

MI030 – Analyse des programmes et sémantique (APS)
Second examen réparti
Vendredi 28 mai 2010, 15h45 – 17h45

Directives

1. Le contrôle dure 2h00.
2. Le total des points des questions est de 27, mais la note obtenue sera ramenée sur 20 (soit n la note sur 27, la note de l'examen sera $e = \text{si } n > 20 \text{ alors } 20 \text{ sinon } n$).
3. Tous les documents sont autorisés.
4. Tous les appareils électroniques sont **prohibés** (y compris les téléphones portables, les assistants numériques personnels et les agendas électroniques).

Question 1. Smalltalk propose une instruction **become** qui permet d'échanger les identifiants d'objets (oid) de deux objets de telle manière que toutes les références aux deux objets se trouvent interchangeables via les identifiants d'objets. On vous demande d'étendre BOPL pour inclure cette nouvelle instruction, don la syntaxe abstraite, est définie par :

become e_1 e_2

où les deux expressions doivent s'évaluer aux deux identifiants des objets à interchanger. Par exemple, le code suivant :

```
Point p1, p2, p3 ;  
p1 := new Point ;  
p1.x := 0 ; p1.y := 0 ;  
p3 := p1 ;  
p2.x := 10 ; p2.y := 20 ;  
become(p1, p2) ;  
writeln(p1.x) ;  
writeln(p3.y) ;
```

imprimera 10 et 20, puisque **p1** et **p3**, bien que contenant toujours l'identifiant d'objets attribué au point (0, 0) au départ, voient maintenant le point (10, 20) puisque les deux points ont été interchangés.

- a) Décrivez en cinq lignes maximum comment vous allez réaliser la sémantique dénotationnelle de cette nouvelle instruction. (3 points)
- b) Donnez la règle en sémantique dénotationnelle pour cette instruction au sein de la sémantique dénotationnelle de BOPL déjà réalisée. (3 points)
- c) Donnez le code Scheme implantation cette règle sémantique au sein de l'implantation de BOPL déjà réalisée. (3 points)

Indication : pensez à bien observer les domaines sémantiques et notez que toutes les références à des objets passent par des identifiants d'objets.

Question 2. Étendre le langage BOPL avec une instruction de syntaxe abstraite :

`forall id cexp i`

dont la sémantique informelle dit qu'on évalue d'abord l'expression de classe *cexp* obtenir l'identifiant d'une classe, par exemple *C*, puis on recherche dans la mémoire tous les objets instances de *C*, on lie ensuite successivement la variable *id* à chacun de ces objets pour exécuter l'instruction *i* dans le contexte d'un environnement et d'une mémoire étendus avec cette liaison. Par exemple, en syntaxe concrète, cela pourrait donner :

```
forall p in Point do
  p.move(10, 20)
end
```

L'exécution de cette instruction « bouge » tous les points (instances de la classe Point en mémoire) de 10 en abscisse et 20 en ordonnée.

- a) Décrivez en cinq lignes maximum comment vous allez réaliser la sémantique dénotationnelle de cette instruction. (6 points)
- b) Donnez la règle en sémantique dénotationnelle pour cette instruction au sein de la sémantique dénotationnelle de BOPL déjà réalisée. (6 points)
- c) Donnez le code Scheme implantation cette règle sémantique au sein de l'implantation de BOPL déjà réalisée. (6 points)

Indication : dans cette instruction, il y a un traitement similaire à la fois à ceux des déclarations de variables locales, de l'itération et de la façon de récupérer les objets en mémoire via les adresses et les identifiants d'objets (oid).

FIN DU CONTRÔLE.