

EXERCÍCIOS

1. Considerando as relações A e B indique o valor das expressões em Alloy:

$$A = \{(a0,d0), (b0,a0), (b1,c1),(d0,d4),(d4,b1) \}$$

- $\{(a0),(b1)\} <: A$
- $\{(a0)\}.^A$

2. Considerando as relações A, B, C e D, indique o valor das expressões em Alloy:

$$A = \{(N1),(N2),(N3)\}$$

$$C = \{(N1,N1),(N1,N2),(N3,N2)\}$$

$$D = \{(X1,N2),(X1,N1),(X3,X1)\}$$

- $C.A \rightarrow C[A]$
- $C ++ \sim D$

3. Considerando as relações A, B e C, indique o valor das expressões em Alloy:

$$A = \{(a0,d0),(a1,c0),(d4,b1),(d0,d4)\}$$

$$B = \{(b0,a0),(b1,d4)\}$$

$$C = \{(c0,b0),(c1,b1),(d4,a0)\}$$

- $C.B.A :> \{(b1)\}$
- $\{(a0)\}.^A$

4. Considerando as relações A, B e C, indique o valor das expressões em Alloy:

$$A = \{(a0,d0), (b0,a0), (b1,c1),(d0,d4),(d4,b1) \}$$

$$B = \{(b0,d4),(c1,a0)\}$$

- $\{(a0),(b1)\} <: (A.\sim B)$
- $\{(d0)\}.^A$

5. Considere a seguinte representação de um grafo em Alloy:

```
sig Point {}
sig Graph{
edge: Point -> some Point
}
```

Figura 1: Representação de um grafo em Alloy

- Especifique uma função (*Inserer*), em Alloy, que adicione uma aresta ao grafo e retorne esse grafo.
- Se o grafo de *input* for bi-conexo, a função *Inserer* (da alínea a.) garante que, após a sua execução, o grafo se mantém bi-conexo? Porquê? Em caso negativo, escreva uma pré-condição para a função *Inserer* que seja suficiente para garantir a bi-conetividade do grafo resultante.
- Escreva uma função, em Alloy, que verifique a bi-conetividade de um grafo. Escreva também a verificação (*check*) em Alloy que lhe poderia testar a bi-conetividade antes e depois de aplicar a função *Inserer* da alínea anterior.
- Escreva uma função, em Alloy, que receba um grafo e retorne o seu grafo complementar (Figura 2).

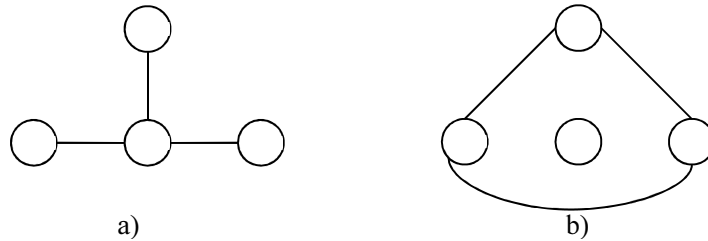


Figura 2: a) Grafo de entrada b) Grafo complementar de saída

6. Considere a seguinte representação de um polígono em Alloy (Fig. 3):

```
sig Point {}
sig Angulo {}
sig Poligono {
  vertices: set Point,
  lados: Point -> Point,
  angulo: Point -> one Angulo
}
```

Fig. 3: Representação de um grafo em Alloy

- Especifique uma função, em Alloy, que verifique se o Polígono tem os ângulos todos iguais.
- Especifique uma função, em Alloy, que verifique se um Polígono está bem construído, isto é, se dos lados e dos ângulos fazem parte apenas pontos que pertencem aos vértices desse Polígono.
- Especifique uma função, em Alloy, que retire um vértice do Polígono mantendo-o fechado.

7. Considere a seguinte representação de uma árvore em Alloy (Fig.):

```
sig Point {}
sig Edge {
  top: one Point,
  right: lone Point,
  left: lone Point
}
sig Tree{
  root: Point,
  edges: set Edge
}
```

Figura 4: Representação de uma árvore em Alloy

- Escreva uma função, em Alloy, que retorne o conjunto de folhas de uma árvore.
- Escreva uma função, em Alloy, que verifique se a raiz (*root*) da árvore é um nó (*Point*) da árvore.
- Escreva uma função, em Alloy, que verifique se uma dada árvore é conexa.

