文　献　紹　介

85. オイラー常数の計算

D.W. Sweeney: On the Computation of Euler's Constant [Mathematics of Computation Vol. 17, No. 82, April 1963, pp. 170～178]

Euler の常数

$$\gamma=\lim_{n\to\infty}\left(1+\frac{1}{2}+\frac{1}{3}+\cdots\cdots+\frac{1}{n}-\ln n\right)$$

を exponential function $-E_i(-x)$ の展開を使った新しい方法で 3,566 桁計算した．その結果とこの計算の一部として計算された $\ln 2$ の 3,683 桁の値が表として与えられている．この方法は高次の Bernoulli number の計算に要求される複雑な計算を避けるために採用された．

方法は Exponential function

$$-E_i(-x)=\int_x^\infty\frac{e^{-t}}{t}dt=-\gamma-\ln x+S(x)$$

ただし

$$S(x)=x-\frac{x^2}{2.2!}+\frac{x^3}{3.3!}-\cdots\cdots$$

および，その漸近展開

$$-E_i(-x)=\int_x^\infty\frac{e^{-t}}{t}dt\simeq\frac{e^{-x}}{x}\left(1-\frac{1}{x}+\frac{2!}{x^2}+\cdots\cdots\right)=R(x)$$

から得られる式

$$\gamma\simeq S(x)-\ln x-R(x)$$

による．すなわち大きな x に対して $S(x)-\ln x$ が γ にたいして誤差 $R(x)$ をもつ近似を与える．

計算は，2 進法の計算機を使うので，$x=2^{13}=8192$ とすると $R(x)=0.22190\cdots\cdots\times10^{-8561}$ を得る．$\ln 2$ の計算は容易にできる．$S(x)$ も計算は簡単であるが，収斂するには，$x=8{,}192$ で 3,000 項が必要であり，また切捨，cancellation のため，答の 3 倍の桁数の計算が必要であった．

プログラムは，誤りを防ぐため，異なる二つの計算を用い，印刷も異なるプログラムでなされた．第一の部分は，

$$13\ln 2=13\left[\frac{5}{1\cdot1\cdot2^3}+\frac{11}{2\cdot3\cdot2^5}+\frac{17}{3\cdot5\cdot2^7}+\cdots\cdots\right]$$

および

$$S(x)=\left(x+\frac{x^3}{3\cdot3!}+\cdots\cdots\right)-\left(\frac{x^2}{2.2}+\frac{x^4}{4.4!}+\cdots\cdots\right)=x+\Sigma D_{2n+1}-\Sigma D_{2n}$$

を計算する．

ただし

$$C_{2n}=\frac{x^{2n+1}}{(2n)!}=\frac{x^2}{2n(2n-1)}D_{2n-2}$$

$$D_{2n+1}=\frac{C_{2n}}{(2n+1)^2},\quad D_{2n}=\frac{C_{2n}}{2nx}$$

を利用する．この二つの結果から γ を得る．

第 2 の部分は，

$$\ln 8{,}192=13\,B_1,\quad B_n=\frac{1}{2}\left(\frac{1}{n}+B_{n+1}\right)$$

を $n=12300$ から始めて計算し，次に

$$S(x)=xA_1,\quad A_n=1-\frac{nx}{(n+1)^2}A_{n+1}$$

を $n=30{,}000$ から始めて計算しその結果 γ を得る．

計算時間は IBM 7094 の engineering model を使い，第一の部分に 20 分，第 2 に 35 分，印刷に 3 分，計 58 分であった．IBM 7090 では 114 分であった．

Knuth の 1,271 桁を得た方法と比較して，1,271 桁求めるには，この方法によれば Knuth の方法の 2/3 の時間で計算できる．Knuth の方法で 3,566 桁を求めるために Bernoulli number に要求される精度の estimation が難しく，3,566 桁では，比較はできなかった．　　(中川圭介)

