

214 Tape Operated Calculating

and Copying Robot

高橋秀俊 和田英一

(東京大学理学部)

この計算機は、パラメトロンによる作用、性能のデモンストレーションのために、またパラメトロンによる10進加算回路、その他各種の基本回路の実験のために試作され、6単位の印刷電信機の入出力装置、 $\pm A_1 B_1 \pm A_2 B_2 \pm A_3 B_3 \pm \dots$ という乗算と累計をおこなう演算部門、この二つを結びつける制御部門、そしてパラメトロン駆動電源より構成されている。プログラミングを簡単にするため、数式プログラムを採用した。すなわち A_i, B_i 等(正負の整数)を加減算符号(+、-)と乗算符号(\times)でつなぎ合わせて数式とし、これを鍵盤で打って最後に等号(=)の鍵を押すと、演算結果は自動的に等号の次に印刷され、等式が完結される。この際不要な零は除外される。またこの計算機では、2台のテープ送信機による命令の読み込みも可能で、この場合は実行中の命令文の印刷をおこなわない。一つの命令文の実行が終ると、計算機は次の命令文が始まるまで、読み込まれた文字の複写を続けるが、これは、一たん計算機からとり出された出力テープを、ふたたび入力テープとするのに必要だからである。一回で実行できる演算が乗算の累計だけで、しかも内部記憶回路のないこの計算機では、テープによる記憶を使用しなければ、ほとんどなにもしないといつてよい。2台のテープ送信機は、命令テープに含まれる呼び出し符号によって動作し、一方の送信機は、呼び出し符号で相手の送信機を呼び出すと、ふたたび呼び出されるまでは、制御の権利を持たない。2台の送信機を用いるこの方式は、相関やフーリエの計算に便利であるばかりでなく、命令テープの編集にも有用である。

さて $3x^2 + 2x + 1$ で、 $x = 10$ を計算するときのことを考えよう。プログラムは右のよ

$$\text{演 } 3 \times 10 + 2 = x \times 10 + 1 =$$

うになる。ここに演の字は命令開始をおわらし、一つの命令が終ると、次に演の字が来るまでは複写がおこなわれ、いま計算機はその状態で、命令テープを左から読み始めた。演の字を読み込んだ計算機は、ここで複写を中止して、 $3 \times 10 + 2$ を計算し、次の等号を読む。ここでテープの読み込みを中止して、この演算結果をさん孔する。次に $x \times 10 + 1 =$ を複写するので、計算機から出て来たテープは

$$\text{演 } 32 \times 10 + 1 =$$

右のようである。複写状態にある計算機にいまのテープを入れてみると、今度は 321 と最後の結果を印刷する。2台のテープ送信機を用いて、 $y = 2x + 1$ の変数 x を $1, 2, \dots$ と順に変えたとき、その函数 y を計算する

$$\text{演 } 2 \times b + 1 =$$

$$1 a 2 a 3 a 4 a \dots$$

としよう。左の上のテープを輪状にし、演の字から a の送信機で読み込む。 $2 \times b$ まで来ると b の送信機を呼び出し、 $1 a$ と読んで、

また a の送信機を呼び返す。これで 2×1 ができ、 $+ 1 =$ でまず $x = 1$ のときの y が印刷される。ふたたび a の送信機は演の字を読み込み、 $2 \times b$ で次に b の送信機から $2a$ を読んでまた a の送信機に戻り $2 \times 2 + 1$ を計算する。以後同じようにして演算が実行されて行く。

以上の例から察せられるように、この計算機は、演算と複写の繰り返し、 a の送信機と b の送信機との呼びかたえ、この二つの方式を組み合わせると、かなり複雑なことをおこない得るが、あまり複雑にしてしまうと実用的でなくなる。一番実用的なのはやはり相関とフーリエの計算であろう。フーリエ等では、計算に使う式の形がきまっているから、一度このテープを作っておくと、あとはその時々、周期函数のテープを作るだけで充分である。1周期を24に分け、3桁の精度で一つの係数を計算するのに約50秒を要した。行列の掛算だの、カーウフロップターとしての使用例は、ここでは省いておきたい。

