

**Especificação e Modelação**  
 Perfil: MÉTODOS FORMAIS EM ENGENHARIA DE SOFTWARE

1/4.º Ano de MEI & MMC / MiEI, Universidade do Minho  
 Ano Lectivo de 2016/17

Exame de recurso — 26 de Janeiro de 2017  
 14h00  
 Sala DI 0.03

**Ler antes de iniciar a prova** — Esta prova consta de **8 alíneas** que valem, cada uma, 2,5 valores. O tempo médio estimado para resolução de cada questão é de 15 min.

PROVA COM CONSULTA (2 horas)

**Questão 1 (2 alíneas)** No problema da cadeira pesada, estudado nas aulas desta disciplina, há uma relação de **adjacência** entre quadrículas que pode ser descrita da forma que se segue,

$$A = \text{right} \cup \text{up} \cup \text{right}^\circ \cup \text{up}^\circ$$

**where**

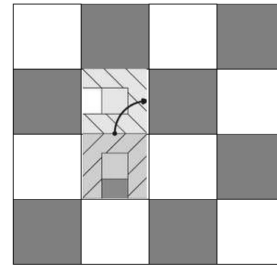
$$\text{right} = \text{id} \times (1+)$$

$$\text{up} = (1+) \times \text{id}$$

onde

$$(1+) : \mathbb{Z} \rightarrow \mathbb{Z}$$

designa a bijecção  $y = 1 + x$  nos inteiros.



1. Mostre que a relação  $A$  é simétrica mas não é transitiva. E o facto  $(y + 1, x) A (y, x - 1)$  nunca se verifica, qualquer que seja a coluna  $x$  e a linha  $y$  — porquê? Justifique.
2. Representando a cor das quadrículas por Booleanos (e.g. branco por falso e preto por verdadeiro), tem-se que a função  $\text{col} : \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{B}$  que associa cores (em  $\mathbb{B}$ ) a quadrículas é do **tipo relacional**

$$A \xrightarrow{\text{col}} \neg \tag{F1}$$

Demonstre a igualdade:

$$\text{col} = \neg \cdot \text{col} \cdot \text{up} \tag{F2}$$

**Questão 2 (2 alíneas)** Recorde o invariante do problema PROPOSITIO DE HOMINE ET CAPRA ET LVPO,

$$\begin{array}{ccc} \text{Being} & \xleftarrow{\text{CanEat}} & \text{Being} \\ \text{where} \downarrow & \subseteq & \downarrow \text{farmer} \\ \text{Bank} & \xleftarrow{\text{where}} & \text{Being} \end{array} \tag{F3}$$

onde

$$\text{CanEat} = \text{Eats} \cap \ker \text{where} \tag{F4}$$

$$\text{Eats} = \text{fox} \cdot \text{goat}^\circ \cup \text{goat} \cdot \text{beans}^\circ \tag{F5}$$

assumindo as abreviaturas (funções **constantes**)  $\text{farmer} = \underline{\text{Farmer}}$ ,  $\text{fox} = \underline{\text{Fox}}$  etc.

1. Mostre que o invariante (F3) se expande na conjunção das duas cláusulas seguintes:

$$fox \cdot goat^\circ \cap \ker where \subseteq \frac{where \cdot farmer}{where} \tag{F6}$$

$$goat \cdot beans^\circ \cap \ker where \subseteq \frac{where \cdot farmer}{where} \tag{F7}$$

2. Suponha que alguém tentava resolver este *puzzle* definindo, em lugar de (F3), o seguinte invariante:

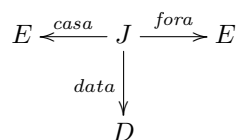
$$where \cdot goat = where \cdot farmer \tag{F8}$$

Exprima o significado de (F8) por palavras suas, e apresente justificações para os passos do seguinte raciocínio que mostra que (F8) é mais **forte** que (F7):

$$\begin{aligned} & goat \cdot beans^\circ \cap \ker where \subseteq \frac{where \cdot farmer}{where} \\ \Leftarrow & \quad \{ \dots\dots\dots \} \\ & goat \cdot beans^\circ \subseteq \frac{where \cdot farmer}{where} \\ \equiv & \quad \{ \dots\dots\dots \} \\ & where \cdot goat \subseteq where \cdot farmer \cdot beans \\ \equiv & \quad \{ \dots\dots\dots \} \\ & \tag{F8} \\ & \square \end{aligned}$$

Será (F8) também mais forte que (F6)? Justifique *informalmente* a sua resposta.

