Récursivité

Récurrence et récursion

(Logique et techniques de programmation – ESI – 2002)

Plan

- Définitions et algorithmes récursifs
- Propriétés des algorithmes récursifs
- Méthode d'élaboration
- Etudes de cas
  - Tours de Hanoï
  - Tri rapide
- Exercices

Définition récursive (1)

- Ensemble de mots explicatifs qui contient directement ou indirectement le mot à définir ! Tautologie ?
- Ordre : Arrangement
- Arrangement : Action d'arranger
- Arranger : Mettre en ordre

Source : Petit Larousse illustré 2000

Définition récursive (2)

- Liste chaînée : ensemble
  - Vide
  - Composé d'un élément auquel est attaché une liste chaînée
- Factorielle : n !

  0 ! = 1

  N ! = (N-1)! x N

Définition récursive (3)

- Chaîne de caractères : ensemble
  - Vide
  - Composé d'un caractère concaténé à une chaîne
- Tableau de type T : ensemble
  - Vide
  - Composé d'un élément de type T contigu à un tableau de type T

(d'autres définitions sont possibles!)

Définition récursive (4)

- Nombre de Fibonacci

  Fib (1) = 0 Fib (2) = 1

  Fib (n) où n > 2 vaut Fib(n-2)+Fib(n-1)

- Triangle de Pascal (récurrence)

  $C_n^0 = 1$ pour $n \geq 0$

  $C_n^m = 0$ pour $m > n \geq 0$

  $C_n^m = C_{n-1}^{m-1} + C_{n-1}^m$ pour $n \geq m > 0$

Algorithme récursif

- Procédure, fonction, méthode de classe dont l'exécution entraîne un ou plusieurs appel à elle-même
- Ceci inclut donc la récursivité indirecte qui est plus générale que la récursivité directe
- Attention au cercle vicieux !!!

Propriétés des algorithmes récursifs

- Objets engendrés doivent être finis
- Clause d'évaluation immédiate (sans appel récursif)
- Quantité(s) de contrôle qui évolue(nt) d'un appel récursif à l'autre (généralement pris dans les paramètres)

  (QC → valeur triviale testée par la clause d'évaluation immédiate)

Structure type d'un algorithme récursif

SI condition

instructions contenant un appel récursif

ALORS

instructions ne contenant plus d'appel récursif

FINSI

D'autres formes évidentes sont permises !

Exemple : Factorielle

- Fonction Factorielle (↓N:entier ≥0) → entier

  SI (N = = 0)

  | Retourner 1

  SINON

  | Retourner N x Factorielle (N-1)

  FINSI

Méthode d'élaboration

- Paramétrage du problème
  – De quoi dépend le problème ?
  – Quelle est sa taille ?
- Recherche d'un cas trivial
  – Evaluable directement
  – Lorsque la taille est nulle (ou proche!)
- Découpe en sous-problèmes, plus simples à résoudre que le cas général

Etude de cas : Tours d'Hanoï

- Exposé du problème
- Application de la méthode d'élaboration
- Solution

<u>Hanoï: Exposé du problème</u>

- Les prêtres d'une secte ancienne devait déplacer une tour faite de <u>64 disques</u> de tailles décroissantes posée sur un socle, vers un autre socle en respectant certaines règles:
  - Un troisième socle est disponible
  - On ne peut déplacer qu'un disque à fois
  - Un disque peut être posé sur un socle vide ou sur un autre disque de taille supérieure

Hanoï : méthode d'élaboration

- Paramètrage :
  - Nombre de disques
  - Fonctionnalité des socles (départ, arrivée, …)
- Cas trivial
  - Pas de disque (tour vide!)
  - Un seul disque
- Quantité de contrôle : nombre de disques

Hanoï : solution

<u>Hanoi</u> ($\downarrow$ Nbd: entier, $\downarrow$ dep : car, $\downarrow$ arr: car, $\downarrow$ inter: car)

