

プログラムのページ

担当 吉 沢 正

7005 自然数の巾和公式の生成と評価

中西 正和 (慶応義塾大学・工学部)

各 p について $S_p = 1^p + 2^p + \dots + n^p$ の公式を得ると

いう問題を解くのは、一般に Bernoulli 数を使えばできるが、そんなものを使わなくても、高校の教科書にのっている方法で、簡単に解くことができる。 $S_0 = n$

..... KLISP-1. SYSTEM

```

YJOB      MF-L-014-F  PSUM      M.NAKANISHI

1      *      SUMMATION OF I ** P
2      *
3      *
4      *      FORM GENERATOR
5      *
6      PSUM[P]
7      = [ZEROP[P]->'N';
8          T->LIST['QUOTIENT';LIST['DIFFERENCE';
9              LIST['SUB1';LIST['EXPT';'(ADD1 N)';
10             ADD1[P]]];[ONEP[P]->'N';
11                 T->CONS['PLUS';
12                     SUMB[ADD1[P];SUB1[P]]]];
13             ADD1[P] ] ;
14      *
15      SUMB[P;K]
16      = [ZEROP[K]->'(N)';
17          T->CONS[LIST['TIMES';COMBN[P;K];PSUM[K]];
18              SUMB[P;SUB1[K]] ] ] ;
19      *
20      COMBN[P;K]
21      = [ZEROP[K]->1;
22          EQ[P;K]->1;
23          T->PLUS[COMBN[SUB1[P];SUB1[K]];
24              COMBN[SUB1[P];K] ] ] ;
25      *
26      *      EVALUATOR FOR TESTING
27      *
28      EPSUM[P;N]
29      = PROG[[A] ;
30          PRINT[BLANK] ;
31          A := PRINT[PSUM[P]] ;
32          RETURN[EVAL[A];LIST[CONS['N';N]]] ] .
33      *
34      *      TEST CASES
35      *
36      *
37      VERBOS[NIL] ;
38      *
39      EPSUM[2;10] ;
40      *
41      EPSUM[5;10] ;

```

**** GLISP PROGRAM TERMINATED! ****

PROCESSING TIME 0 MIN 48n MSEC

プログラム・リスト

と、 S_1, S_2, \dots, S_{p-1} がわかってさえいれば、 S_p はすぐに得られる。これは帰納的定義の好例である。そこで、LISP を使ってこの公式を作りだすプログラムを作ってみる。ついでに、この公式に数を与えて値を求める。

1. アルゴリズム

高校の教科書の級数のところに出ているものをそのまま利用する。

$$(i+1)^{p+1} - i^{p+1} = \sum_{k=0}^p \binom{p+1}{k} i^k$$

この式の i に順に $1, 2, \dots, n$ を与え、すべてを加えることにより

$$(n+1)^{p+1} - 1^{p+1} = \sum_{k=0}^p \binom{p+1}{k} \sum_{i=1}^n i^k$$

を得る。これからつぎの帰納的な定義ができる。

$$S_p = \begin{cases} n & (p=0) \\ \frac{(n+1)^{p+1} - 1 - \sum_{k=0}^{p-1} \binom{p+1}{k} S_k}{p+1} & (p>0) \end{cases}$$

2. プログラム

LISP は帰納的定義を特徴とするために、このようなアルゴリズムに対しては1対1の変換ができる。値を計算するために、この公式も LISP プログラムにしておく。すなわち、LISP で LISP プログラムを生成

..... KLISP-1. SYSTEM

```
FUNCTION EVALQUOTE HAS BEEN ENTERED, ARGUMENTS..
DEFINE
((PSUM (LAMBDA (P) (COND ((ZEROP P) (QUOTE N)) (T (LIST (
QUOTE QUOTIENT) (LIST (QUOTE DIFFERENCE) (LIST (QUOTE
SUB1) (LIST (QUOTE EXPT) (QUOTE (ADD1 N)) (ADD1 P)))) (
COND ((ONEP P) (QUOTE N)) (T (CONS (QUOTE PLUS) (SUMA (
ADD1 P) (SUB1 P)))))) (ADD1 P)))))) (SUMB (LAMBDA (P K)
(COND ((ZEROP K) (QUOTE (N))) (T (CONS (LIST (QUOTE
TIMES) (COMBN P K) (PSUM K)) (SUMB P (SUB1 K)))))) (
COMBN (LAMBDA (P K) (COND ((ZEROP K) 1) ((EQ P K) 1) (T
(PLUS (COMBN (SUB1 P) (SUB1 K)) (COMBN (SUB1 P) K))))))
) (EPSUM (LAMBDA (P N) (PROG (A) (PRINT BLANK) (SETO A
(PRINT (PSUM P))) (RETURN (EVAL A (LIST (CONS (QUOTE N)
N))))))))))
END OF EVALQUOTE, VALUE IS..
(PSUM SUMB COMBN EPSUM)
```

```
FUNCTION EVALQUOTE HAS BEEN ENTERED, ARGUMENTS..
VERBOS
(NIL)
END OF EVALQUOTE, VALUE IS..
NIL
```

```
FUNCTION EVALQUOTE HAS BEEN ENTERED, ARGUMENTS..
EPSUM
(2 10)
QUOTIENT (DIFFERENCE (SUB1 (EXPT (ADD1 N) 3)) (PLUS (TIMES
3 (QUOTIENT (DIFFERENCE (SUB1 (EXPT (ADD1 N) 2)) N) 2)
) N)) 3)
END OF EVALQUOTE, VALUE IS..
385
```

