TD/TME2 — Vérification des types

Jacques Malenfant, Olena Rogovchenko

1 Un premier exemple

Dérouler l'algorithme de typage pour l'exemple BOPL suivant :

```
program
  class IBox is
  vars
     Int v ;
  methods
    IBox initialize()
    begin
      self.v := 0 ;
      return self
    end
  end
let
  IBox b ;
in
begin
  b := new IBox ;
  b := b.initialize() ;
  writeln(b.v)
end
```

Examinez maintenant attentivement l'exemple 3 et vérifiez-en les types. Que constatez-vous pour cet exemple ? Expliquez ?

2 Méthodes récursives

Que se passe-t'il lorsque l'on essaie de typer :

```
program
  class C is
  methods
    Int f(Int n)
    begin
      if i = 0 then
```

```
        begin
        return 1
        end
      else
        begin
        return self.f(i-1)
        end
    end
  end
let
  C c ;
in
begin
  c := new C ;
  writeln(c.f(2))
end
```

— Comment peut-on arriver à typer ces méthodes ?
— Quelle(s) règle(s) de typage faut-il rajouter/modifier pour rendre le typage de cette méthode possible ?
— Comment se traduit cette modification au niveau du code ?

3 Méthodes mutuellement récursives

On veut être capable de typer une classe possédant deux méthodes mutuellement récursives :

```
program
  class C is
  methods
    Bool even(Int n)
    begin
      if i = 0 then
        begin
        return true
        end
      else
        begin
        return not self.odd(i-1)
        end
    end
    Bool odd(Int n)
    begin
      if i = 0 then
        begin
        return false
        end
      else
        begin
        return not self.even(i-1)
        end
    end
  end
let
```

```
  C c ;
in
begin
  c := new C ;
  writeln(c.even(2))
end
```

Pour cela, il faut connaitre les types de toutes les méthodes d'une classe avant de commencer à typer leurs corps. Il faut donc parcourir la liste des méthodes une première fois pour mettre dans le contexte de typage leurs signatures. On pourra typer les corps des méthodes lors du deuxième passage.

Mettre en oeuvre les modifications nécessaires sur le vérificateur des types pour permettre un parcours en deux temps. Tester le vérificateur des types sur l'exemple précédent.

4 Classes mutuellement récursives

S'inspirer de l'exercice précédent pour apporter au système de typage les modifications nécessaires pour pouvoir typer des définitions de classes mutuellement récursives :

```
program
  class Even is
  methods
    Bool even(Int n)
    begin
      if i = 0 then
        begin
        return true
        end
      else
        begin
        return not self.odd(i-1)
        end
    end
  end
  class Odd is
  methods
    Bool odd(Int n)
    begin
      if i = 0 then
        begin
        return false
        end
      else
        begin
        return not self.even(i-1)
        end
    end
  end
let
  Even e ;
in
begin
```

```
  e := new Even ;
  writeln(e.even(10))
end
```

5 Partie TME

1. Récupérez les sources du vérificateur de types BOPL et les exemples de programmes sur le site de l'UE et essayez d'exécuter le vérificateur pour typer les exemples de programmes.
2. Implantez dans le vérificateur les extensions des exrcices 2, 3 et 4, et testez-les sur des exemples de programmes que vous écrirez.