

## 迴圈轉遞迴 (稿)

2023-07-20 by Yoxem (2024-10-05 : 修正發佈日期)

迴圈和遞迴其實習習相關，最近略讀論文 *Transforming Programs into Recursive Functions* (Myreen & Gordon, 2008)，提到迴圈如何轉換成遞迴的方法。我在這邊想到一些思路。一個迴圈可以理解為條件 (cond) 和滿足條件即執行的程式本體 (body)，即：

```
while (cond):  
    body
```

其中一些 while 外的變數，被 body 處理之後，在 body 的最後就會變。假設輸入到 body 改變數值的變數為  $x_i$  ( $i = 0, 1, \dots$ )，輸出的結果是  $x'_i$ ，那我們可以假設  $\vec{x}$  是  $x_i$  組成的向量， $\vec{x}'$  是  $x'_i$  組成的向量，則 body 可以視為函數，且  $\vec{x}' = \text{body}(\vec{x})$ 。

舉下列求除法商數的 Quotient( $a, b$ ) =  $a$  除以  $b$  的商數"為例，為求簡化， $a, b$  均為自然數：

```
QUOTIENT( $a, b$ ):  
1  let q = 0           // 商數  
2  while  $a \geq b$ :     // cond  
3       $a \leftarrow a - b$  // body 的第 1 行  
4       $q \leftarrow q + 1$ : // body 的第 2 行  
5  return q
```

其中  $a, b$  和  $q$  是 body 外的變數， $\text{body}(a, b, q) = (a - b, b, q + 1)$ ，比較下， $\vec{x} = (a, b, q)$ ， $\vec{x}' = (a - b, b, q + 1)$  然後可以把  $\text{while}(\text{cond})\{\text{body}\}$  組合  $\vec{x}$  想成這樣的 OCaml 函數：

(\* 以下括號非必要則略 \*)

```
WHILE cond body XVec=  
if cond then  
    let NewXVec = body XVec in (* NewXVec = body XVec *)  
    WHILE cond body NewXVec (* 帶 NewXVec 重新進去 WHILE 函數，做 if 判斷的手續 *)  
else  
    XVec (* XVec 不變 *)
```

另可以這樣改寫，其中 cond 視為輸入  $\vec{x}$  的函數：

```
let WHILE cond body XVec =  
if (cond XVec) then  
    WHILE cond (body XVec)  
else  
    XVec
```

那麼這個除法可以這樣寫成類 OCaml 風格：

.....

```
let q = 0 in
(* 原樣 *)
while a >= b do
  let a = a - b
  let q = q + 1
```

但是要找出 while 外變數，可以找出 a, b, q，且把狀態更新改為 let...in 模式，則

.....

```
let q = 0 in

let cond = fun a b q -> a >= b in
WHILE cond
  (fun a b q ->(
    let a = b - a in
    let q = q + 1 in
    (WHILE cond [a; b; q] )))
a b q (* XVec*)
```

用 let 的規約規則轉換一下變成尾遞迴

.....

```
let q = 0 in

let cond = fun a b q -> a >= b in
WHILE cond
  (fun a b q ->(
    (WHILE cond [b - a; b; q + 1] )))
a, b, q
```

最後修改：

.....

```
let cond = fun a b q -> a >= b in
letrec WHILE cond =
  (if (cond X) then
   (WHILE cond (b - a) b (q + 1))
  else
```

```
[a; b; q] in  
WHILE cond a b q (* 執行 LOOP *)
```