Algorithme de complétion partielle

Algorithme

```
completion_partielle_verif( l: list list data){
        cms = list.empty
        i <-- 0
        j <-- 0

        while (i < (list.size l)) do
                directCM <-- extractCM ()
                indirectCM <-- extractCM ()
                resCM <-- produit_matriciel(directCM, indirectCM)
                j<-- j+2
                while (j <  list.size (list.nth l i)) do
                directCM <-- resCM
                indirectCM <-- extractCM ()
                resCM <-- produit_matriciel(directCM, indirectCM)
                j<-- j+1
        done
        directCM <-- resCM
        indirectCM <-- extractCM ()
        resCM <-- produit_matriciel(directCM, indirectCM)
        cms.add resCM
        i<-- i+1
        done
        for each cm in cms do
                if analyse(cm) = false then return false
                else continue
        done
        return true
}
```

Illustration

Prenons le cycle **f** $\longrightarrow$ **g** $\longrightarrow$ **h** $\longrightarrow$ **i** $\longrightarrow$ **f**

On a : **f** fait un appel à **g**, puis **g** fait un appel à **h**. Il y a donc un appel indirect de **f** à **h**. Il faut donc construire la matrice d'appel de **f** sur **h** qui n'est pas connue. La fonction ***extractCM()*** permet d'obtenir les matrices d'appels correspondantes et l'on effectue le produit matriciel en se basant sur les règles fournis dans **foetus** *page 17, table 1.*
Ensuite, **h** fait un appel à **i**, donc **f** réalise un appel indirect à **i**. Grâce à

la matrice d'appel obtenu précédemment, on calcule la matrice d'appel de **f** sur **i**. On réitère le procédé jusqu'à la fin du cycle.
Le résultat de cette étape est une liste de matrice d'appel de **f** sur elle même par tous les appels possibles. On vérifie la diagonale de chacune d'elle. S'il y a au moins un $<$ et pas de **?**, alors le programme termine. Si l'on rencontre un **?** lors de la vérification, on s'arrête et on conclut que le programme ne termine pas.

Pointeur

Le seul et unique : **foetus**, pages 15-22