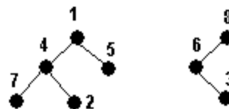


Figura 5: Exemplo de árvore desconexa

8. O Klondike, mais conhecido por "solitário", é um jogo de 52 cartas (sem os Jokers) e apenas 1 jogador. O tabuleiro está dividido em três zonas: (1) duas pilhas, (2) quatro fundações, e (3) sete filas. O objectivo do jogo consiste em preencher as 4 fundações, cada uma de acordo com a sequência de cartas do mesmo naipe desde o ás (valor = 1) até ao rei (valor = 13). A Figura 1 mostra a configuração inicial do tabuleiro.



Figura 1: Solitário

Considere a seguinte formalização do jogo Solitário em Alloy:

```

enum Vermelhas { Ouros, Copas }
enum Pretas { Paus, Espadas }
enum Estado { Cima, Baixo }

sig Carta {
  valor: Int,
  naipe: Vermelhas + Pretas,
  estado: Estado
}

abstract sig Sequencia { cartas: Int -> Carta }
abstract sig Fila, Fundacao, Pilha, Lixo extends Sequencia {}
one sig F1, F2, F3, F4, F5, F6, F7 extends Fila { }
one sig M1, M2, M3, M4 extends Fundacao { }

```

- Formalize o facto “todas as cartas de uma fundação têm o mesmo Naipe”.
- Escreva um predicado de fim de jogo (i.e., todas as fundações estão preenchidas).
- Assuma que existe uma função “fun allCardsTurnedUp[f: Fila]: set Carta” que recebe uma Fila como parâmetro e retorna as cartas dessa fila que estão viradas para cima. Formalize o predicado “Início de jogo”, em que todas as filas têm apenas uma carta virada para cima.
- Escreva a função (fun allCardsTurnedUp[f:Fila]: set Carta) que, recebendo uma Fila como parâmetro, retorne as cartas que estão viradas para cima.

9. Para cada uma das perguntas abaixo, assinale com uma cruz a resposta verdadeira. Cada resposta correcta vale 1 valor. Cada resposta errada desconta 0.5 valores.

9.a). Considere a restrição “no x : univ | $x \rightarrow x$ in r ”. Qual das seguintes expressões é equivalente?

- no iden & r
 - univ in r.univ
 - r.r in r
 - Todas as anteriores
 - Nenhuma das anteriores
-

9.b) Considere a especificação “**sig** X { r : lone X } **fact** { no x : X | x in $x.^r$ }”. Qual das seguintes relações satisfaz o modelo?

- a. $X = \{(X0, X2), (X1, X2)\}$
 - b. $X = \{(X1, X2), (X2, X2)\}$
 - c. $X = \{(X3, X2), (X2, X1), (X1, X3)\}$
 - d. Todas as anteriores
 - e. Nenhuma das anteriores
-

10. Chama-se de quadrado mágico a uma matriz de números diferentes cuja soma de todas as células de qualquer linha ou coluna resulta sempre no mesmo valor. Por exemplo, no quadrado mágico da figura ao lado, a soma de todas as células, numa linha ou coluna, é sempre 15. Neste problema decidiu-se usar o Alloy Analyzer para encontrar soluções possíveis para quadrados mágicos de dimensão 3.

4	3	8
9	5	1
2	7	6

- a. Suponha que cada célula possui duas adjacências (direita e baixo) e um valor numérico. Formalize a assinatura *Cell*.
- b. Formalize o facto: “Todas as células possuem um valor numérico diferente”.
- c. Suponha que as adjacências foram preenchidas manualmente. Formalize a asserção: “Nenhuma relação de adjacência é transitiva”. Escreva um teste que verifique essa asserção.
- d. Escreva uma função *BottomRightCell* que devolve o valor da célula mais abaixo à direita.
- e. Escreva uma função *TopLeftCell* que devolva o valor da célula mais acima à esquerda. Se não respondeu à pergunta anterior, pode considerar que *BottomRightCell* se encontra corretamente implementada.