← 生成された公式

← 値

```
FUNCTION EVALQUOTE HAS BEEN ENTERED, ARGUMENTS..
EPSUM
(5 10)
```

```
(QUOTIENT (DIFFERENCE (SUB1 (EXPT (ADD1 N) 6)) (PLUS (TIMES
15 (QUOTIENT (DIFFERENCE (SUB1 (EXPT (ADD1 N) 5)) (
PLUS (TIMES 10 (QUOTIENT (DIFFERENCE (SUB1 (EXPT (ADD1
N) 4)) (PLUS (TIMES 6 (QUOTIENT (DIFFERENCE (SUB1 (EXPT
(ADD1 N) 3)) (PLUS (TIMES 3 (QUOTIENT (DIFFERENCE (
SUB1 (EXPT (ADD1 N) 2)) N) 2)) N)) 3)) (TIMES 4 (
QUOTIENT (DIFFERENCE (SUB1 (EXPT (ADD1 N) 2)) N) 2)) N)
) 4)) (TIMES 10 (QUOTIENT (DIFFERENCE (SUB1 (EXPT (ADD1
N) 3)) (PLUS (TIMES 3 (QUOTIENT (DIFFERENCE (SUB1 (
EXPT (ADD1 N) 2)) N) 2)) N)) 3)) (TIMES 5 (QUOTIENT (
DIFFERENCE (SUB1 (EXPT (ADD1 N) 2)) N) 2)) N)) 5)) (
TIMES 20 (QUOTIENT (DIFFERENCE (SUB1 (EXPT (ADD1 N) 4))
(PLUS (TIMES 6 (QUOTIENT (DIFFERENCE (SUB1 (EXPT (ADD1
N) 3)) (PLUS (TIMES 3 (QUOTIENT (DIFFERENCE (SUB1 (
EXPT (ADD1 N) 2)) N) 2)) N)) 3)) (TIMES 4 (QUOTIENT (
DIFFERENCE (SUB1 (EXPT (ADD1 N) 2)) N) 2)) N)) 4)) (
TIMES 15 (QUOTIENT (DIFFERENCE (SUB1 (EXPT (ADD1 N) 3))
(PLUS (TIMES 3 (QUOTIENT (DIFFERENCE (SUB1 (EXPT (ADD1
N) 2)) N) 2)) N)) 3)) (TIMES 6 (QUOTIENT (DIFFERENCE (
SUB1 (EXPT (ADD1 N) 2)) N) 2)) N)) 6)
```

END OF EVALQUOTE VALUE IS.
220825

RUN TIME 0 MIN 23560 MSEC

**** END OF KLISP JOB! ****

結果

する。LISP は使用者が言語翻訳の機能を使うことができる (eval あるいは evalquote 関数を使用することができる) ため、プログラムを生成すること、このプログラムを実行させることを同時にできる。

関数 psum は、LISP プログラムの S_p を作りだすもの。sumb はその副関数で、 $\sum_{k=0}^{p-1} \binom{p+1}{k} S_k$ を作りだす関数である。

できあがった公式の n に数を代入するのは、eval 関数の第 2 引数のアソシエーション・リストをとおして行なう。

公式の形をみるために、途中で print 擬関数を使用している。

3. 結果

epsum 関数は、公式を生成し、代入結果を得る関数であり、epsum [$p; n$] の値は $\sum_{i=1}^n i^p$ の値である。こ

こでは $p=2, 5$ について $n=10$ の場合を評価している。

このプログラムは慶応義塾大学情報科学研究所の TOSBAC-3400 KOS を使用し、私が作成した KLISP システムの GLISP 言語で組んである。KLISP の数値演算の速度が遅いため、全実行時間の約 9 割が生成プログラムに n を与えて値をだすために使用されている。プログラムの生成だけでは、約 0.8 秒である。

参考文献

- 1) 中西正和: KLISP 説明書. 慶応義塾大学情報科学研究所 (1969).
- 2) 中西, 小林, 他: KLISP プログラム集. 慶応義塾大学情報科学研究所 (1970).

(受付 昭和 45 年 4 月 13 日)