I.F.I.P. 論文紹介*

46. 帰納的推論を行なうオートマトンについて

L.J. Fogel: Toward Inductive Inference Automata [XI-3, pp. 172～176]

オートマトンは逆境の中に置かれていると考える．そうするとその中でオートマトンが生き残ってゆけるかどうかは，環境の厳しさ，オートマトン自体の知能程度，諸種の経験の中から有害な影響を克服して行く能力に関係することとなる．

まずオートマトンの知能水準は，ある有限時間内に残存できる確率で定義されるものとする．知能程度が低いときはオートマトンはほとんど完全に受動的であるが，知能が高くなると自らが置かれている環境に感じ，かつすでに経験したことのあるこれと類似の効果を思い起すことができるようになる．これはオートマトンのパターン認識能力と考えられる．

しかしさらに高い生存確率は，現在置かれている環境を認識しようとする前に，経験上有害であると思われる効果を事前に避ける能力を持つことによって，達成されることになる．この能力は未来の環境に対する予測能力であると考えられ，現実に行なわれている科学的方法は，丁度これに該当している．すなわち現実の世界から取り入れた知識を帰納的推論によって一般化し，これをもとにしてモデルを作り，それから演繹的推論によって解を求め，それを特殊化して現実のものを再現させるという方法である．これによって現実のより深い認識と未来への予測が可能となる．

以上の考えを具体化する一つのモデルを想定して，これについて議論がなされた．まず記号によって表現されるオートマトンの内言語を考える．これは環境に対する知覚量から翻訳されるものであるが，翻訳のルールは過去の経験によって得た各種の知覚量のマルコフ過程に関する条件確率によって定められ，経験を重ねて行くに従って連続的に変化して行くものと仮定する．新らしい経験をより重視するようにすれば，頻繁に現われる知覚量に対する言語は永く存続するが，そうでない言語は寿命が短いことになる．また連続的な変化を破るような新らしい知覚量が受入れられた時は，有害な情報をもたらすとして阻止すべきものと判断する．言語を表現する記号の種類の総数を k とし，t なる時刻に i 番目の記号の出現する確率を $P_i(t)$ とすれば，時刻 t における相対情報量 $H(t)$ は

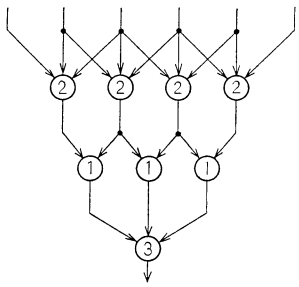
$$H(t)=\frac{\sum_{i=1}^{k} P_i(t)\log_2 P_i(t)}{\log_2 1/k}$$

で与えられる．これはオートマトンの健康状態を示す測度と考えられる．このような方法によって帰納的な推論を行なうオートマトンが，モデル的に実現されることを示し，検討を行なった．　（飯島泰蔵）

47. Threshold Element の Digital Filter

G. Hotz: Digital Filters of Threshold Elements [XX-2, pp. 337～340]

$S_n^k(n\geq k)$ を input n で，threshold k の threshold element とする．すなわち，n 本の input のうち k 本以上が 1 のとき，1 を output する single output element である．こういう threshold element をくみあわせて symetrical digital filter とよばれるswitching circuit をくみたてる．この digital filter を略して DF とよぶが，1-stage の DF は threshold element そのものである．$r>1$ なる r stage の DF は $(r-1)$ stage の DF からつぎのようにつくられる．$F_1, F_2, \cdots\cdots, F_n$ を pairwise に，等しい $(r-1)$ stage の DF とし，それぞれの input を $x_{i1}, x_{i2}, \cdots\cdots, x_{im}$ $(i=1, 2, \cdots\cdots, n)$ とする．その output が S_n^k の input になる．また F_i の input x_{ij} は，$i+j-1=t$ の x_t' とする．これで r-stage の DF ができた．r-stage の DF は，i-1 stage から i-stage の DF を作る threshold element を (n_i, k_i) とかくと



* 前号に引き続き第2回国際情報処理学会の提出論文（本誌 Vol. 3 No. 4 参照）を Preprint of the Proceedings of the IFIP CONGRESS 62 (Aug. 1962) から紹介します．

(n_1, k_1), (n_2, k_2) ……, (n_r, k_r)

であらわされる. (3,2), (2,1), (3,3) はつぎの回路(前ページ)となる.

