

Algorithme de construction de la liste exhaustive des cycles d'un graphe d'appel

Yoann Couillec

30 janvier 2012

Résumé

On cherche à obtenir une liste de tous les cycles d'un graphe, les cycles simples et les cycles composés.

1 Algorithme

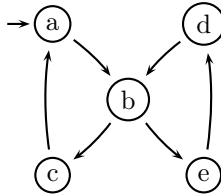
```
Proc cycles (g:Graphe, n:Noeud)
  b = CreerListeVide ()
  b = AjoutListe (b, n)
  l = CreerListeVide ()
  cycles ' (g, b, l)
  Retourner l
Fin Proc

Proc cycles ' (g:Graphe, b:Liste, l:Liste)
  Pour chaque ancetre anc de n dans b jusqu'a la racine Faire
    Si anc = n Et anc n'appartient pas a un cycle Alors
      AjoutCycle (l, anc, n)
    Fin Si
  Fin Pour
  Pour chaque enfant e de n dans g Faire
    cycles ' (g, AjoutListe (b, e), l)
  Fin Pour
Fin Proc
```

2 Exemple

2.1 Huit

Soit le graphe g suivant :



On cherche les 3 cycles : $abca$, $bedb$ et $abedbca$.

1. L'appel initial est $cycle(g, a)$
2. Qui appelle $cycle'(g, [a], [])$
3. a n'a pas d'ancêtre
4. $cycle'(g, [a, b], [])$
5. 2 enfants : c et e
6. $cycle'(g, [a, b, c], [])$
 - (a) $cycle'(g, [a, b, c, a], [abca])$, on a trouvé le cycle $abca$
 - (b) a est la racine on s'arrête
7. $cycle'(g, [a, b, e], [abca])$
 - (a) $cycle'(g, [a, b, e, d], [abca])$
 - (b) $cycle'(g, [a, b, e, d, b], [abca, bedb])$, on a trouvé le cycle $bedb$
 - (c) $cycle'(g, [a, b, e, d, b, c], [abca, bedb])$
 - (d) $cycle'(g, [a, b, e, d, b, c, a], [abca, bedb, abedbca])$, on a trouvé le cycle $abedbca$, a est la racine on s'arrête

On suppose que l est un pointeur et que les accès en concurrence sont gérés.

Références

- [1] http://fr.wikipedia.org/wiki/Algorithme_de_Tarjan