**Questão 3 (1 alínea)** As equipas ( $E$ ) de um campeonato têm jogos ( $J$ ) em casa e jogos fora, e todo o jogo ocorre numa data:



Mais ainda:

- Nenhuma equipa deverá fazer dois jogos na mesma data:

A relação  $\langle casa \cup fora, data \rangle$  é injectiva.

- Nenhuma equipa pode jogar consigo própria:

A relação  $casa \cdot fora^\circ$  é irreflexiva.

- A um jogo em casa corresponde **um e só um** outro jogo fora, com a mesma equipa:

A relação  $\frac{\langle fora, casa \rangle}{\langle casa, fora \rangle}$  é uma bijecção, cf.

$$\begin{array}{ccc} & \xleftarrow[\langle casa, fora \rangle]{\langle fora, casa \rangle} & \\ J & & J \\ & \searrow \langle casa, fora \rangle & \swarrow \langle fora, casa \rangle \\ & E \times E & \end{array} \tag{F9}$$

Recordando a lei

$$\langle R, S \rangle^\circ \cdot \langle X, Y \rangle = (R^\circ \cdot X) \cap (S^\circ \cdot Y) \quad (\text{F10})$$

codifique o invariante (F9) em Alloy.

---

**Questão 4 (2 alíneas)** Considere o seguinte modelo Alloy do problema PROPOSITIO DE HOMINE ET CAPRA ET LVPO, onde também se registam quais os seres que estão vivos em cada momento (a ideia é que se o *Farmer* deixar sozinho presa e predador então a presa morre). Esta é uma versão mais genérica do problema, onde não se discrimina quais os seres envolvidos.

```
open util / ordering [Time]
sig Time {}
sig Being {
  Eats : set Being,
  where : Bank one -> Time,
  Alive : set Time
}
abstract sig Bank {}
one sig Left, Right extends Bank {}
one sig Farmer extends Being {}
```

1. O invariante fundamental deste problema (na sua versão estática) é representado no diagrama (F3) da questão 2. Especifique em Alloy a versão dinâmica deste invariante no seguinte predicado,

```
pred inv [t : Time] {
}
```

fazendo as devidas adaptações para ter em consideração a relação *Alive*, ou seja, presa e predador não podem ficar sem o *Farmer* se estiverem vivos.

2. Complete a especificação do predicado

```
pred cross [t, t' : Time, b : Being] {
}
```

que pretende modelar a operação de atravessar o rio (se  $b = \textit{Farmer}$ , então o *Farmer* atravessa sozinho) especificando as respectivas pré-condições (o ser  $b$  deve estar vivo e deve estar no mesmo *Bank* do *Farmer*) e pós-condições (como se alteram as relações *where* e *Alive*). Tente garantir a preservação do invariante acima especificado.

---

**Questão 5 (1 alínea)** Considere o seguinte modelo SMV (incompleto) do problema anterior, onde *being* modela o ser escolhido para atravessar o rio em cada momento (se  $\textit{being} = \textit{farmer}$  então o *farmer* atravessa sozinho).

```
MODULE main

DEFINE
  farmer := 0;
  fox := 1;
  goose := 2;
  beans := 3;

VAR
```

```
where : array farmer..beans of boolean;
alive : array goose..beans of boolean;

IVAR
  being : farmer..beans;

ASSIGN
  init(where[farmer]) := FALSE;
  init(where[fox]) := FALSE;
  init(where[goose]) := FALSE;
  init(where[beans]) := FALSE;
  init(alive[goose]) := TRUE;
  init(alive[beans]) := TRUE;

LTLSPEC
  G !(where[farmer] & where[fox] & where[goose] & where[beans] & alive[goose] & alive[beans])
```

Complete o modelo por forma a que a verificação da propriedade especificada produza como contra-exemplo a solução para o problema. Tal como na alínea anterior, se o farmer deixar sozinho presa e predador então a presa deve morrer.

---