X を $x=$……, x_{-2}, x_{-1}, x_0, x_1, x_2, ……, $x_i=0$ or 1 for all $i\epsilon G$ という sequence の集合とする. G は integer である. $x\epsilon X$ と $y\epsilon Y$ は $x_i=y_i$ for all $i\epsilon G$ のとき $x=y$ とする.

x, y に対して $z=$……, z_{-1}, z_0, z_1, …… をかんがえ

$z=x\vee y$, when $z_i=\max(x_i, y_i)$

$z=xy$, when $z_i=\min(x_i, y_j)$

$z=x'$ when $z_i=0$ if $x_i=1$

$z_i=1$ if $x_i=0$

$x\subset y$, when $x_i\leq y_i$

$z=x+y$, when $z_i=x_i+y_i$ (mod 2)

とする.

$x\epsilon X$ で $x_{i-1}=0$, $x_i=1$ なる x_i を left marginal point といい, $x_i=1$, $x_{i+1}=0$ なる x_i を right marginal point という. x_i, x_k を left と right の marginal point し, そのあいだの x_j が1ならその部分は $k_{-i}+1$ のながさの 1-block をつくる. 一つでも 1-block があれば, その最小のもののながさを $l(x)$ であらわす, また最大のものを $L(x)$ であらわす. この block が一つしかないとき, そのながさを $N(x)$ であらわす.

$X_k=\{x\epsilon X \mid l(x)\geq k\}$,

$x\epsilon X_k$, $y\epsilon X_k$ なら $x\vee y\epsilon X_k$

$Y_k=\{x\epsilon \times \mid L(x)\leq k\}$,

$x\epsilon Y_k$, $y\epsilon Y_k$ なら, $xy\epsilon Y_k$

$I_k=\{x\epsilon X \mid N(x)\leq k\}$ のとき

$x\epsilon I_k$, $y\epsilon I_k$ なら $x\vee y\epsilon I_k$ また

$x\epsilon I_k$ $y\subset x$ なら $y\epsilon I_k$

ある DF, F に関する関数を $f(x_1, \cdots\cdots, x_m)$ とかく,

$y_i=f(x_i, x_{i+1}, \cdots\cdots, x_{i+m-1})$

できまる $y=$……, y_{-1}, y_0, y_1, …… も $y\epsilon X$ である. これを $y=Fx$ とかく.

別の関数 $y=Tx$ は $y_i=x_{i+1}$ とすると

$F(Tx)=T(Fx)$

となる.

二つの DF, F, F' があったとき, 積 $F\cdot F'$ を

$(F\cdot F')x=F(F'x)$

と定義する. そのとき (n_1, k_1), (n_2, k_2), ……, (n_r, k_r) の DF による関数 F は, 積のかたちで

$F=F_{n_1,k_1}\cdot F_{n_2,k_2}\cdot\cdots\cdots F_{n_r,k_r}$

とかけて, F_{n_i,k_i} は (n_i, k_i) の 1−stage DF である.　　(和田英一)

48. 一つまたは一つ以上の Threshold をもつ Threshold Logic

P. Ercoli and L. Mercurio: Threshold Logic with one or more than one Threshold 〔XX-3, pp. 341～345〕

Linear threshold element は, n 個の binary input $x_i(i=1, 2, \cdots\cdots, n)$ から, level l に対して binary single-value function $f(l)$ を output する element である. ただし l は

$$l=\sum_{i=1}^{n} w_i x_i=\boldsymbol{WX}$$

で, $\boldsymbol{W}=w_i(i=1, \cdots\cdots, n)$ は weight である.

Threshold s_1, s_2, ……, s_m $(s_1>s_2>\cdots\cdots>s_m)$は l がふえていくと, value が変化するところの l の値である. ふつうの threshold は $m=1$ をあつかっている. Type t は $l\geq s_1$ のときの output の value であって,ふつうのresistor-transistor の one-threshold element では $t=0$ また Goto pair では $t=1$ である. element は weight と threshold と type とできまるから, それを

$T^t s_1, \cdots\cdots, s_m[w_1x_1, \cdots\cdots, w_nx_n]$ または

$T^t s_1, \cdots\cdots, s_m[\boldsymbol{WX}]$

とかく.