86. 非同期順序スイッチ回路の状態変数指定法

C.N. Liu: A State Variable Assignment Method for Asynchronous Sequential Switching Circuits [J. of A.C.M., Vol. 10, No. 2, April 1963, pp. 209～216]

非同期回路では，同期回路と違って，回路の内部状態と入力との二つから，次の状態は unique にはきまらないが，流れ表を使って最終的な動作を記述する方法がある．これは，一つの行列で，列は特定の入力状態に対応し，行は一つの内部状態に対応している．要素は，つぎの状態と出力を表している．

次の状態が，元の状態と同じとき安定状態と呼ばれる．内部状態のそれぞれを 2 進数で指定すれば，非同期回路設計の問題は組み合わせスイッチ回路の問題になってしまう．

いま，回路に循環動作はないもの（循環動作を一つの安定状態とみて変数を一箇増やして問題を簡単にす

ることもできるが）として，過渡的な状態変化をつねに一箇の変数だけがその値を変えるように（その前後の状態は隣接しているという），内部状態を指定することは，変数をふやしさえすれば可能であると証明される．二つ以上の変数が同時に変ると，次の状態をきちんと指定できない場合があって，critical race（きわどい競走）と呼ばれる．“きわどい競走”を許すような指定の仕方は避ける．

ここで紹介する方法は，流れ表から状態の遷移図を作ることから始まる．次に，安定状態を多く含む列について，安定状態を区別するだけの2進変数を用意し，遷移図を見て，結線があれば隣接するようにその列内の安定状態を符号化する．これを別の列についても続け，安定状態の少ない列については，その列内で同じ状態にも別の符号をあてる．できあがった指定表に同一の列がいくつか表われたときには縮少する．

こうして作られた指定表が，最少の変数であるかどうかはわからない．得られた符号が等距離誤り訂正符号ならば，“きわどい競走”がないことが証明されているが，この符号では 2^m 個の状態を区別するのに 2^m-1 ビット必要なことを考える上述の方法はかなり能率のよい方法である．

実際，説明に用いている例では，14 の状態で，入力が4種の流れ表を指定するのに5ビットで間にあっている．

（川合英俊）

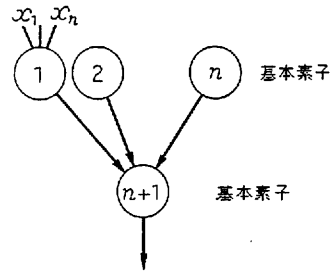
87. 三角回路の信頼度理論

S. Amarel and J.A. Brozozowski: Theoretical Consideration on Reliability Properties of Recursive Triangular Switching Networks [Redundancy Techniques for Computing Systems edited. by R.H. Wilcox and W.C. Mann, Spartan Books, 1962. pp. 70～128]

一時性誤り（tempory error）のあるゲート型論理素子を用いて，高信頼度の論理回路を構成することを論じている．

本論文では，各論理素子は，論理関数 f_q を実現すべく設計されているが，f_q 以外にも $f_1, \cdots\cdots f_m$ をそれぞれ確率 $p(f_i)$ で表現すると考える．

この種の素子（$n+1$）個を，第1図のように結合した回路を，1次の三角回路という，この回路は，各基本素子が種々の状態を取るに従って，さまざまな論理関数を実現する．しかし適当な仮定のもとでは，基本素子よりも高い確率で f_q を表現するので，第1図の



第 1 図

各基本素子を N 次回路でおきかえて $N+1$ 次回路を構成して，次数を順次高めていけば任意に信頼度を向上しうるように考えられる．

基本素子が状態 F または T をとる確率が0でない限り，上述の回路の次数を高めていくと，回路はほとんど常に F または T を，表現するようになることが示される．しかし基本素子が F または T をとる確率が0ならば，ある種の関数，たとえば AND, OR, 多数決などは，上述の回路の次数を高めることにより任意の信頼度で実現できる．

