

Université de Sherbrooke
Département d'informatique

IGL501-IGL710 : Méthodes formelles en génie logiciel

Examen périodique

Professeur : Marc Frappier

Mardi 13 octobre 2020, 13 h 30 à 16 h 30.

Notes importantes :

- Examen en ligne: toute documentation et tout logiciel permis.
- 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.
- nombre de pages de l'examen, incluant celle-ci : 3.
- Remettez vos réponses avec l'application Web <https://turnin.dinf.usherbrooke.ca/>

Pondération :

Question	Point	Résultat
1	20	
2	20	
3	35	
4	25	
total	100	

Nom : _____ Prénom : _____

Signature : _____ CIP : _____

1. (20 pt) Traduisez les énoncés suivants avec le langage de Tarski. Utilisez comme point de départ le fichier `q1.json`; il contient chacun de ces énoncés. Remettez un fichier `q1.json`.
 - (a) Il existe un triangle et un carré de même taille, situés sur la même ligne, et ces deux objets sont plus grands que tous les autres objets.
 - (b) Chaque objet est seul sur sa ligne et sur sa colonne.
 - (c) Ou bien le pentagone `a` est grand, ou bien le triangle `b` est grand.
 - (d) Il n'y a pas de ligne avec deux pentagones.
 - (e) Une condition suffisante pour que les objets soient tous sur la même colonne est qu'il existe un petit triangle.

2. (20 pt) Pour chaque opération suivante, indiquez si elle préserve l'invariant. Si elle le préserve, justifiez votre réponse par une preuve. Si elle ne le préserve pas, donnez un contre-exemple et trouvez la précondition la plus faible (la moins restrictive) qui permet de préserver l'invariant. Remettez un fichier `q2.txt`.
 - (a) Opération : `A(x) = PRE x > y & x : NAT THEN y := y-x END`
 Invariant : `y : NAT & y > 0`
 - (b) Opération : `B(x) = PRE x : -2..-1 & x < y THEN CHOICE y:=x OR y:=x+1 END END`
 Invariant : `y : -2..0`
 - (c) Opération : `C = ANY x WHERE x : -1..1 THEN y := x END`
 Invariant : `y : -2..2`
 - (d) Opération : `D(x) =`
 `PRE x : -1..1 THEN`
 `SELECT x <= 0 THEN y := -x WHEN x >= 0 THEN y := x END`
 `END`
 Invariant : `y : -2..2`

3. (35 pt) Modélisez une file de nombres naturels en B satisfaisant les exigences suivantes. La file a une capacité maximale de k . Remettez un fichier `q3.mch`.
- `enfiler($x : \text{NAT}$)`
Ajoute l'élément x à la fin de la file.
 - `defiler`
Supprime l'élément en tête de la file.
 - `e ← tete`
Retourne l'élément qui est en tête de la file; la file demeure inchangée.
 - `sup($i : \text{NAT}$)`
Supprime le $i^{\text{ième}}$ élément de la file, en utilisant comme convention que l'élément à la tête de la file est numéroté 1 et celui à la fin de la file numéroté n , pour une file de longueur n .
 - `sup_e($e : \text{NAT}$)`
Supprime un élément e de la file, s'il en existe un. La file est inchangée s'il n'y a pas d'élément e dans la file.
 - `trier`
Trie les éléments de la file en ordre croissant, la position 1 contenant la plus petite valeur de la file.
4. (25 pt) Écrivez un programme `q4.dfy` avec Dafny qui retourne vrai ssi tous les éléments du vecteur a sont supérieurs à 0. Votre programme doit contenir toutes les déclarations permettant de prouver sa correction.

Fin de l'examen