さて, ある binary function $f(x_1, \cdots\cdots, x_n)$ は, 適当な linear threshold element で実現できる. これで実現できたときの, うえのような表現を threshold form とよぶ. 問題はこのような form であたえられた function を, threshold element で実現することである.

(訳注　あたえられた function の threshold の数と使用できる element の threshold の数, type がちがう場合をあつかう.)

実現の方法は2段階にわけられ, conversion では $f(x_1, \cdots\cdots, x_n)$ からうまい resolving vector $\boldsymbol{W}$ をみつけること, これをできるだけ“convert”してうまい $\boldsymbol{W}$ にちかづける. 第2に synthesis では, conversion がいきついたところで, 手持ちの element の組み合わせで実現できるように接続をかんがえる.

(訳注　ここであたえているいくつかの conversion の規則は, ふつうの conversion では, threshold がひとつの場合の equivalence を論じているのに対し, multi-threshold の場合の equivalence を論じている.)

一方 synthesis に必要な公式はつぎのとうりである．$t'=1-t$ として

$T^t s_1, \cdots\cdots, s_m[\boldsymbol{WX}]=T^0 t[(-1)^{t'}T^0 s_1, \cdots\cdots, s_{h1}[\boldsymbol{WX}]$;

$(-1)^{(h_1+t')}T^0 s_{h_1+1}, \cdots\cdots, s_{h_2}[\boldsymbol{WX}]$;

$(-1)^{(h_2+t')}T^0 s_{h_2+1}, \cdots\cdots, s_{h_3}[\boldsymbol{WX}]$;

$\cdots\cdots$

$(-1)^{(h_r+t')}T^0 s_{hr+1}, \cdots\cdots, s_m[\boldsymbol{WX}]]$

この意味するところは，おなじ resolving vector $\boldsymbol{W}$ をもつ $r+1$ 個の type 0 の element，しかもそのそれぞれがちがった threshold をもつような element をもってきて，output を 1 threshold element につなぐと左辺の operation が実現できるということである．この $r+1$ 個の element を first-stage element，それから input される 1-threshold element を second stage element とよぶ．この方法では，weight $\boldsymbol{W}$ がみなおなじで，この関係を isobaric という．一方 $\boldsymbol{W}$ をかえることもできる．それは

$T^{t_1}s_{11}, s_{12}, \cdots\cdots, s_{1m}[\boldsymbol{W}_1\boldsymbol{X}]=T^{t_2}s_{21}, \cdots\cdots, s_{2r}[\boldsymbol{W}_2\boldsymbol{X}]$

の equivalence をつかうものである．l のとりうる範囲 l_M, $l_m(l_M>l_m)$ のうち，$l_M\geq l\geq l_m+p$ なる l が $f(l)=f(l-p)$ なら f は周期 p について periodic であるという．このとき

$w_{21}=w_{11}-p$, $w_{2i}=w_{1i}$ $(i=2, 3, \cdots\cdots, n)$

とすることができる．また recursive formula から $s_{21}, \cdots\cdots, s_{2r}$ がきまるが，このうちいくつかは $s_{11}, \cdots\cdots, s_{1m}$ と一致している．$w_{11}w_{21}\geq 0$ または $w_{11}w_{21}<0$ でかつ $s_{11}, \cdots\cdots, s_{1m}$ のうちで $s_{21}, \cdots\cdots, s_{2r}$ と一致しないものの数が偶数なら $t_2=t_1$ であり，そうでなければ，$t_2=1-t_1$ である．

multi-threshold element は tunnel-diode をつかって実現できる．この diode 一つは 3-threshold だが，一般にN型の電圧電流特性をもつm個の element からは，直列につないで $2m+1$ threshold がえられる．（和田英一）

