \{ \dots \} \\
& \langle R, S \rangle = \langle [R, T] \cdot i_1, [S, V] \cdot i_1 \rangle \wedge \langle T, V \rangle = \langle [R, T] \cdot i_2, [S, V] \cdot i_2 \rangle \\
\equiv & \{ \dots \} \\
& \langle R, S \rangle = \langle R, S \rangle \wedge \langle T, V \rangle = \langle T, V \rangle \\
\equiv & \{ \dots \} \\
& \text{TRUE}
\end{aligned}$$

Questão 4 Considere o seguinte operador relacional

$$R \ominus S = R \cap \perp / S^\circ \quad (\text{F4})$$

que restringe a relação R às suas entradas onde S não está definida:

$$b (R \ominus S) a \Leftrightarrow \begin{cases} b R a \\ \langle \forall x :: \neg (x S a) \rangle \end{cases}$$

Use igualdade indirecta para demonstrar a seguinte propriedade desse operador:

$$R \ominus (S \cup Q) = (R \ominus S) \cap (R \ominus Q) \quad (\text{F5})$$

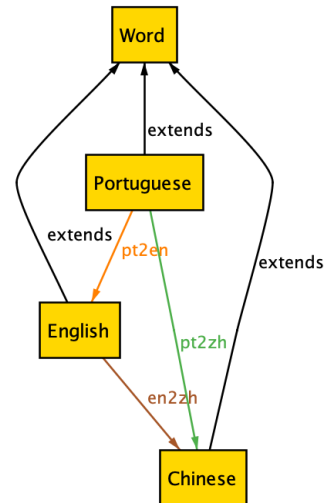
Questão 5 O modelo Alloy dado em anexo e representado ao lado foi a resposta do chatGPT ao seguinte pedido:

Could you please write an Alloy model for the following requirement:

A dictionary is a binary relationship between words from different languages. Suppose you have: (a) a Portuguese to English dictionary; (b) a Portuguese to Chinese dictionary; (c) an English to Chinese dictionary.

We want to obtain: (d) the dictionary of Portuguese words that do not have a direct translation into Chinese but whose translation can be obtained using the English dictionaries.

Comente a qualidade do modelo que o chatGPT produziu e investigue como escrever, usando a notação relacional que estudou nesta UC, o que se pede na alínea (d).



Questão 6 Sempre que uma função $f : A \rightarrow A$ opera sobre um tipo sujeito a uma propriedade obrigatória $q : A \rightarrow \mathbb{B}$, f tem de garantir q . A pré-condição mais fraca que f tem de observar à entrada para garantir q é dada por:

$$p = q \cdot f \quad (\text{F6})$$

Seja dada a seguinte definição em Haskell,

data *Buffer* *a* = *B* { *buffer* :: [*a*], *maxsize* :: $\mathbb{Z}$ }

isto é,

$$\begin{array}{ccc} & \xrightarrow{\text{outB} = \langle \text{buffer}, \text{maxsize} \rangle} & \\ \text{Buffer } A & \cong & A^* \times \mathbb{Z} \\ & \xleftarrow{B} & \end{array} \quad (\text{F7})$$

sujeita à propriedade captada pelo seguinte quadrado “mágico”:

$$\begin{array}{ccc} A^* & \xleftarrow{\text{buffer}} & \text{Buffer} \\ \text{length} \downarrow & \subseteq & \downarrow \text{maxsize} \\ \mathbb{Z} & \xleftarrow{(\leq)} & \mathbb{Z} \end{array} \quad . \quad (\text{F8})$$

Comece por interpretar (F8) derivando a correspondente versão *pointwise*. De seguida, considere a função

queue :: *a* → *Buffer* *a* → *Buffer* *a*
queue *a* (*B* *x* *m*) = *B* (*x* ++ [*a*]) *m*

que acrescenta *a* a um *buffer*, por exemplo:

queue 2 (*B* { *buffer* = [1], *maxsize* = 3 }) = *B* { *buffer* = [1, 2], *maxsize* = 3 }

Derive a pré-condição mais fraca para (*queue* *a*) garantir (F8). **NB:** pode assumir a seguinte propriedade:

$$\text{length } (x ++ y) = \text{length } x + \text{length } y \quad (\text{F9})$$

Questão 7 Um *relator* *G* é uma construção relacional associada a um dado tipo paramétrico com o mesmo nome. Permite relacionar, estruturalmente (*G* *R*), instâncias desse tipo quando há uma relação *R* entre os respectivos objectos:

$$\begin{array}{ccc} A & \dots\dots\dots & GA \\ \downarrow R & & \downarrow GR \\ B & \dots\dots\dots & GB \end{array}$$

Mas, para ser um *relator*, *G* tem de satisfazer as seguintes propriedades:

$$G \text{ id} = \text{id} \quad (\text{F10})$$

$$G (R \cdot S) = (G R) \cdot (G S) \quad (\text{F11})$$

$$G(R^\circ) = (G R)^\circ \quad (\text{F12})$$

$$(R \subseteq S) \Rightarrow (G R) \subseteq (G S) \quad (\text{F13})$$

Mostre que

$$G \left(\frac{f}{g} \right) = \frac{G f}{G g} \quad (\text{F14})$$

se verifica.

Questão 8 A seguinte função paramétrica, em Haskell,

$\text{grd} :: (a \rightarrow \text{Bool}) \rightarrow a \rightarrow \text{Either } a \ a$

tem tipo relacional

$$\text{grd} : (R \rightarrow \text{id}) \rightarrow (R \rightarrow R + R)$$

a que corresponde o teorema grátis:

$$f (R \rightarrow \text{id}) g \Rightarrow (\text{grd } f) (R \rightarrow R + R) (\text{grd } g) \quad (\text{F15})$$

- Desenhe os dois quadrados mágicos que (F15) encerra.
- Mostre que, instanciado para funções — i.e. fazendo $R := r$ — (F15) se converte no corolário:

$$(\text{grd } f) \cdot r = (r + r) \cdot (\text{grd } (f \cdot r)) \quad (\text{F16})$$

Anexo

Modelo Alloy dado pelo chatGPT para o problema da questão 5:

```
// === Core Signatures ===
abstract sig Word {}

sig Portuguese extends Word {}
sig English extends Word {}
sig Chinese extends Word {}

// === Dictionary Container ===
one sig Dicts {
  pt2en: Portuguese -> English,
  pt2zh: Portuguese -> Chinese,
  en2zh: English -> Chinese
}

// === Derived Relation ===
// Portuguese words with NO direct translation to Chinese
// but with translation via English
fun indirectPt2zh(): Portuguese -> Chinese {
  { p: Portuguese, zh: Chinese |
    // p has no direct Portuguese -> Chinese translation
    not zh in Dicts.pt2zh[p] and

    // There exists an English word e such that:
    some e: English |
      e in Dicts.pt2en[p] and
      zh in Dicts.en2zh[e]
  }
}
```