



UFR 919 Informatique – Master Informatique

Spécialité STL – UE ILP

TME10 — Mots clés

Christian Queinnec

1 Généralités

Objectif : Adjoindre au langage ILP4 les nouveaux mots-clefs `exists` (resp. `defined`) permettant de former des expressions testant si une variable est présente (resp. définie) ou pas.

Buts :

- Réviser les notions d'environnement dans ILP.
- Mieux maîtriser la différence entre statique et dynamique.
- S'assurer d'obtenir une bonne note à l'examen final.

2 Existence de variables

On dira qu'une variable locale est présente si le mot-clef `exists` est dans la portée de cette variable locale; on dira qu'une variable globale est présente si elle est lue ou écrite quelque part dans le programme (tester son existence avec `exists` ne lit ni n'écrit la variable concernée). Ainsi, tous les programmes suivants impriment (entre autres) OK.

```
1 if not (exists x)    let x = 1 in    let x = 1 in
2 then print "OK"      if exists x      print(x);
3 fi                  then print "OK"    if not (exists x)
4                      fi                then print "OK"
5                      fi                fi
6 x = 1;              if exists x
7 if exists x          then print "OK"
8 then print "OK"      fi
9 fi                  print(x);
```

2.1 Grammaire

Écrire une grammaire (le schéma RelaxNG compact) d'ILP4, nommée `grammar4tme10.rnc`, dans le paquet `fr.ilp.ilp4tme10`. Justifier la nécessité d'étendre la grammaire pour implanter ce nouveau trait. Écrire quelques programmes de test, par exemple en transcrivant les exemples donnés en pseudo-code.

2.2 AST

Étendre l'AST pour ajouter ce nouveau mot-clef. Vous nommerez votre classe principale `CEASTexists.java` dans le paquet `fr.upmc.ilp.ilp4tme10`, et étendrez comme nécessaire l'infrastructure d'ILP afin de prendre en charge ce nouveau type de nœud. Pensez à utiliser la carte de référence pour étendre ILP4.

2.3 Évaluation et compilation

Implanter, pour l'interprète puis pour la compilation vers C, le mot-clef `exists`. Vous pourrez commencer par les seules variables locales puis l'étendrez aux variables globales.

3 Définition de variables

On souhaite maintenant adjoindre au langage ILP4 un nouveau mot-clef nommé `defined` permettant de former des expressions testant si une variable a une valeur ou pas.

Une variable locale a toujours une valeur puisque les instructions les introduisant comportent nécessairement une expression pour l'initialiser. Une variable globale n'a de valeur qu'après avoir été initialisée c'est-à-dire avoir été affectée. Les programmes qui suivent impriment tous (entre autres) OK et n'impriment jamais KO.

```
1 if not (defined x)    let x = 1 in          let x = 1 in
2 then print "OK"       if defined x         print(x);
3 fi                   then print "OK"       if not (defined x)
4                       fi                   then print "OK"
5                                     fi
6 x = 1;               if defined x
7 if defined x          then print "KO"
8 then print "OK"       fi
9 fi                   x = 1;
10                    if defined x
11                    then print "OK"
12                    fi
```

3.1 Implantation

Suivre la même procédure afin d'implanter le nouveau mot-clef au sein d'une classe `fr.upmc.ilp.ilp4tme10.CEASTdefined.java`.