49. Threshold Logic で Boolean Function をかぞえるのに有用な定理

E. Goto and H. Takahashi: Some Theorems Useful in Threshold Logic for Enumerating Boolean Functions〔XX-4, pp. 346～350〕

Threshold logic の分野では，Boolean switching function を分類するのに，SD(self dual) にしておこなうのがよい．それは，このおなじ分類に属するものは，おなじ回路で実現できるというようなことがあるからであり，また PN (permutation negation) や NPN (function の negation, variable の permutation, negation) などの分類に比し，種類がすくなくなり，表をつくったり，種類をかぞえるのに便利だからでもある．一つの threshold device で実現できる n variable までの Boolean function の種類 $N(n)$ がどのくらいあるかということも Threshold logic の興味ある問題である．

n variable x_i の Boolean function $b(x_i)$ $(i=1, \cdots\cdots, n)$ から

$B(x)=x_0 b(x_i)+\bar{x}_0\overline{b(x_i)}$

でつくった $n+1$ variable $x_i(i=0, 1, \cdots\cdots, n)$ の Boolean function は b の self-dualized だといい，また $B(x_i)$ から

$b(x_1, \cdots\cdots, x_n)=B(x_0=1, x_1, \cdots\cdots, x_n)$

でつくった $b(x_i)$ を B の anti-self-dualized だというとき

1. self-dual function の anti-self-dualized の self-dualized も，non-self-dual function の self-dualized の anti-self-dualized も，もとのものであり，

2. self-dualized B と anti-self-dualized b のあいだの対応は 1：1 であり，

3. function $b_1, \cdots\cdots, b_m$ の function h の self-dualized は self-dualized function $B_1, \cdots\cdots, B_m$ の self-dualized function H にひとしい，という定理がみちびかれる．

ところで Boolean function b_1 と b_2 とがおなじ SD class に属するのは，b_1 を self-dualize するか anti-self-dualized するか，function を negate するか，variable を permute するか negate するかして b_2 と一致することである．

self-dualize と anti-self-dualize で function の種類はぐっとへり，4 variable までの general function は PN で 402, NPN で 222 もあったのが SD は 83 しかない．

n variable x_i の Boolean function f が W_i と T をつかって

$f=D(\Sigma W_i x_i-T)$

とあらわされるとき f を linear input function という．D は non linear unit step function である．$b(x_i)$ は k 個の linear input function f_r をつ

かって $f_{rx}=f_r(x_i, f_{1x}, \cdots\cdots f_{(r-1)x})$ の操作により，最後に

$$b(x_i)=f_{kx}$$

と実現することができる．minimization ではこの k をちいさくしなければならない．この数は step function の数 N_D と必ずしもひとしくない．それは f_r に linear function I をつかうことのできるところもあるからで，その数を N_I とすれば

$$k=N_D+N_I$$

となり，これらの数は，SD class のなかで保存される．

$N(n)$ の下限は $2^{0.25n^2}$ である．それをもとめるには，まず $2m$ variable $x_1, \cdots\cdots, x_{2m}$ に W_i, W_j の weight をかけ $1\leq i\leq m$ について $W_i=2^{i-1}$, $m+1\leq j\leq 2m$ について $1\leq W_j\leq 2^m$ とする．さらに

$$f=D\left(\sum_1^m W_i x_i+\sum_{m+1}^{2m} W_j x_j-2^m\right)$$

の linear input function をかんがえる．W_j をふたくみ用意し，$j=k$ だけでことなっていてそこでは $W_k{}^{(1)}>W_k{}^{(2)}\geq 1$ とする．x_k を1とし，$j\neq k$ については x_j を0とする．

$$2^m-2\geq\sum_1^m W_i x_i=2^m-W_k{}^{(1)}>0$$

と $W_k{}^{(1)}$ をきめる．x_i がどういう構成かで $W_k{}^{(1)}$ がかわってくる．そうすると $f^{(1)}=1$, $f^{(2)}=0$ となる．こういうふうに $f^{(1)}=1, f^{(2)}=0$ となる W_j の数は $(2^m)^m$ 個ある．x_j はまた non-idle だから，これを negate すれば別の linear input function がえられることを考慮して，結局 n が偶数なら $n=2m$ とおき

$$N(n)>2^m(2^m)^m>2^{0.25n^2}$$

である．奇数の n についても，同様な議論でこの下限がでてくる．　　（和田英一）

50. SYNTOL（文法構造をもった言語）

J.C. Gardin: Le SYNTOL (Syntagmatic Organization Language)〔Ⅷ-2, pp. 171～120〕

SYNTOL は，フランス国立科学研究センタの資料自動蒐集整備課と高等実践学校の記号・資料整備研究部とにおいて開発された，電子計算機による自動情報検索システムである．このシステムは，特別の分野を対象としておらず，IBM 7090 を用いた実験では，心理学，社会学，人類学を対象として扱っている．

