Informatique tronc commun
Algorithmes usuels et correction des boucles

Sujet

16 octobre 2007

Introduction

Considérez les deux déclarations de procédure suivantes :

```
f := proc(x)
  if x = 0 then 1 else 1/x fi ;
end ;

g := proc(a,b)
local S,i ;
global f ;
  S :=0 ;
  for i from a to b do
    S := S+f(i) ;
  od ;
  S ;
end ;
```

Que renvoie un appel de `g(a,b)` ?

La procédure `g` appelle la procédure `f` qui a été définie avant. On peut gnéraliser ce genre d'écriture : une procédure peut faire appel à plusieurs autres procédures qui ont été définies avant dans le même programme. On peut même faire mieux : une procédure peut faire appel à elle-même. On dit alors que la procédure est *récursive*. Quand elle s'appelle elle-même on dit qu'elle fait un appel récursif. Il faut cependant faire très attention au problème de l'arêt des appels récursifs.

1 Un premier exemple

On considère la suite u définie par $u_0 = 1$ et pour tout $n > 0$, $u_n = 2u_{n-1} + 4$. On veut écrire une procédure `calcul` telle que `calcul(n)` renvoie la valeur de u_n. On tape :

```
calcul := proc(n)
  local res ;
    print (¨je vais calculer u¨,n) ;
    if n = 0 then res := 1 else res := 2*calcul(n-1) +' fi ;
    print (¨je viens de calculer u¨,n) ;
    res ;
end ;
```

a D'après vous, quand on appelle `calcul(5)`, que va-t-il se passer à l'écran ?

b Taper la fonction pour vérifier ce qui se passe.

c Montrer que `calcul(n)` renvoie bien la valeur de u_n.

d Combien de multiplications sont effecutuées par un appel de `calcul(n)` ?

2 Le calcul des puissances entières

Soit n un entier naturel et x un réel. On pose $u_n(x) = x^n$. On se propose d'étudier trois algorithmes de calcul de x^n.

2.1 Algorithme naïf itératif

a Pour $n \geq 1$, exprimer $u_n(x)$ à l'aide de $u_{n-1}(x)$. Que vaut $u_0(x)$?

b Soit u une variable locale. On pose H_n la propriété $u = u_n(x)$. Que doit-on faire pour avoir H_0 vraie ?
Pour $i \geq 0$, on suppose H_i vraie. Que doit-on faire pour avoir H_{i+1} vraie ?
Pour quelles valeurs de i doit-on faire ce qui précède dans le but de calculer x^n ?

c Donner une fonction `puissance` telle que `puissance(x,n)` renvoie la valeur de x^n. On utilisera une boucle `for` et on prouvera sa validité à l'aide de commentaires utilisant les propriétés H_k. On dira à présent que la propriété H_k est un invariant de boucle : elle est vraie tout au long de l'exécution du programme.

d Déterminer le nombre de multiplications effectuées par la fonction `puissance`.

2.2 Algorithme naïf récursif

On reprend la récurrence de la question précédente.

a Soit u une variable locale. On pose H_n la propriété $u = u_n(x)$. Que doit-on faire pour avoir H_0 vraie ?
Pour $n \geq 1$, on supose H_{n-1} vraie. Que doit-on faire pour avoir H_n vraie ?

b Donner une fonction récursive `puissance2` telle que `puissance2(x,n)` renvoie la valeur de x^n. Pourver l'arrêt et la validité de cette fonction.

c Déterminer le nombre de multiplications effectuées par la fonction `puissance2`.

2.3 Algorithme rapide de calcul des puissances

a Pour $n \geq 1$, on fait la division euclidienne de n par 2 : $n = 2p+r$ où r vaut 0 ou 1.
Comment obtient-on p en maple ? et r ?
Exprimer $u_n(x)$ à l'aide de $u_p(x^2)$ et de x.

b Pour $n \geq 1$, on suppose $H_n - 1$ vraie. Que doit-on faire pour avoir H_n vraie ?

c Donner une fonction récursive `puissance_rapide` telle que `puissance_rapide(x,n)` renvoie la valeur de x^n. Prouver son arrêt et sa validité.

d Déterminer le nombre de multiplications effectuées par la fonction `puissance_rapide`.

2.4 Le calcul de la multiplication par un entier

Reprendre les questions précédentes pour $u_n(x) = nx$.

3 Suites récurrentes du premier ordre

Soit E un ensemble, a un élément de E et f une applicaiton de E dans E. Soit u la suite définie pas $u_0 = a$ et pour $n \geq 0$, $u_{n+1} = f(u_n)$.

