

Prova com consulta. Duração: 50 minutos.

Nome do estudante: \_\_\_\_\_ N° \_\_\_\_\_

**1. [8 valores]** Para cada uma das perguntas abaixo, assinale com uma cruz a resposta verdadeira. Cada resposta **correta** vale **2 valores**. Cada resposta **errada** vale **-0.5 valor**.

**a)** Considere as relações  $A = \{1 \rightarrow 2, 2 \rightarrow 3, 1 \rightarrow 4\}$  e  $B = \{(2), (4)\}$ . Qual das seguintes asserções é verdadeira?

- $\{(1)\}.A \neq B$
- $\{(1)\}.A \text{ in } B$
- $A.B = \text{univ} - B$
- Todas as anteriores
- Nenhuma das anteriores

**b)** Considere a restrição " $r \Rightarrow g \text{ else } h$ ". Qual das seguintes expressões é equivalente?

- $r + g = h$
- $(r \ \&\& \ g) \ || \ (!r \ \&\& \ h)$
- $(r \ \text{and} \ g) \ \text{or} \ (\text{not} \ r \ \text{and} \ h)$
- As duas anteriores
- Nenhuma das anteriores

**c)** Considere a restrição " $\text{all } r:A \rightarrow B \ | \ \text{some } x:\text{univ} \ | \ x.r = r.x$ ". Qual das seguintes expressões é equivalente?

- $\#A = \#B$
- $\text{univ}.r = B$
- $r.\text{univ} \text{ in } A$
- As duas anteriores
- Nenhuma das anteriores

**d)** Considere as relações  $A = \{1 \rightarrow 2, 2 \rightarrow 3, 2 \rightarrow 4\}$  e  $B = \{2 \rightarrow 1, 3 \rightarrow 2, 4 \rightarrow 2\}$ . Qual das seguintes asserções é verdadeira?

- $A.B = \text{iden}$
- $\#A = \#B = \#A.B$
- $\{(1)\} <: A = \sim(B :> \{(1)\})$
- As duas anteriores
- Nenhuma das anteriores

Prova com consulta. Duração: 50 minutos.

Nome do estudante: \_\_\_\_\_ Nº \_\_\_\_\_

1. [12 pontos] Existe uma ferramenta “Capture” que grava as ações de um utilizador sobre uma aplicação web. Estas ações são guardadas em vários traços de execução. Cada traço de execução é uma sequência de ações. Cada ação tem um tipo, poderá ter um conjunto de parâmetros e um URL caso provoque uma alteração de página. Considere a seguinte representação em Alloy de traços de execução.

```

abstract sig TYPE {
  one sig click, text, select extends TYPE {}

  sig PARAMS {}
  sig URL {}

  sig Acao {
    tipo : one TYPE,
    params : set PARAMS,
    url : lone URL
  }

  sig TracoEx {
    init : one URL,
    traco : Int one -> Acao
  }

```

Fig1: Representação de UM Traço de Execução

1.1. [2 valores] Escreva uma função que receba um *TracoEx* e retorne o conjunto de *Acao* desse traço que provocam alteração de página, isto é, que têm *URL*. No caso do exemplo da Fig.1, são as *Acao2* e *Acao3*.

```

fun updateURL [te: TracoEx]: set Acao {
  {a: te.traco[Int] | one a.url}
}

```

1.2. [2.5 valores] Escreva uma função que retorne a primeira ação (*Acao*) de um Traço de Execução. No caso da Fig. 1 é a *Acao2*.

```

fun firstAcao[te:TracoEx]: Acao {
  {a:te.traco[Int] | no b:te.traco[Int] | te.traco.b < te.traco.a}
}

```

1.3. [2.5 valores] Escreva um facto que assegure que o URL da primeira *Acao* de um traço é igual ao *URL* que inicia esse traço (o que está guardado em *init*).

Prova com consulta. Duração: 50 minutos.

```
fact { all te:TracoEx | firstAcao[te].url = te.init }
```

1.4. [2.5 valores] Escreva uma função que receba um *TracoEx* e retorne o URL da última *Acao* (desse Traço de Execução) a provocar uma alteração de página. No caso da Fig. 1 será URL1.

```
fun lastURL [te: TracoEx] : URL {  
  {u:URL | some b:updateURL[te] | u=b.url &&  
    no c:updateURL[te]-b | te.traco.c>te.traco.b  
  }  
}
```

1.5. [2.5 valores] Escreva um facto que garanta que duas ações (que provocam alteração de URL) consecutivas têm URLs diferentes. Assuma que existe a função (*URLAcao[te: TracoEx]: set Acao*) que recebe um traço de execução e retorna as *Acao* desse traço que provocam alteração de página. As *Acao2* e a *Acao3* da Fig. 1 são exemplos de duas *Acao* consecutivas que provocam alteração de página.

```
fact {  
  all te:TracoEx | all a1,a2: URLAcao[te] |  
    (a1 != a2 && no c:URLAcao[te] | te.traco.c > te.traco.a1 &&  
    te.traco.c < te.traco.a2)  
    => a1.url != a2.url  
}
```

Boa sorte!