SYNTOL システムでは，情報は一連の，二つの基語よりなる句（Syntagm）により構成される．ここで用いられる基語は，日常言語と同じように，述語，主体語，状態を表わす語，動作を表わす語の4種に分類され，同時に，この分類によって，いずれの二つの基語をとっても，その間に4種の関係のいずれかが成り立つようになっている．四つの関係とは，同格関係，因果関係，補足関係，述語的関係である．このような基語の分類によって，各句（Syntagm）の意味はほとんど一意性を保つようになるが，それでもなお意味が曖昧な場合には，句を構成する基語に，その文法的性格を示すオペレータを付加することができる．

情報が与えられると，一連の二つの基語からなる句（Syntagm）と，各 Syntagm のもつ関係とに変換され記憶される．一方，質問も同じ形式に変換されて検索が行なわれるのであるが，質問の場合には，その変換が単なる基語の羅列にすぎないものから，オペレータ付の文章にまで行なうものがあり，その結果でてくる解答によって記憶されている情報が効果的に解析されているかどうかを試すことができる．

質問の方法により，出てくる答を狭い範囲に限定したり，その反対に，なるべく広い範囲にわたる一般性のあるものにしたり，あるいは質問を部分的にいろいろかえてみて答を引き出すこともできる．

このシステムは現在もなお実験が行なわれており，同時に，日常言語で記されている論文の梗概を，SYNTOL 語に飜訳する方法の研究，できるだけ多くの情報を有効に記憶させておく方法の研究などが続行されているという．　　（関本年彦）

51. 情報の分類に計算機を使用する方法

R.M. Needham: A Method for Using Computer in Information Classification〔Ⅷ-3 pp. 121～123〕

情報の分類においては，項目間の類似したものを一つの group にするが，その類似を表わす何か数量的なものがほしい．それで，次の二つを問題として，話をすすめている．

1. 数量的で，類似性を表わすものを定義する．
2. グループを得るための algorithm を見つける．

まず1に対して，項 i と j の間の connection c_{ij} を

$$\frac{\text{項 } i, j \text{ の両方を含む document の数}}{\text{項 } i, j \text{ のいずれか一方を含む document の数}}$$

で定義し，これらを element とする対称行列 connection matrix A を作る．また，項の subset T' に

対して，項 i が T' に含まれているかどうかで，u_i' を $+1$，-1 とする vector u' を作る．

次に2に対しては，

$AV=QV$ (Q:non-negative diagonal matrix)

なる V によって定まる subset T_V を clump と定義する．

また cohesion を次で定義する．

$$\frac{1'A1-V'AV}{1'A1+V'AV} \quad 1=(1,1,\cdots\cdots,1)$$

ここで，clump は cohesion が極小のときであるということから，algorithm は cohesion を減すように進む，(V の初期値は任意)

最初 V の初期値を U として，

$AU=RU$, R: 対角線行列

ここで，R の元に負があれば U の対応する元の符号を換えて行く．

次に Connection matrix の計算．clumpfinding の program 上のことが述べられ，さらに，実験報告がされている．そこでは，計算時間のほとんどが Tape の wind および rewind の時間にかかっている．また結果の評価において，演繹的なものには弱いということである．

さらに，これらの他方面（医学，言語学，人類学）への応用も述べられている．そして，scale が大きい場合の問題として，次数Nよりも小なる次数で，計算できるような改良案を示している．（菅野良夫）

