1 Génération de termes

1.1 Vocabulaire utilisé

1.2 Présentation de l'algorithme

1.3 Utilisation des termes générés dans un contexte de vérification de programme

On souhaite pouvoir décrire quel est le fonctionnement attendu du programme. On utilise pour cela les résultats des calculs de prédicats au niveau de la syntaxe. On donne un sens à une phrase logique en lui adjuvant un ou plusieurs modèle.

1.3.1 Syntaxe

Il est possible d'écrire des phrases mathématique telles que

$$\forall x \in D,\ P(x) \Rightarrow Q(f(x))$$

On utilise pour cela la syntaxe du listing 1.

Listing 1 – Définition d'une formule

```
Args = ListArgs(Term*)

Term = Var(name:String)
    | Sig(name:String, args:Args)

Formula = Predicate(name:String, args:Args)
       | And(f1:Formula, f2:Formula)
       | Or(f1:Formula, f2:Formula)
       | Imply(f1:Formula, f2:Formula)
       | Not(f:Formula)
       | Forall(var:String, domain:String, f:Formula)
       | Exists(var:String, domain:String, f:Formula)
```

La phrase mathématique précédente sera donc sous la forme du listing 2.
On peut ainsi essayer de décrire le fonctionnement d'un programme. Une fois la syntaxe écrite, il faut lui donner un sens. En effet, chaque mot qui n'est pas définit dans la grammaire doit l'être par ailleurs. Ainsi, on voudra dire par exemple que la phrase ci-dessus signifie

Pour tout entier x, x pair implique $x + 1$ impair.

1.3.2 Semantique

2 Réduction des contre-exemples

Listing 2 – Exemple d'une formule

```
Forall("x",
        "D",
        Imply(
                Predicate(
                        "P",
                        ListArgs(
                                Var(
                                        "x"
                                )
                        )
                )
                Predicate(
                        "Q",
                        ListArgs(
                                Sig(
                                        "f",
                                        ListArgs(
                                                Var(
                                                        "x"
                                                )
                                        )
                                )
                        )
                )
        )
)
```