これらの結果が興味あるばかりでなく，これらを導く手法が厳密である点も本論文の特色である．

（戸田　巌）

88. 四重化論理

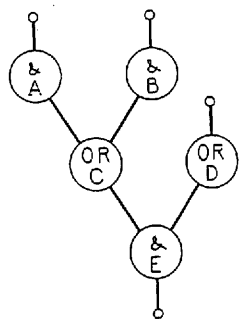
J.G. Tryon: Quadded Logic Redundancy Techniques for Computing Systems [Redundancy Techniques for Computing System, edited. by R.H. Wilcox and W.C. Mann, Spartan Books, 1962, pp. 205～228]

AND, OR, NOT を基本素子とする回路で，素子に生ずる誤りを自動的に訂正する工夫を述べている．

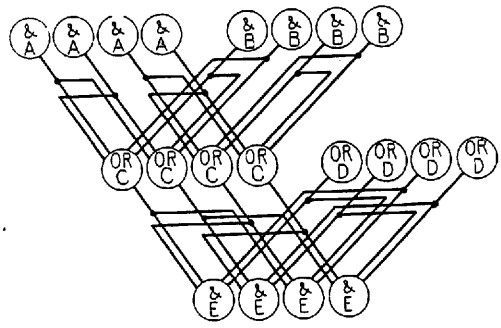
第1図の回路が与えられたとする．各基本素子を4個ずつ用いて第2図のような回路を構成する．もし AND 素子 A が，誤って出力0を与えても，次段の OR 素子で完全に訂正される．もし AND 素子 A が誤って出力1を与えるとき，次段の OR 素子では，2個の誤りとなるが次の段の AND 素子では，誤りが完全に除かれる．

このように，誤りが近接して発生しない限り，回路を四重化することにより，誤り訂正を行なうことができる．

本論文では，四重化回路によって誤り訂正を行なうための，結線法の規則を述べ，さらに，この種の工夫



第 1 図



第 2 図

により計算機を高信頼度化するときに生じうる種々の問題点について論じている. (戸田 巌)

89. Coding Theory の File Address Problem への応用

M. Hanan and E.P. Palermo: An Application of Coding Theory to a File Address Problem [IBM Jour. Res. and Devel. Vol. 7 No. 2, April 1963, pp. 127～129]

ある情報をディスク（またはディスクのようなもの）に貯えるとき，その情報の Key とディスクの記憶場所との対応は次のようになる．K_i および a_i を S シンボルのアルファベットとして $(K_1, K_2, \cdots\cdots K_k) \rightarrow (a_1, a_2, \cdots\cdots a_a)(a<k)$．したがって一つの Key に対して二つ以上のアドレスが対応することが起りうる．File Address Problem は，このことを最小にすることを目的とする．

そこで問題は Key の全体（k-tuple）を K，記憶場所の全体（a-tuple）を A としたとき，$u \neq v$ かつ $u \epsilon K$, $v \epsilon K$ なる u, v に対し $T(u)=T(v)$ は u と v との距離 $d(u, v)$ が w よりも大となるような最大距離 w と函数 $T:K \rightarrow A$ をみつけることになる．

ここで次のような仮定をする．アルファベット S は s 要素の環とする．そうすれば Key の集合 K は位数 k の S の上での free module となる．address の集合 A は位数 a の S の上での free module となる．そうすれば問題は次の性質（すなわち K の任意の元 u, v に対して $T(u)=T(v)$, $u \neq v$ ならば u と v との距離 $d(u, v)>w$）をもった A の上への module homomorphism $T:K \rightarrow A$ をみつけることである．ここで S の上の $k-a$ 次元のコード C は位数 $k-a$ の K の free submodule である．

コードの重さとは 0 でないコード語の最小の重さであり，コード語の重さとはコード語が k-tuple として表わされたときの 0 でない位置の数とする．問題は次のコーディング理論の問題と同値である．すなわち，重さ w が最大である環 S の上の位数 k の free module で $k-a$ 次元のコードをみつけよ．ここでこのコーディングの問題が，次にのべる file address の問題と同値であることを示す．すなわち，$T(u)=T(v)$, $u \neq v$ なら $d(u, v)>w$ となるような K から A の上への module homomorphism T と最大の w をみつけよ．ここに $d(u, v)$ は u と v の距離で $u-v$ の重さ $w(u-v)$ である．

