

MI030 – Analyse des programmes et sémantique (APS)

Premier examen réparti

Lundi 26 mars 2012, 13h45 – 15h45

Directives

1. Le contrôle dure 2h00.
2. Le contrôle comporte cinq questions sur deux pages en tout.
3. Les points attribués à chaque question sont donnés à titre indicatif et totalisent 20 points sur l'ensemble du contrôle.
4. Tous les documents sont autorisés.
5. Tous les appareils électroniques sont **prohibés** (y compris les téléphones portables, les assistants numériques personnels et les agendas électroniques).

L'instruction switch dans BOPL

Nous vous demandons d'introduire dans BOPL une instruction d'alternative mutl-voie du type **switch**, mais dans une forma un peu particulière. Contrairement à l'instruction **switch** de C ou Java, l'instruction **switch** que nous vous demandons de définir autorise l'utilisation des expressions dans les cas (un peu à la manière de l'instruction **xsl:choose** de XSLT, pour ceux qui connaissent). En utilisant une syntaxe concrète de la forme suivante :

```
switch exp in
  case exp1 : inst1 ;
  ...
  case expn : instn ;
  otherwise : inst
end
```

où la clause **otherwise** est facultative. On pourrait écrire un exemple comme :

```
switch x + 1 in
  case y : x := x + 1 ;
  case y*10 : x := x - 1 ;
  otherwise : x := x * 100
end
```

Informellement, la sémantique de l'instruction consiste à évaluer l'expression *exp* introduite par le mot-clé **switch**, puis de traiter les cas dans l'ordre jusqu'à ce qu'on en trouve un qui apparie le résultat de l'expression initiale, c'est-à-dire le cas *i* tel que le résultat de *exp*_{*i*} est égal au résultat de *exp*. L'instruction *inst*_{*i*} est alors exécutée et le programme se continue après le switch. Si aucune des expressions dans les cas ne s'apparie à l'expression initiale et qu'une clause **otherwise** est

présente, alors l'instruction *inst* est exécutée et le programme se poursuit après le switch ; s'il n'y a pas de clause **otherwise** ; alors le programme continue simplement après le **switch**.

- Question 1.** Définir une syntaxe abstraite pour ces instructions intégrée à celle de BOPL. (2 points)
- Question 2.** Étendre les règles de vérification des types de BOPL pour ces instructions. (4 points)
- Question 3.** Donnez la version Prolog de vos règles de vérification des types de BOPL pour ces instructions. (4 points)
- Question 4.** Étendre les règles de sémantique opérationnelle structurelle de BOPL pour ces instructions. (5 points)
- Question 5.** Donnez la version Prolog de vos règles de sémantique opérationnelle structurelle de BOPL pour ces instructions. (5 points)

FIN DU CONTRÔLE.