ACKERMANN considered the following function $\xi$ in 1928 [1] and PÉTER modified it in 1935 [2], as an example of double recursive but not primitive recursive function:

$$
\xi(n, b, a) = \begin{cases} 
  a + b & \text{if } n = 0 \\
  a(n-1, a) & \text{if } n > 0 \land b = 0 \\
  \xi(n-1, \xi(n, b-1, a), a) & \text{otherwise,}
\end{cases}
$$

where

$$
a(n, a) = \begin{cases} 
  n & \text{if } n \leq 1 \\
  a & \text{otherwise.}
\end{cases}
$$

We know that double recursive functions are included in the general recursive functions and that the latters are possible theoretically to be calculated by a computer. But, in the case of not primitive recursive functions their programings are very complicated, though in the case of primitive ones they are simple. Because that the formers need all of past datas without upper bound in the number of them, therefore they require subroutines to call themselves essentially in addition to other subroutines. While the latters need only past datas with an upper bound, therefore they require only other subroutines.

In the case of primitive recursive function if we neglect the capacity of individual memories, we need only one memory for one variable (including implicit variables i.e. not only free variables but also bound variables). But, in the case of not primitive recursive one even if we neglect it, we need essentially infinite (countable) memories for one variable. For example, in the case of the computation (Figures 1 and 2) of Péter's $\xi$, in the following Figure 1, BB requires infinite memories.

Of course, if we express a sequence of natural numbers $<a_1, ..., a_n>$ as $2^{a_1}3^{a_2}...p_{n-1}^{a_n}$, we can do one memory for BB, too. In such case, each part of left side in the following table (Figure 5) becomes as a corresponding part of right side in the same one. Yet, RM and PRIM in Figure 5 are as showed by Figures 3 and 4.

*) The author is partially supported by Grant-in-Aid for Scientific Research Project No. 56340009.

---

**PROGRAMMING OF PÉTER FUNCTION**

By

Masamichi Wate

(Recieved October 28, 1981)
XI(N, B, A)

Read N, B, A

N → M
B → ANS
1 → CNT
1 → BB(1)

CNT = M + 1

yes

ANS = 0

yes

A + ANS → ANS
ALPHA(M - CNT, A) → ANS

no

yes

CNT = BB(1)

no

BB(CNT - 1) → BB(CNT)
CNT + 1 → CNT

no

yes

M → M

BB(CNT) = 0

yes

no

yes

CNT = 0

no

yes

XI(N, B, A) = ANS

STOP

Fig. 1
PROGRAMMING OF PÉTER FUNCTION

**Fig. 2**

ALPHA(N, A)

\[ N \leq 1 \]

\[ N \rightarrow \text{ANS} \]

\[ A \rightarrow \text{ANS} \]

\[ \text{RETURN} \]

**Fig. 3**

RM(BB, PR)

\[ BB \rightarrow WK1 \]

\[ PR \rightarrow WK2 \]

\[ WK1 \rightarrow WK1/WK2 \rightarrow WK2 \rightarrow \text{REM} \]

\[ \text{RETURN} \]

**Fig. 4**

PRIM(CNT)

1 \rightarrow CNT2

2 \rightarrow PR

\[ CNT2 = CNT \]

\[ CNT2 + 1 \rightarrow CNT2 \]

\[ PR + 1 \rightarrow PR \]

2 \rightarrow Q

\[ \text{RM}(PR, Q) \rightarrow \text{REM} \]

\[ \text{REM} = 0 \]

\[ Q + 1 \rightarrow Q \]

\[ Q = PR \]

\[ \text{RETURN} \]
Fig. 5
References


Masamichi Wate
Department of Mathematics
College of Humanities & Sciences
Nihon University
Sakurajosui 3-25-40 Setagaya-ku
Tokyo, Japan