3.1 une version itérative

a Ecrire une fonction itérative `calcul_u` telle que `calcul_u(a,f,n)` renvoie la valeur de u_n. On prouvera la validité de cette fonction à l'aide de la propriété $H_k : u = u_k$.

b Déterminer le nombre d'appels de f effectués par cette procédure.

c Soit v définie par $v_O = 1$ et $v_{n+1} = 2 + v_n$. Préciser E, a, f, puis calculer v_{10} et v_{100}.

d Soit w définie par $w_O = 1$ et $w_{n+1} = \sqrt{(2 + w_n)}$. Préciser E, a, f. Ecrire une boucle d'affichage des valeurs de w_i pour $i = 0$ à 100. Propriétés de cette suite ?

e Soit t définie par $\mathtt{t}_0$ `= ""` (chaîne vide) et $\mathtt{t}_{n+1}$ `= "ab"|t`$_n$`|"ba"`. Préciser E, a, f. Ecrire une boucle d'affichage des valeurs de t_i pour i = 0 à 10. Décrire t_n.

3.2 une version récursive

a Donner une fonction récursive `calcul_u2` telle que `calcul_u2(a,f,n)` renvoie la valeur de u_n. On prouvera la validité et l'arrêt de cette fonction.

b Déterminer le nombre d'appels de f effectués par cette procédure.

c Tester cette fonction sur des exemples de votre choix

4 Suites récurrentes du second ordre

Soit E un ensemble, a un élément de E et f une applicaiton de $E \times E$ dans E. Soit u la suite définie pas $u_0 = a$, $u_1 = b$ et pour $n \geq 0$, $u_{n+2} = f(u_n + 1, u_n)$.

4.1 une version itérative

a Ecrire une fonction itérative `calcul_u` telle que `calcul_u(a,b,f,n)` renvoie la valeur de u_n. On prouvera la validité de cette fonction à l'aide de la propriété $H_k : u = u_k \wedge v = u_{k+1}$.

b Déterminer le nombre d'appels de f effectués par cette procédure.

c Soit v définie par $v_O = 1$, $v_1 = 1$ et $v_{n+2} = v_{n+1} + v_n$. Préciser E, a, f, puis calculer v_{10} et v_{100}.

4.2 une version récursive

a Donner une fonction récursive `calcul_u2` telle que `calcul_u2(a,b,f,n)` renvoie la valeur de u_n. On prouvera la validité et l'arrêt de cette fonction.

b Déterminer le nombre d'appels de f effectués par cette procédure.

c Tester cette fonction sur la suite v définie précédemment. Tentez de calculer v_{100}.

d Proposez une fonction récursive qui résoud le problème rencontré.

5 Recherche d'un zéro par dichotomie

Soit f une fonction continue sur l'intervalle $[a, b]$ et qui admet une seule valeur d'annulation c dans cet intervalle. On suppose que $f(a)$ et $f(b)$ sont de signe contraire, et on se propose de chercher une valeur approchée de c à e près.

On définit dans ce but deux suites u et v par :
- $u_0 = a$, $v_0 = b$
- Pour $n \geq 0$: on pose $m = \frac{u_n+v_n}{2}$ et si $f(u)f(m) \leq 0$ alors $u_{n+1} = u_n$ et $v_{n+1} = m$, et sinon, $u_{n+1} = m$ et $v_{n+1} = v_n$.

On pourrait montrer que les suites u et v convergent, avec pour limite c, avec pour tout n, $u_n \leq c \leq v_n$.

a Ecrire une procedure calcul_c renvoyant une valeur approchée de c à e près.

b Soit $g : x \mapsto \cos(x)$. g vérifie les hypothèses de l'énoncé dans $[0, \pi]$. Donner une approximation de sa valeur d'annulation à 10^{-6} près.

c On peut montrer que pour tout n on a $c \in [u_n, v_n]$ et que $|u_n - v_n| \leq \frac{b-a}{2n}$

(a) On choisit $a = 0$, $b = 1$ et $e = 10^{-6}$.Donner un majorant du nombre d'étapes faitres par cette procédure.

(b) Donner, en fonction de e, a, b uen majoration du nombre d'addition et de divisions que fait cette procédure.

(c) Que pensez-vous de la rapidité de cette procédure ?

(d) Modifier légèrement votre procéure pour qu'elle renvoir le nombre d'étapes faites. Vériier les résultats des deux quesitons précédentes sur des exemples.