52. 日常の言語を使用して計算機に照会させること

D.R. Swanson: Interrogating a Computer in Natural Language〔Ⅷ-4, pp. 124～127〕

text searching についての以前の経験をもとにして，それを拡張したものを述べている．

まず，text 中の information の search の形式化を計算機にさせることとして

（1） 計算機は単語およびその一連の単語に作用する．

（2） “辞書”によって同義語も考慮され，さらにその辞書に適当な文法 Code を入れて word 間の文章論的な関係も指定できる．

（3） 日常言語を search 命令で計算機が変換できる．

ならば，indexing，必要な形式化，retrieval などの全過程が自動的になるということで，この日常言語の問題点は拡大した text searching の問題を根本とみなせる．そこで，この仕事の目的はそのプロセスの根本的なものを考えることにしている．

それで，それについての問題点として，information を relevant なものと，irrelevant なものとの区別ができるような類似性の問題と統語法的問題を挙げている．この点に関して，同義語をもった“辞書”をそなえ，さらに，retrieval の目的に応じて，word の重要性を識別できる能力を機械にそなえるなどが述べられている．質問に relevant なもの全体およびそれのみをさがすことは，不可能ではあるけれども，irrelevant なものを最大にする information system を設計することを目標としている．

初期には，“辞書”は search の形式化を助ける意味で，同義語の list の役をはたした．その後，人間のやる search の研究を基礎として，より正確なものを，より少なく選ぶようになされた．そこで，phrase としては，意味のある word の対に限定された．また，logical な問題の search はかなりの難問を引き起こしたが，項目間の logical な関連を指定する方法の形式化の代りに，search される document に対して，項の数およびその重さによって定まる relevence score というものを対応させることにした．

その実験結果として，次のことが述べてある．

（1） search が形式化されたときに，全部機械によって retrieve された relevent な情報の割合は，人間が形式化されたとおりにやるよりも高いパーセンテージを示した．

（2） irrelevant な retrieve がいくらかでも，認められれば，それ以上に本質的に relevant な retrive は可能であった．

“辞書”によって，日常言語と document を matching することの方が，人間が subject index term に適当に対応させるよりも正確のようである．

（菅野良夫）

53. 多重路を用いた構文の分析法

S. Kuno and A.G. Oettinger: Multiple-path syntactic analyser〔Ⅸ-1, pp. 128～133〕

文章をただ1回読む間，すなわち1語読んでは考え，次にはどんな語が出るだろうかを“予測”しながら進めば，ピリオドまで来ると直ちに文の意味がわかるはずで，人間が文を理解する時の思考状況に似ている．この方針に則ったプログラムで機械飜訳をやらせ

Table 1. The Subrules of G-(Sentence, PRN).

g_1 (SENTENCE, PRN)	=|PERIOD|PREDICATE
g_2	=|PERIOD|PREDICATE|ADJECTIVE CLAUSE
g_3	=|PERIOD|PREDICATE|COMMA|ADJECTIVE PHRASE|COMMA
g_4	=|PERIOD|PREDICATE|COMMA|SUBJECT PHRASE|COMMA
g_5	=|PERIOD|PREDICATE|SUBJECT PHRASE|AND-OR
g_6	=|SENTENCE|COMMA|PARTICIPLE
g_7	=|SENTENCE|COMMA|PARTICIPLE|SUBJECT PHRASE|AND-OR
g_8	=|SENTENCE|COMMA|PARTICIPLE|COMMA|SUBJECT PHRASE|COMMA

ることを predictive analysis の法という．著者はさきにこの実験を行なったが，いくつかの解釈の考えられる場合に苦しんだ．そこで前回 pool と呼んでいた記憶装置を，たくさんの subpool に分割して，前回の欠点を補うべく実験を試みた．この論文はその実験結果を詳しく記述したものである．

まず単語字引には syntactic word class を含めてある．文法としては，その時点までの情報から，引続いて起り得る全てのパターンを用意しておく．

They are flying planes を例にとると，They (PRN), are (Be 1, Be 2, Be 3), flying (RI 1, RT 1 GI 1, GT 1), planes (NOU, VI 1, VT 1), (PRD) とコードされている．