<u>SI</u> Nbd > 0

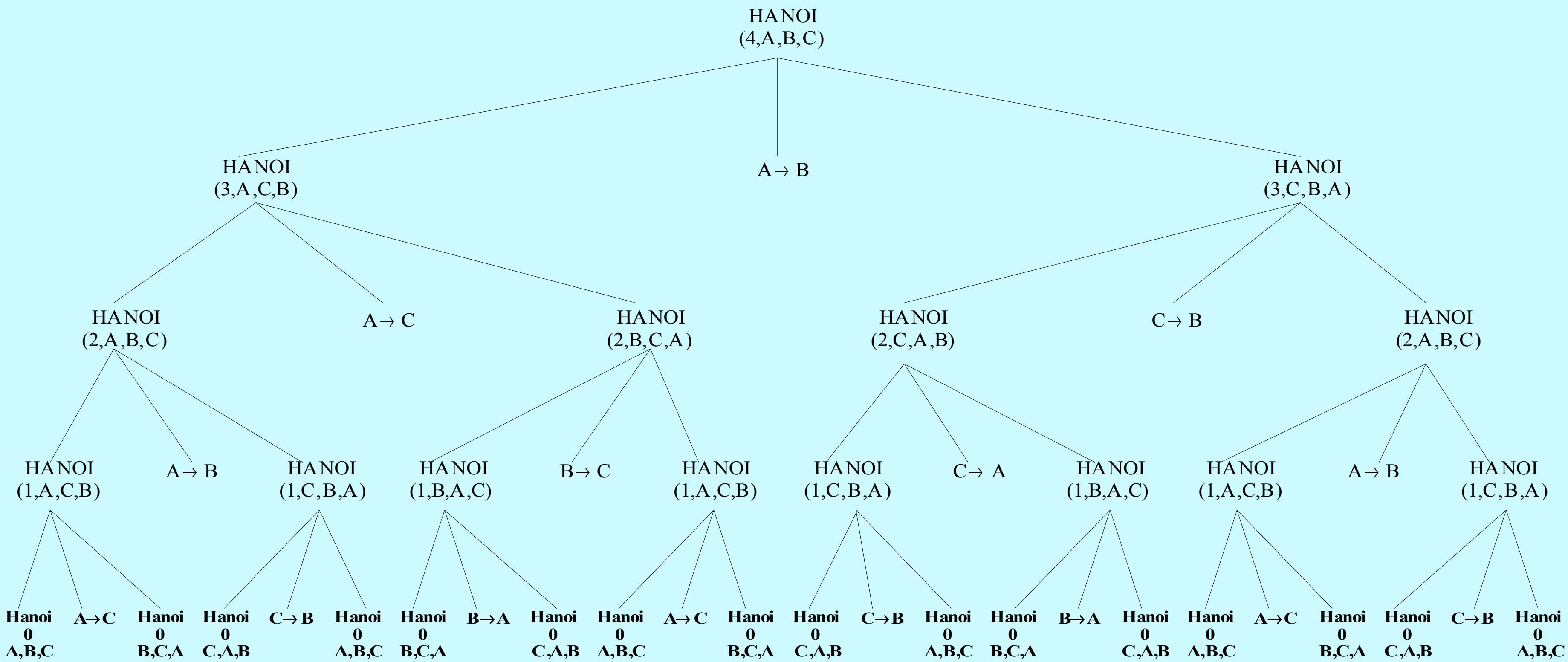
<u>Hanoi</u> (Nbd-1, dep, inter, arr)

<u>Déplacer</u> (dep, arr)

<u>Hanoi</u> (Nbd-1, inter, arr, dep)

<u>FINSI</u>

(voir programme de démonstration)



La chronologie des déplacements générée par l'exécution récursive de la fonction donne cet arbre qui doit se lire en parcours infixé (voir plus loin), c'est-à-dire en parcourant d'abord le sous-arbre de gauche, puis le sous-arbre central, et enfin le sous-arbre de droite (et ce de manière récursive). Dans l'arbre ci-dessus, x → y signifie: DEPLACER UN DISQUE DU SOMMET DE x VERS y. La chronologie des déplacements d'un disque est reprise ci-dessous. En grisé, les arrivées définitives d'un disque sur le socle choisi initialement comme arrivée.

A→ C	A→ B	C→ B	A→ C	B→ A	B→ C	A→ C	**A→ B**	C→ B	C→ A	B→ A	**C→ B**	A→ C	**A→ B**	**C→ B**

Etude de cas : tri rapide

- Quick sort
- Exposé du problème
- Solution

QuickSort : exposé

- Tableau TAB de bornes i, j   (0, N-1)
- Indice « pivot » du tableau TAB
- Découpe en deux sous-tableaux:
  - TAB [i, pivot-1]
  - TAB [pivot+1,j]
- Appel récursif pour opérer sur les sous-tableaux

QuickSort : solution

- QSORT (↓↑TAB, ↓bi, ↓bs)

  SI bi < bs

  Partition

  QSORT (↓↑TAB,↓bi,↓Gauche-1)

  QSORT (↓↑TAB,↓Droite+1,↓bs)

  FINSI

QuickSort : module Partition

- **Partition**

  G ← bi-1; D ← bs+1; X ← Tab[bi] ; // Tab[(bi+bs)/2]

  Exécuter

  Exécuter G++ Jusqu'àceque Tab[G] ≥ X

  Exécuter D- - Jusqu'àceque X ≥ Tab[D]

  SI G < D Swap (↓↑Tab[G], ↓↑Tab[D]) FinSi

  Jusqu'àceque G ≥ D

 FinPartition

Exercices

- Longueur d'une liste chaînée
- Recherche dichotomique dans un tableau
- Un tableau est-il un palindrome ?
- Insertion dans une liste chaînée
- …