

J602 の新しい機能 M. について

西川 利男

先月の例会で志村正人氏より J602(beta)のご提供とその紹介があったが、新しい機能 M. については、まだ用途不明ということであった。その後、Help の Vocabulary で私なりにやってみて、これは大変な機能であることがわかった。

M. は定義した動詞を高速実行する機能を持つ副詞で、そのしかけは引数やその実行の結果を記憶・保存し(memorize)、再利用することで高速化を行う。したがって繰り返し再帰呼び出し(multiply-recursive call)の動詞に用いて絶大な効果を発揮する。

再帰呼び出しの例としては、次のフィボナッチ数の計算をとりあげる。フィボナッチ数の計算アルゴリズムは以下のようである。

n=0 のとき fib(0) = 0

n=1 のとき fib(1) = 1

n>1 のとき fib(n) = fib(n-1) + fib(n-2)

(1) 名前をつけた定義動詞の場合

上のアルゴリズムをそのまま、J の if. do. else. end. 構文で定義し、副詞 M. を付加する。従来の定義もつくり、これらを実行時間の計測の動詞 timer で比較した。

```
fib=: 3 : 0 M.      高速実行の副詞 M. を使用する
  if. 1>:y do. y else. (fib y-1)+fib y-2 end.
)
fibx=: 3 : 0      従来の定義
  if. 1>:y do. y else. (fibx y-1)+fibx y-2 end.
)
  fib"0 i.18
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597
  fib 32
2178309
  timer=: 6!:2 実行時間の計測ツール
  timer 'fib 32'
0.000479377
  timer 'fibx 32'
43.696

再度実行すると引数を記憶しているの、さらに短時間で実行される。
  timer 'fib 32'
2.62393e_5
```

(2) 名前を付けない tacit 定義の場合一同様に高速化される。

```
timer '+/@:($:"0)@:(-&1 2)`]@.(1>:)] M. 32'
0.000186387
timer '+/@:($:"0)@:(-&1 2)`]@.(1>:)] 32'
8.61349
```

Help の Vocabulary には分割(partition)と組み合わせ(combination)への適用例もあり、J の興味深いコーディングもあげてあるが、これは別の機会にしたい。

