

Algorithmes de détection de terminaison de programmes F-Safe

Équipe F-Safe

19 janvier 2012

Plan

- 1 Algorithme général
- 2 AST et graphe
- 3 Détection des cycles
- 4 Complétion partielle
- 5 Vérification de la terminaison

Algorithme général

- ➊ Construction de l'arbre de syntaxe abstraite
- ➋ Construction du graphe d'appel
- ➌ Détection des cycles
- ➍ Complétion partielle du graphe
- ➎ Vérification de la terminaison

Arbre de syntaxe abstraite

- Noeud : appels de fonctions
- Table de définitions des fonctions

Construction du graphe d'appels

- Matrice d'appel

Détection des cycles

- Entrée : Graphe d'appel
- Sortie : Liste de chemins
- Algorithme par pile

```
Proc f (g :Graphe, n : Noeud)
a = CréerArbre(n); l = CréerListeVide(); f' (g, a, l, n)
return l
end Proc
Proc f' (g :Graphe, a :Arbre, l :Liste, n :Noeud)
for chaque ancêtre anc de n dans a jusqu'à la racine do
  if anc = n then
    if l'arc (n, n->pred) est unique dans a jusqu'à la racine
    then
      AjoutCycle (l, pred, n)
    else
      return
    end if
  end if
end for
end Proc
```

Complétion partielle du graphe d'appels

- Multiplication de matrice
- Sur chaque arrête du cycle considéré

Vérification de la terminaison des cycles

```
for chaque cycle do
  for chaque boucle sur le noeud considéré do
    if la matrice d'appel ne présente pas un ordre then
      return Le programme ne termine pas
    end if
  end for
end for
return Le programme termine
```