

使用 RE 與 bash 實作試算表語言

作者：一卡

時間：2013 年 05 月 22 日

數值計算

加法包含純數值計算與資料檢查兩部份，純數值計算的部份以加法器為基礎原理，透過查表法處理位數相加與進位，檢查部份則判別輸入資料是否為整數，如果數值處理正確，則回傳值必為整數，因此不檢查回傳值的型別，確定兩者皆為整數，則分析正負號，根據正負號決定計算方式，請參考以下演算法。

```
type Sign = + | -

Check_Sign :: Sign -> Sign -> (Int -> Int -> Int)
Check_Sign sign_1 sign_2
    | sign_1 == sign_2 = Calculate
    | otherwise = (\int_1 int_2 -> Calculate int_1 (Complement int_2))
```

決定計算方式後，根據兩數的絕對值大小，決定何者為先計算的數值，較大的數值將決定計算後數值的正負數，由於使用加法來處理兩數相減的情況，如果先將數值轉為二進位計算後再轉回十進位將浪費許多時間，因此需要使用補數法來轉換數值，並透過補數法與加法取代減法，這裡採用「n進位補數法」，證明如下。

[證明]

於 n 進位時，設有一個數值 a，若存在一個數值 b 使以下條件成立，則我們稱 a 與 b 互補，若稱 a 為正數，則 b 即稱負數，反之亦同。

$$a + b = 0$$

假設存在兩個數值 a、b，若 a 為 r 位數，b 為 k 位數，假設 b 的補數為 b'，則可獲得以下結果。

$$|b| < |a| < |b'| \text{ or } |b'| < |a| < |b|$$

因為 a、b 正負號相反，相加後所得結果必小於兩者絕對值中較大者，即代表從 r 與 k 中取較大者 $\max(r, k)$ ，若相加後得到的位數 j 大於 $\max(r, k)$ ，則我們只取 $\max(r, k)$ 位數，稱此為「補數法最大位數取位原則」。

設 n 進位 r 位數的最大值為 b'，根據取位原則我們可以得到如下結果。

$$\begin{aligned} b' + 1 &= 0 \\ &= a + b \\ \Rightarrow b &= b' - a + 1 \end{aligned}$$

若 n 為二，即為常見的二補數法，設 n 為 10，則十進位補數法最大值如下。

$$9 \times 10^r + 9 \times 10^{(r-1)} + \dots + 9$$

若 a 值以相同方式表示則如下。

$$(a_r) \times 10^r + a_{(r-1)} \times 10^{(r-1)} + \dots + a_1$$

則可得知對於 $b' - a$ 的所有位數皆有以下公式。

$$f(x) = 9 - x$$

根據此公式求完 $b' - a$ 的值後加一即為 a 的補數 b 。