

Prova com consulta. Duração: 120 minutos.

Nome do estudante: _____ Nº _____

Assinale a resposta certa nas 10 questões seguintes. Cada resposta certa vale 0.5, cada resposta errada desconta 0.2.

1. Considere o comando 'run predicate for 3 but 2 Book'. Qual das seguinte afirmações é **FALSA**:
 - ☐ O modelo resultante nunca terá mais do que 3 instâncias em qualquer assinatura.
 - ☐ O modelo resultante nunca terá mais que 2 instâncias para a assinatura Book.
 - ☐ Só serão pesquisados modelos com exactamente 2 instâncias para a assinatura Book.
 - ☐ Serão pesquisados modelos que tentem satisfazer o predicado 'predicate'.
2. Considere a especificação ' $\text{sig } X \{ r: \text{lone } X \} \text{ fact } \{ \text{no } x: X \mid x \text{ in } x.^r \}$ '. Qual das seguintes relações satisfaz o modelo?
 - ☐ $X = \{(X0, X2), (X1, X2)\}$
 - ☐ $X = \{(X1, X2), (X2, X2)\}$
 - ☐ $X = \{(X3, X2), (X2, X1), (X1, X3)\}$
 - ☐ Todas as anteriores
 - ☐ Nenhuma das anteriores
3. Considere a seguinte relação: ' $X = \{(N0, D0), (N1, D1), (N1, D2)\}$ '. Qual das seguintes afirmações se aplica?
 - ☐ A relação é funcional e injectiva
 - ☐ A relação é funcional mas não injectiva
 - ☐ A relação não é funcional mas é injectiva
 - ☐ A relação não é nem injectiva nem funcional
4. Considere a expressão ' $\{(N0, A0)\} \cdot \{(A0, D0)\}$ '. O resultado é:
 - ☐ $\{(N0, D0)\}$
 - ☐ $\{\}$
 - ☐ $\{(D0)\}$
 - ☐ $\{(N0)\}$
 - ☐ $\{(A0)\}$
5. Considerando as seguintes relações $A = \{(3), (1)\}$ e $B = \{(2, 3), (3, 2), (3, 3)\}$, qual das seguintes expressões é verdadeira?
 - ☐ $A.B \ \& \ B.A = \text{none}$
 - ☐ $B :> A - A \rightarrow A = \{(3, 3)\}$
 - ☐ $(A <: B) \text{ in } ((B :> A) - \text{iden})$
 - ☐ Nenhuma é correta.

Prova com consulta. Duração: 120 minutos.

6. Qual das seguintes relações poderá, nalgumas situações, satisfazer $\#(r.B) < \#(A.r)$?

- ☐ r: A some \rightarrow some B
- ☐ r: A one \rightarrow some B
- ☐ Todas as anteriores
- ☐ Nenhuma

7. Dada a relação $r:A \rightarrow A$, qual das seguintes expressões afirma que r é uma relação fortemente conexa (existe um caminho entre qualquer par de elementos de A)?

- ☐ $\text{iden in } r$
- ☐ $\sim r \text{ in } r$
- ☐ $\text{all } a,b:A \mid a \text{ in } b.^{\wedge}r$
- ☐ $^{\wedge}r = A \rightarrow A$
- ☐ As duas anteriores

8. Assuma a seguinte definição em Alloy

`sig Pessoa {filhos: set Pessoa}`

Qual é que das seguintes expressões dá todos as pessoas com filhos?

- ☐ Todas as respostas são válidas.
- ☐ `filhos[Pessoa]`
- ☐ `{p: Pessoa | p.filhos = p}`
- ☐ `filhos.Pessoa`

9. Considere o seguinte modelo de um *Bag* com elementos do tipo *Elem* em Alloy,

`sig Elem{}
sig Bag{
 count: Elem \rightarrow one Int
}`

Qual das seguintes é a formalização mais correta da operação *delete* (do elemento e do Bag b , resultando noutro Bag b')?

- ☐ `pred delete [e: Elem, b, b': Bag] {
 b'.count[e] = b.count[e] - 1 }`
- ☐ `pred delete[e: Elem, b, b': Bag] {
 b'.count = b.count ++ e \rightarrow b.count[e].sub[1] }`
- ☐ `pred delete[e: Elem, b, b': Bag] {
 b.count = b'.count ++ e \rightarrow b.count[e] + 1 }`
- ☐ `pred delete[e: Elem, b, b': Bag] {
 b'.count[e] = sub[b.count[e], 1] }`

Prova com consulta. Duração: 120 minutos.

Nome do estudante: _____ Nº _____

10. Considere o seguinte código em Alloy:

```
abstract sig Person{}
sig Grad, Under extends Person{}
sig GTeacher, GStudent extends Grad{}
sig UTeacher, UStudent extends Under {}
sig Course {
    taughtby: one (GTeacher + UTeacher),
    enrolled: some (GStudent + UStudent)
}
```

Indique a melhor forma de formalizar em Alloy a seguinte restrição: os UStudents só se podem inscrever (*enroll*) em disciplinas (*Course*) ensinados por UTeachers, e os GStudents só se podem inscrever em disciplinas ensinadas por GTeachers.

