

# MGL7160: Exercice #6

6 mars 2001

Soit la spécification de machine suivante:

```
MACHINE mach_0{ max_elems: nat }
  STATE( elems: map{nat, nat} )
  INVARIANT
    ALL( x: nat SUCH THAT x IN elems :: 2 | x ),
    size(domain(elems)) <= max_elems,
    default(elems) = 0
  INITIALLY
    domain(elems) = {}

  MESSAGE ajouter( x: nat )
    WHEN size(domain(elems)) < max_elems
      TRANSITION
        elems = bind( x, *elems[x]+1, *elems )
    OTHERWISE
      REPLY EXCEPTION maximum_atteint

  MESSAGE obtenir_min
    REPLY( r: nat )
      WHERE r = MINIMUM( x: nat SUCH THAT x IN elems :: x )

  MESSAGE present( x: nat )
    REPLY( r: boolean )
      WHERE r <=> elems[x] > 0
END
```

- [10] a) Vérifiez les diverses obligations de preuve associées à la vérification de la cohérence interne de la spécification mach\_0. En d'autres mots, répondez aux questions suivantes:
1. Est-ce que l'invariant est établi par l'état initial? Justifiez brièvement votre réponse et, si nécessaire, indiquez les modifications requises pour établir l'invariant.
  2. Est-ce que l'invariant est préservé par chacune des opérations? Justifiez brièvement et, si nécessaire, indiquez les modifications requises.
  3. Est-ce que les pré-conditions peuvent toujours être évaluées correctement? Justifiez brièvement et, si nécessaire, indiquez les modifications requises.
  4. Est-ce que les réponses retournées par les messages peuvent toujours être produites? Justifiez brièvement et, si nécessaire, indiquez les modifications requises.
- [10] b) Biais de mise en oeuvre.
1. Cette spécification possède un biais de mise en oeuvre. Expliquez ou donnez un exemple justifiant cette affirmation.
  2. Est-il possible d'ajouter un message à cette machine qui ferait en sorte que cette spécification deviendrait suffisamment abstraite (donc ne posséderait plus de biais de mise en oeuvre)? Si oui, donnez la spécification de ce message; si non, expliquez pourquoi ce n'est pas possible.