まず SENTENCE と予想して，they を調べると，この範囲から第1表に示すように文法 G(Sentence, PRN) には，8個の sub-rule が出て来る．ここで右に記入されているものが prediction であって，大文字は文章の場合と逆に，右側のものから読むように記されている．なお g の次にあるものを著者は syntactic role indicator と名ずけている．

次に単語 are を読み込むと，class Be 1, Be 2, Be 3 が出る．それぞれを8個の subrule と内容と比べると，8×3 個の場合の可能性があるが，文法表によって，わずか3組の G(PREDICATE, Be)しか残らない．この各々にいくつかの subrule があるが，flying を調べることによって前と同じように減ずるが，文章にいくつかの解釈がある場合は，それらが全て打ち出される．

〔紹介者註〕 predictive にすることの意味は判るが，この方法のために用意しなければならない文法に関する尨大な努力を考えると，その長所を指摘することは難かしいようだ．

（和田　弘）

54. Malgrammars と言語における公理系

K. Čulik: Some Axiomatic Systems for Malgrammars and Languages〔IX-2, pp. 134-137〕

本論文は，歴史的に言えば，言語における N. Chomsky の“文法”を semi Thue system における semi Thue rules を使って表現しなおしたものと言えるだろう．

内容は，3節から成り，第1節では，“文法”を言いかえたものを“semi-algorithm”と名ずけ，定義している．“semi-algorithm”とは言語から言語への変換で，おきかえ則と選択条件で決められるものである．いま，$A \neq \phi$ を半自由群 A^∞ の生成元の集合とすると，言語 M, N,……は A^∞ の部分集合である．おきかえ則と選択条件には次のようなものを考える．

おきかえ則：$\mathfrak{R} \subset A^\infty \times A^\infty$ を与えると $\rho=(x, y) \in \mathfrak{R}$ に対して $\bar{\rho}=\{(u, v) \mid u=z_0 x z_1,\ v=z_0 y z_1;\ z_i \in A^\infty,\ i=0, 1\}$ とし，これをおきかえ則（つまり，semi Thue rules）とする．その集合を $\bar{\mathfrak{R}}$ とする，$\bar{\mathfrak{R}}=\bigcup_{\rho \in \mathfrak{R}} \bar{\rho}$.

選択条件：$B \neq \phi$, $B \subset A$ と $\mathfrak{P} \subset \mathfrak{R}$ を与えると，$C_B=B^\infty$, $C\mathfrak{P}=\{(x \epsilon \omega^\infty \mid (x, y) \in \mathfrak{P}$ となるような $y \epsilon \omega^\infty$ がない．$\}$ の二つの言語が定義出来，これらを選択条件とする．

さて，“semi-algorithm”とは，$S=\langle A, \mathfrak{R};\ B, \mathfrak{P}\rangle$ を指し，$M \subset A^\infty$ なら，$S(M)=C_B \cap C\mathfrak{P} \cap Q(M)$……(1)；$Q(M)=\{x \mid 1 \leq i < k,\ 2 \leq k$ に対して，$(w_i,\ w_{i+1}) \in \bar{\mathfrak{R}}$ となるような $w_i \epsilon \omega^\infty$ が存在する．$w_1 \epsilon M, w_K=x\}$.

第2節では“semi-algorithm”によって，N. Chomsky の“文法”が表現できて，逆も成り立つ事を証明している．“文法” G は terminal と non-terminal vacabularies V_T, V_N と $V=V_T \cup V_N$ で定義されている関係→とで，決定される．関係→は四つの公理で運用される．“文法” G によって生成される言語 L_G は $\{x \epsilon V_T \mid 1 \leqq i < k$ に対しては，$w_i \rightarrow w_{i+1}$ となるような $w_i \epsilon V^\infty$ がある．ここで $w_1=\#S\#$, $w_K=x$, 亦，$x \rightarrow y$ となるような $y \epsilon V^\infty$ がない．$\}$ と定義できる．