最初に C が，重さ w の K の $k-a$ 次元のコードなら，$T(u)=T(v)$, $u \neq v \Rightarrow d(u, v)>w$ となるような A の上への写像 $T:K \rightarrow A$ を定義できることを示す．M を C の 0 面に対する基底のマトリックスとする．M は $a \times k$ マトリックスである．$v \epsilon K$ に対して $T(v)=M \cdot v^t$ によって $T:K \rightarrow A$ を定義する（v^t は行ベクトルの転置）．いま，もし $T(v_1)=T(v_2)$ かつ $v_1 \neq v_2$ で $d(v_1, v_2) \leq w$ なら $w(v_2-v_1) \leq w$ かつ $T(v_2-v_1)=0$ をうる．$v_2-v_1 \epsilon C$ でどちらも w より大きい重さをもつか 0 である．

$w(v_2-v_1) \leq w$ だから $v_2-v_1=0$ となるが，$v_1 \neq v_2$ なる仮定に反する．逆に $T:K \rightarrow A$ を $T(v_1)=T(v_2)$ かつ $v_1 \neq v_2 \Rightarrow d(v_1, v_2)>w$ なる module homomorphism とすると C は T の核となる．すなわち，$C=\{v \epsilon K \mid T(v)=0\}$．そのときコード C の重さは w より大きくなる．というのはもし $x \epsilon C$, $x \neq 0$, $T(x)=0=T(0)$．故に $d(x, 0)>w$．すなわち $w(x)>w$．

かくして file address の問題とコーディング理論の同値が証明された．ここで，もし環 S がガロア体ならば K も A もそれぞれ S の上での k 次元，a 次元のベクトル空間であり T は線型写像である． (水田幸夫) IBM

90. 賃借通信線の最適設置

J.E. Hosford: Optimal Allocation of Leased Communication Lines[Management Science, Vol. 9, No. 4, July 1963, p. 613～622]

多くの企業，政府機関はその事業所間の大量の通信に専用電話ラインを賃借することによって解決している．1カ月の電話ラインの賃借料はラインの本数と空中距離によるがその使用度には関係しない．いま分析の都合上二つの事業所間に敷かれた電話ラインの集合をリンクと呼び，企業体全体の電話ラインの集合をネットワークと呼ぶ．ここでは事業所間の各リンクに何本のラインを設置することが最適かを分析する．分析は次のようなステップにて行なわれる．

（1） ネットワークの i 番目のリンクにおいて，1日のうち使用量の最も多い時間に，1時間当りの平均使用時間（分）を測り u_i で表わす．

（2） i リンクが m ラインを持っているとき，100回呼んだうち待たされる回数の百分率（bussies-per-100 call）を下の式から求めそれを ${}_mB_i$ で表わす．

$$ {}_mB_i=\frac{100}{\dfrac{(1-M/60\,a)\left[1-\sum_{n=a}^{\infty}\dfrac{e^{-M/60}(M/60)^n}{n!}\right]}{\dfrac{e^{-M/60}(M/60)^a}{a!}}+1 \quad (1) $$

ただし，m ラインをもつ i リンクにおいて1時間当り平均の使用時間を μ，平均の空き時間（call と call の間の平均時間）を λ としてシステムに n 回の call がある確率は

$$ P_n=P_0(\lambda/\mu)^n/n! \quad (n<a) \qquad (2) $$

$$ {}_nP_a=p_0(\lambda/\mu)^n/a!\,a^{n-a} \quad (n\geqq a) \qquad (3) $$

(2)は $n<a$ で n 回全部通話される場合の確率で，(3)は $n\geqq a$ で n 回の call のうち $n-a$ 回待たされる場合の確率を表わしている．

$$ \sum_{n=0}^{\infty}P_n=1 \qquad (4) $$

$$ \therefore\ P_0=\frac{1}{\sum_{n=0}^{a-1}\dfrac{(\lambda/\mu)^n}{n!}+\sum_{n=a}^{\infty}\dfrac{(\lambda/\mu)^n}{a!\,a^{n-a}}} \qquad (5) $$

