

Université de Sherbrooke, Département d'informatique

IGL501/IGL710 : Méthodes formelles en génie logiciel, Examen intra

Professeur : Marc Frappier, 22 octobre 2021, 9 h à 12 h

Documentation permise. La correction est, entre autres, basée sur le fait que chacune de vos réponses soit *claire*, c'est-à-dire lisible et compréhensible pour le lecteur; *précise*, c'est-à-dire exacte et sans erreur; *concise*, c'est-à-dire qu'il n'y ait pas d'élément superflu; *complète*, c'est-à-dire que tous les éléments requis sont présents. Cet examen comporte 2 pages, incluant celle-ci.

Pondération :

Question	1	2	3	4	5	Total
Valeur	20	20	20	20	20	100

1. (20 pts) Traduisez les énoncés suivants avec le langage de Tarski.

0. Pour chaque carré, il existe un pentagone plus grand que lui, sur sa ligne.
1. Il existe une ligne qui ne contient que des petits pentagones.
2. Tous les pentagones sont situés sur la même ligne, et cette ligne doit contenir au moins un carré et au moins un pentagone.
3. Chaque ligne ne contient que des objets de la même taille.
4. Pour chaque objet, on retrouve sur sa ligne une paire de carré.

2. (20 pts) Complétez la spécification B ci-dessous afin de spécifier une pile.

- La taille maximale de la pile est donnée par la constante k.
- La pile doit offrir les opérations suivantes
 - Push(e) : ajoute l'élément e au sommet de la pile. Le sommet de la pile est donné par la dernière position occupée (ie, la plus grande position du domaine) dans la fonction p
 - Pop : supprime l'élément au sommet de la pile.
 - $e \leftarrow \text{Top}$: retourne l'élément au sommet de la pile
 - Sort : trie les éléments de la pile, de sorte que le plus grand élément se retrouve au sommet de la pile.
 - $e \leftarrow \text{Max}$: retourne le plus grand élément de la pile.

```
MACHINE Pile
CONSTANTS k
PROPERTIES k : NAT
VARIABLES p
INVARIANT
  p : 1..k +-> NAT
& ... /* à compléter */
INITIALISATION p := ...
OPERATIONS
  Push(e) = ...
; Pop = ...
; e <-- Top = ...
; Sort = ...
; e <-- Max = ...
END
```

3. (20 pts) Raffinez la spécification de la question 2 à partir de l'ébauche de raffinement ci-dessous; vous devez compléter l'invariant et fournir le raffinement de toutes les opérations et de l'initialisation. Vous ne pouvez pas ajouter d'autres variables dans le raffinement.

```

REFINEMENT Pile_ref
REFINES Pile
VARIABLES v,t
INVARIANT
  v : 1..k --> NAT. /* vecteur de taille fixe */
  & t : 0..k      /* donne la position du top de la pile dans v */
                  /* les éléments de la pile sont stockés entre 1 et t */
  & ... /* compléter le reste de l'invariant et du raffinement des opérations */

```

4. (20 pts) Écrivez une méthode en Dafny qui retourne la position d'un élément minimal d'un vecteur d'entiers (ie, une des positions d'un élément minimal). Voici la signature de cette opération :

```
method Minimum(a: array<int>) returns (j : int)
```

Donnez le requires, le ensures et l'implémentation de cette méthode. Comme elle devra utiliser un while, n'oubliez pas de donner l'invariant et le decreases de ce while.

5. (20 pts) Dites si les opérations suivantes préservent l'invariant de cette machine. Si l'invariant est préservé, donnez une preuve. Si il est violé, donnez un contre-exemple.

```

MACHINE Q5_invariant
VARIABLES x
INVARIANT x : 0..2
INITIALISATION x := 0
OPERATIONS
  Op1(y) = PRE y : 3..4 THEN x := x-y END
; Op2(y) = PRE y : 0..x THEN x := x-y END
; Op3   = ANY y WHERE y : 0..x THEN x := x-y END
; Op4(y) = PRE y : -2..2 THEN
  SELECT y < 0 THEN x := x + y
  WHEN y > 0 THEN x := x - y
  END
  END
; Op5   = CHOICE x := -1 OR x := 1 END
END

```

Fin de l'examen; Bonne relâche!