- ☐ Adicionando o facto
fact enrollment{
all c :Course | c.taughtby + c.enrolled in Grad or c.taughtby + c.enrolled in Under }
- ☐ Modificando a definição de 'Course':
sig Course {
 taughtby: one (GTeacher + UTeacher),
 enrolled: some (GStudent + UStudent)
} { taughtby + enrolled in Grad or taughtby + enrolled in Under }
- ☐ Todas as respostas anteriores são formas válidas
- ☐ Nenhuma resposta é válida

11. Considere a seguinte formalização em Alloy de Pessoa e seus descendentes:

```
sig Name {}
sig Pessoa {
    n: one Name,
    descendentes : set of Pessoa
}
```

11.a) [1.5 valor] Escreva um facto que obrigue a que uma pessoa só tenha dois ascendentes.

11.b) [1.5 valor] Escreva uma operação que retorne, caso exista, a Pessoa origem, ou seja, a Pessoa de quem todas as outras pessoas descendem. Se não existir uma Pessoa nessas condições deverá retornar o conjunto vazio.

Prova com consulta. Duração: 120 minutos.

12. [1.5 valores] Através do cálculo da wp, indique a pré-condição mais fraca para o seguinte programa:

$\{P\} y:=x; \text{ if } (x>50) \text{ then } y:=0; \{y=0 \vee y=1\}$

13. [1.5 valores] Numa sequência de passos refina

$\{y<10\} S \{y>0\}$

até

$\{y<10\} \text{ if } ((x>0) \vee (y<10)) \text{ then } y:=10 \text{ endif } \{y>0\}$

Indique as regras que aplicou em cada passo.

14. [1.5 valores] Numa sequência de passos refina

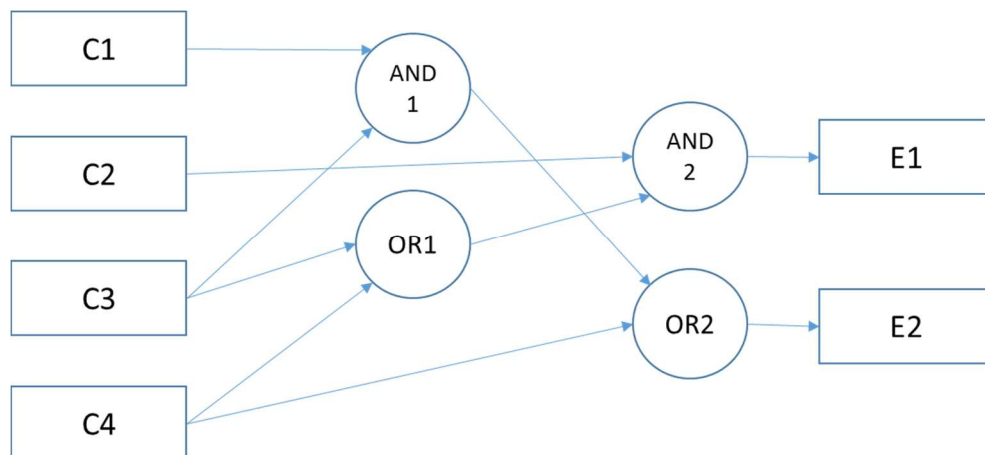
$\{i \geq 0\} S \{i = 0\}$

até

$\{i \geq 0\} \text{ while } i > 0 \text{ do } i := i-1 \{i = 0\}$

Indique as regras que aplicou em cada passo.

15. Considere a seguinte figura onde se representa as Causas (Ci) de determinados Efeitos (Ej).



No caso da figura, $(C3 \text{ OR } C4) \text{ AND } C2$ é a expressão necessária para o Efeito E1 e $(C1 \text{ AND } C3) \text{ OR } C4$ é a expressão necessária para o Efeito E2.

Prova com consulta. Duração: 120 minutos.

Considere a seguinte representação em VDM++:

```
abstract class No {  
  instance variables  
    public str : seq of char;  
} end No  
  
class Retangulo is subclass of No {  
} end Retangulo  
  
class Circulo is subclass of No {  
} end Circulo  
  
class CauseEffectGraph {  
  instance variables  
    public graph: map No to set of No; -- relaciona cada No com os seus sucessores  
  
    -- inv1: "Cada nó pode ter no máximo dois nós como antecessores".  
  
} end CauseEffectGraph
```

- a) [2 valores] Escreva uma operação (*Antecessores*) em VDM++ que devolva os Nós antecessores de um determinado Nó do Grafo (*CauseEffectGraph*). Escreva a pré-condição desta operação. Considerando a figura exemplo, *Antecessores(AND2)* = {C2, OR1}.
- b) [1.5 valor] Escreva a especificação em VDM++ do invariante 1 (inv1) para a classe *CauseEffectGraph*.
- c) [2 valores] Escreva uma operação (*Causas*) que determine todos os Nós do grafo que representam causas e outra operação (*Efeitos*) que determine todos os nós do grafo que representam efeitos. No exemplo da figura, os Nós Causa são os C1, C2, C3 e C4 e os Nós Efeito são os E1 e E2.
- d) [2 valores] Escreva uma operação que verifique se uma Causa e um Efeito passados como argumentos estão ligados entre si. Escreva a pré-condição necessária. Assuma que não há ciclos no grafo.

Boa sorte!