ここで，$M=(\lambda/\mu)60$ とおきかえると

$$ \frac{\text{bussies}}{100\text{ call}}=100\sum_{n=a}^{\infty}P_n $$

となり (1) が導かれる．

また ${}_mB_i$ は必ずしも計算によって求める必要はなく，実測によっても求められる．

（3） m ラインをもつ i 番目のリンクのコストを計算し，それを ${}_mC_i$ で表わす．

（4） それぞれのリンクにおける通信の重要性を測りそれを F_i とする．

（5） i 番目のリンクが賃借すべき最適な電話ラインの数は下の式を満足する $m*$ の最大の整数で表わされる．

$$ {}_{m*-1}B_i-{}_{m*}B_i\geqq K\frac{{}_{m*}C_i-{}_{m*-1}C_i}{F_i} $$

ここで K は常数

（6） 総計 l ラインをもつネットワークのコスト $\sum_{i=1}^{l}{}_{m*}C_i$ を計算し，このコストが高ければ K の値を増やし，反対にさらにサービスが可能なときは K の値を減じてステップ6を繰り返す．

この分析法は ${}_mB_i$ をどれだけ正確に掴めるかによってその成果が決まるが，現状においてはこの点に関して次のような制約を受けている．一つは ${}_mB_i$ が50を越えると分析の精度が悪くなること次にその回線が非常に混むと人は通信すべき要件があっても混むという先入観から電話による通信をしない場合も起り，真に現実の姿を表わしていないこともあること．しかし，この方法は簡単にリンクに設置すべき最適数のラインを決定する目安となりうる．　（森本泰生）

91. Hardware, Software および応用に共通な言語

K.E. Iverson: A Common Language for Hardware, Software, and Applications [AFIPS Conf. Proc. Vol. 22, 1962 F.J.C.C. pp. 121～129]

著者はすでに “A Programming Language” (John Wiley, New York, 1962) において，一つの言語を提供しているが，本論文ではこの言語が，hardware にも software にも共通して使われる有用な言語であることを示している．

そのため，例をあげて説明しているのであるが，それらは，命令取り出し，括弧のある式から，ない式 (Lukasiewicz) への翻訳，検索算法，記号論理，逆行列を求めること，といったもので，いずれもプログラムは 10 語程度で記述されている．

それらの中の一つ，命令取り出しのプログラムを紹介しょう．

「メモリ M（マトリックス）から，2進法表示のレジスタ s（ベクトル）の値 i=⊥s によって，語 M^i（マトリックス M の第 i 行）を取り出して，それを共通のレジスタ c に移す」

という演算は

$c \leftarrow M^{\perp s}$

と記述される．ここで，

（1） レジスタsには，2進法表示にする機能はなく，そのためには，特別なレジスタ a(s から移せる) を使わねばならない．という制約をもうけると，

$a \leftarrow s$

$c \leftarrow M^{\perp a}$

と記述される．なお，さらに，

（2） メモリとのやりとりは，レジスタbを通さねばならない．

（3） M^i から移した後では，M^i がゼロにリセットされる．つまり，$M^i \leftarrow \varepsilon$ が実行される．ここで ε はゼロベクトル．

（4） レジスタ，または語を x, y とすると，xからyへ移すには，

$y \leftarrow x \vee y$

という形を用いなければならない．

以上の (1)～(4) の制約のもとでは，

1 $a \leftarrow \varepsilon$
2 $a \leftarrow s \vee a$
3 $b \leftarrow \varepsilon$
4 $b \leftarrow M^{\perp a} \vee b$
5 $M^{\perp a} \leftarrow \varepsilon$
6 $M^{\perp a} \leftarrow b \vee M^{\perp a}$
7 $c \leftarrow \varepsilon$
8 $c \leftarrow b \vee c$