Tape Operated Calculating and Copying Robot 拾遺

和田英一 (IIJ 技術研究所)

情報処理学会誌に「和田英一先生が研究報告に初めて登場するのは、随分と遅くて、... この点では、私の方がはるかに先輩...」と記せし仁あり [0]。子曰、... 人不知而不愠，不亦君子乎。(i1) 随伴の復刻予稿は私の修士論文 [1] の抄録である。パラメトロン計算機 PC-1[2] のイニシアルオーダー R0 のハックや、ライブラリの整備より 2 年ほど前の仕事だ。(情報処理学会は未だ存在せず。) 1955 年、修士課程進学するとき、パラメトロンはあったが 2 周波メモリーは未開発だったので、修士 1 年ではテープレコーダをメモリーに使う実験をした。デジタル信号と Licklider 流の音声の零交差波を書き込み読み出して遊んでいた。修士 2 年では、前年に高橋先生が設計し、同僚が組み立てたパラメトロン 10 進計算機から押しボタン入力、CRT 出力を撤去し、代りに印刷電信機と紙テープ送信機を接続して、その制御回路を設計/製作/評価した。それを修士論文にした。紙テープ送信機 2 台をコルーティン風に制御し、命令文を部分的にコピーしたりそれに中間結果を挿入したりし、さらに出力を再入力すると、いわゆるキューマシンとなり、かなり複雑な計算が可能となった。暫時の後 Dijkstra もコピーによる計算制御方式を提案していると知る。

前段に於て命令文といったのは当時 BNF など知らず、計算命令の構成法記述には各構成要素に品詞を与え、自然言語の文法様の表現を使ったからである。 <45 星霜を経て... >

2001 年の夏のプログラミング・シンポジウム、大人のためのプログラミングコンテスト [3] の言語 FIFOL でキューマシンと再会した。三つ子の魂が生き残っていたのか、予選は楽勝だった。修論の計算機の遅かった如く、FIFOL のプログラムもアーキテクチャが単純なだけあって、データが使えるようになるまで待たねばならず、あまつさえプログラミングも困難至極であった。

転じて 2004 年秋 9 月、Niel Jones にそそのかされ、FIFOL の自己シミュレータ (万能 FIFOL 機械—UFM) を考える羽目になる。即座に出来ると直感するも、通勤の車中でも考え続け、工夫に工夫を重ね、大方は予想通りに完成し、第 46 回プログラミング・シンポジウムで報告した [4]。このプログラムハックでは、FIFOL 用各種ツールも開発/改良した。シミュレータはさらなり。記号シミュレータ、変数置換プログラム列発生器、FIFOL UFM コード変換器などである。UFM のデモ例題の π を計算する Spigot アルゴリズムなど、ツールなしでは FIFOL や UFM への変換はまらずおぼつかない。 子貢問為仁，子曰，工欲善其事，必先利其器，... (xv10)

IT の発展にも CS の進展にも皆目役立たぬも、自分では楽しくプログラムを書き続けられ、幸せなハッカー人生を過ごせたと感謝している。子曰、知之者不如好之者，好之者不如樂之者。(vi20)

参考文献

- [0] 萩谷昌己: “研究会たより:電子化された研究報告”, 情報処理学会誌, Vol.43, No.4(2002 年 4 月), p.461.
- [1] 高橋秀俊, 和田英一: “鑽孔テープを使ったパラメトロン自動計算機”, 電気通信学会電子計算機研究専門委員会資料, 1957 年 1 月 18 日.
- [2] 高橋秀俊, 和田英一他: “Parametron Computer PC-1”, 電気通信学会第 57 回電子計算機研究専門委員会資料, 1958 年 9 月 25 日 <http://www.iijlab.net/~ew/pc1/singakukai.html>.
- [3] 田中哲朗: “プログラミングコンテスト実施報告”, 夏のプログラミング・シンポジウム「プログラミングの鉄人 — プログラミングの技」報告集, pp.3-14(2001 年 8 月 22 日 ~ 8 月 24 日).
- [4] 和田英一: “万能 FIFOL 機械”, 第 46 回プログラミング・シンポジウム報告集, pp.71-82(2005 年 1 月 11 日 ~ 13 日).