と記述される．

一般に言語が，a）初等的概念，b）それらの名称，および c）それらを運用する規則，の3部分からなるとすれば，ここで使われている言語は，上の例からみられるように，c）よりむしろ a), b) に重点をおいたものであり，名称の表現には，斬新的なものが見受けられる． （五十嵐実子）

92. FORTRAN の方言

I.C. Pyle: Dialects of FORTRAN [Comm. ACM, Vol. 6, No. 8 Aug 1963, pp. 462～467]

完全なプログラム言語というものがないため，ある言語がすっかり他の言語でおきかえられたり，あるいは経験からの批判によって変更をうけ方言を生じたりする．FORTRAN はわりあいによく生残っているほうだが，ここに四つの方言をあげる．四つというのは FORTRAN II，FORTRAN IV，英原子力局の STRETCH 用の S1 (別名 UK FORTRAN)，英原子力局（著者所属）の ATLAS FORTRAN である．S1 はIIに，ATLAS はIVにそれぞれ似ている．

IIについての批判 30 項目中実現されている項目数をしめす．

	II	S1	IV	A
除いた方がよい9項目のうち	0	6	5	6
加えた方がよい21項目のうち	0	3	10	19

この 30 項目には命名法の規則，変数・配列・関数の型の宣言，mixed mode expression，Bool 式，EQUIVALENCE & COMMON，DIMENSION & COMMON，PARAMETER & DIMENSION (load 時にきめる位数)，PUBLIC(変数名で対応づけられる common) などが論評されていて，FORTRAN の現在における発展の方向をほぼつかむことができる．

各言語のあいだの差違だけを論じているが，基本的な骨組においては FORTRAN としての特徴はあくまでも共通であること，モニタについての考察も実際上の評価には必要なこと，方言間の書きかえのためのルーチン（たとえば SIFT）が準備されていることなども触れている． （西村恕彦）

93. プロセス計算機用実行制御ルーチン

J.H. Shannon: Special Systems Div., Honeywell, Executive Control Routines for Process Computers [Control Engineering, Vol. 10, No. 4, April 1963, pp. 85～88]

プロセス制御の目的に従来は単能電子計算機を使用するのが普通であった．最近プロセスの複雑化に伴ない多方面にわたる厳密なプロセス制御が要求され中央で大形電子計算機を駆使してプロセス全体を統括的に制御する必要が増大した．これによりプロセス制御用電子計算機をどんな制御の目的で，いつ，何時間使用するかを自動的に割り当てる実行制御ルーチンが必要になった．

プロセス制御用プログラムは前もって磁気ドラムの中に記憶してあり，実行制御ルーチンがプロセスの制御を要求する時点で磁気ドラムから制御用プログラムをコアの中に読み込みプロセスの制御をさせる．

したがって実行制御ルーチンの持つべき機能として

（1） データの走査ができること

プロセスの制御に必要なデータを走査する．かつ制御の緊急度に従って別々の走査径路を持たせて制御データの走査に必要な時間をできるだけ短くする．

（2） データの記録をする

単位時間ごとにデータの記録をする．たとえば1時間単位，8時間単位にデータを記録する．さらに日単位でサマリーを記録する．

（3） プロセスに必要な計算の処理ができる．

原料の配合および熱平衡状態の計算，プロセス効率，在庫原料の管理計算を行なう．

（4） 入出力制御の判断

制御用プログラムの実行に必要な入出力を制御用プログラムの優先度を考慮しながらバッファーを使用して有効に行なう．

（5） オペレーターと電子計算機間の情報交換

緊急時にオペレーターがキーボードを使って電子計算機に緊急なデーターの入出力を行なわせることができる．

実行制御のタイミングの決定法

まず，プロセス制御に必要な実行時間の間隔 T_i を決定する．次に一つのプログラムが実行しおわった時間を T_l，実行制御ルーチンがプログラムを実行させるかどうか判断する時点を T_n とする．$T_n-T_l \leqq T_i$ ならばプログラムをドラムからコアに読み込み制御用プログラムを実行させる，$T_n-T_l > T_i$ ならばプログラムを実行させず次に移る．実行制御ルーチンは制御用プログラム全体についてそれぞれの T_l, T_i の表を検討し，制御用プログラムをタイムリーに実行させていく．実際に T_n としては1日を周期としてリセットされる時間を使用する．あるプログラムの実行時間が長びき，入出力の遅延により全体のプログラムの実行が順に遅れるのを防ぐために，1日のうちに適当な調整時点を設け，プログラムの実行時間の遅れなどをリセットする．

他にアナログ計算機を使用した場合に制御が容易なプロセスについてはアナログ計算機の併用をすすめている．

（半田　福）

94. 問題向記号処理計算機——ADAM

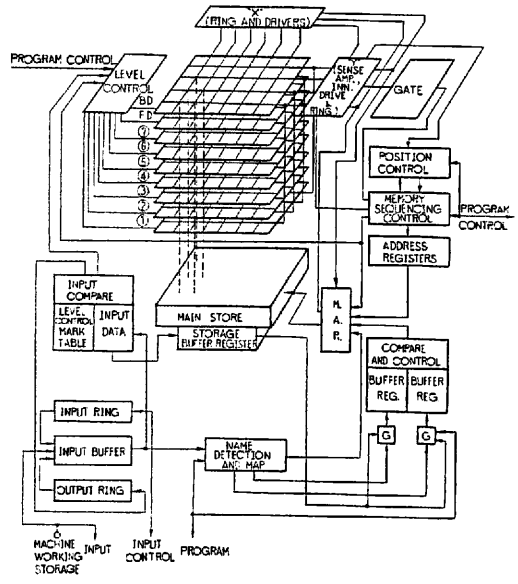
A.P. Mullery, R.F. Schauer and R. Rice: ADAM-A problem oriented symbol processor [AFIPS Proc. Vol. 23, 1963 S.J.C.C., pp. 367～380]

記号処理の問題を掘り下げて，根本的に新しい実験計算機を作ろうとする野心的試みがここにある．

長さ，形式，構造が可変であって，必ずしも整列していないデータを処理しなければならないとなれば，これを制限するより逆にとらえなおし，積極的に，

1. 可変長データだけでなく可変長命令の扱い，
2. データ構造を内部的に具現し，これに操作を加えること．
3. データの記号指定
4. データのつなぎ（link）の解釈，挿入．

が可能となる言語と計算組織を考える．データを造り出し，移動し，消去するような基本命令のほかに，新しい命令を簡単かつ柔軟に定義できること，プログラ



第 1 図

ムによる家事きりまわしを最小限に——たとえば記憶場所の自動割付などを——するには，記憶装置，処理機構，入出力などに新しい観念を必要とする．などと力むが何のことはなく，重層記憶を用いて，金物で言語を解釈実行するものにほかならない．ただおもしろいのは，データの扱いであって，(文字)記号，句，文，節，章，巻，文庫のレベルを示す符号①②……⑦（データに付すものと指定に用いるものでは異ならざるをえない）およびデータのつなぎを示す符号⑧を金物で設けることだ．例をみょう．

nAn ④ Brown ① is ① a ⑧ Des ① school ① with ① many ⑧ Des ① students ④ nDes ② very ① good ②

nは名称の示す符号である．Des という名の very good という句（②で示される）が2度用いられている．

Brown is a very good school with many very good students の意である．

このようなデータは機械語に分断されて入るが，新しいレベルが表われると必ず新しい機械語に配置される．このようにしておけば各機械語に対応して，そのレベルのはじまりを示すことは数ビットで足る．物理的には第1図の如きものとなる．語単位計算機における部分語指定は，ここではある文の，あるレベルのもつ幾番目かのものを指定するものとなる．先の例では

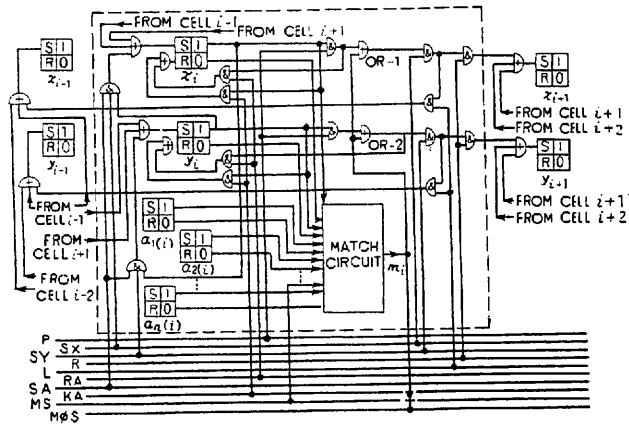
A②↑0①↑1

は文 A の第1句の第2記号，すなわち，good を指定する．データのつなぎは，よく知られている chain link にほかならない．また入出力については図から推測されるところのものである．（丸山 武）

95. 内容により索引可能な論理を備えた記憶方式

C.Y. Lee and M.C. Paull: A Content Addressable Distributed Logic Memory with Applications to Information Retrieval [Proc. IREE. Vol. 51 No. 6, June 1963, pp. 924～932]

計算機回路が高速化されるに従って，配線長の制約からのがれるために，論理機能を備えた記憶方式が考えられている．この論文はこうした記憶方式の一例を提示したもので，その特徴は



第 1 図

1. 内容により索引可能である．
2. 記憶される情報列の長さについての拘束がない．
3. 情報は記憶容量に無関係な速度で取り出される．
4. 記憶素子は単一化，小形化されている．

基本となる記憶回路は，セルと呼ばれる順序回路よりなる．これは図のような構成で，セルの状態をあらわすフリップフロップ X_i, Y_i と，情報を保持するフリップフロップ $a_{1(i)}, a_{2(i)}, \cdots\cdots, a_{n(i)}$ を持つ．この制御は外部から送られる 10 本の制御線でおこなわれ，セット，リセットの外に記憶内容の伝播，内容の一致検出をおこなう．これらの制御はすべてのセルに並列におこなわれるので，情報の索引による取り出しなどの動作は速く，しかも記憶容量によらない．

記憶に関する指令は，外部の制御部分でプログラミングにより作られる．指令は，基本的には記憶 (STORE) 情報取出 (RETRIEVE) 転位 (PROPAGATE) 一致照合 (MATCH) の4種のパターンの組み合わせからなり，16 個の指令を持つ．各命令は4個までの補助変数をとることができ，$\alpha, \beta, \gamma, \lambda$ と名付けられる．α は転位の方向を示し，L(左) または R(右) と書かれる．β は X_i および Y_i の状態を指定し，X,Y であらわす．γ は主に一致照合の際セルの内容を指定するのに用い，(A, X=1, Y=0) などと書かれる．λ はアドレス指定のために用い，直接記憶番地を指定するので，セルの動作には関係しない．たとえば MKA (Match Propagate and Kill Activity) という指令は，α, β, γ, の補助変数をとり，MKAL×(S, X=1, Y=0) とあれば，もしセルの内容が (S, X=1, Y=0) であるものがあればそのXの内容を左隣りのセルへ移し，同時に他のセルの X, Y の内容を全部0にリセットすることを意味する．

これらの指令から，索引の際キーを読出して項目全体を取り出す動作，取り出した後のセルの内容を消して新しい項目で埋める動作などを効果的におこなうことができる．ただ各セルに要する素子の数が極めて多いという欠点があり，また照合によって一致するセルが2個以上ある場合の処理などに今後の問題を残している．（加藤隆二）