一方，次の条件：1) A, $\mathfrak{R}=\mathfrak{P}$ を有限集合とする．

2) $B \ni \#$

3) $\mathfrak{R}$ の元を (p, q) とすれば，p, q は # を含まないが，含むとすれば片端にしか含まない．

を充たす“semi-algorithm”を S_1 とし，両端にのみ#がある元から成る集合を M とすると，

$$L_G = S_1(M)$$

を示すことがここでの目的となる．

第3節では，Parker-Rhodes の“A New Model of Syntactic Description”(Nat. Phs. Lab. of Teddington, 1961) をもとにして，前節同様“semi-algorithm”を使っていい直しているのであるが，その結果，二つの lemmas が成立しないことをそれぞれ反例をあげることで証明している．

〔紹介者註〕ミスプリントのため（1）式は推量して出したものである．なお本文第2節では句構造文法となっているが，内容からただの“文法”のことであると判断した．

（五十嵐実子）

55. 文章の解釈法

M. Kay: Rules of Interpretation, An approach to the problem of computation in the semantics of language〔IX-3, pp. 138～141〕

言語についての文法は従来から句などの構成について記述したものが主で，一般的に解釈法を教えるものではない．ここで解釈法とは，句なり文章などの多数の単語から成り立っているものの意味をその成分の意味から誘導する指針を指している．

snow man（雪だるま），ice man（氷を運ぶ人）の例に見るように，修飾語にさえ統一された解釈はできない．

そこで，この修飾の関係を計算によって調べる方法を提案する．ここでは便宜上，英語で名詞と形容詞に限り，単語と句との関係を調べる．単語を a, b ……とし，a が b を修飾することを ab で表わすとする．なお，次の制限を加える．

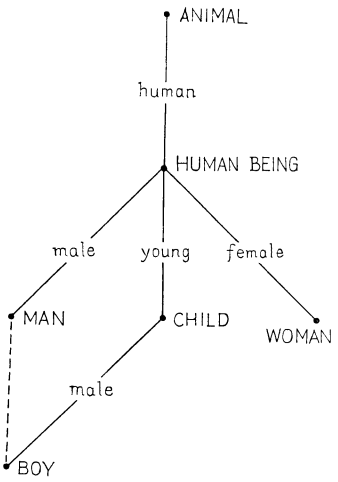
$aa=a$, $ab=ba$ なら $a=b$,

$(ab)\cdot(bc) \to ac$ は不成立，$a(bc)=b(ac)$

いま young human=child, male child=boy, male human=man なる三式を与えれば，上記諸式から young man=boy を導くことができる．

このようにうまく行くものが多ければ規則にしたい．残りのものについては，後日その規則を考えてもよかろう．それにしても，これを実験してみるのが大変な作業になる．モデルを考える必要が起った．

第1図について説明はいるまい．この net は Cambridge LRC の lattice に似たものとなった．



第1図

機械翻訳の研究をすると，一国語の訳が目標とする国語の単語で相当するものがなくて，句で表わされねばならぬことがある．まだ実用に到ってないが，意味の問題に後日当面するだろう．そして syntax と semantics の関係，grammar と vocabulary の関係を結ぶ解釈法の研究を急ぐ必要がある．（和田　弘）

56. 機械翻訳と国際語

K.G. Sellin: MT and/or International Language〔IX-4, pp. 142～143〕

国際的に見ると言語が政治上でも科学上でも障害になっている．機械翻訳が発展しているので，学術論文については共通な言葉 Interlingua を通すことによって，機械翻訳のプログラムが簡単になるか，あるいは翻訳者がいらなくなるか，など問題になっている．

しかし科学的でない言葉はとてもそこまでゆきそうもない．これでは会議での討論にも参加しにくい．そこで，これらの分野をも含めた国際語が望まれる．

かってのエスペラントは国際語ができそうなことを示した．しかし機械翻訳とこの国際語の研究とは，それぞれ協力して進めなくてはいけない．

国際語が備えるべき性質を述べてみよう．論理的に構成する言語であるから，1）普通の言語のように矛盾があったり不合理ではなく，2）習得に時間がかからず，3）政治的に偏せず，4）計算機の記号言語への基礎言語としての役割を果し，5）計算機に関連のある科学分野では直接科学的な言葉として実用になり，6）豊かでない国家でも，この国際語への機械翻訳の計

画なら実行できる．などの利点がある．

事実このような計画はスカジナビヤで作られそうだ．

従来言語の解析は言語学者にゆだねられていたが，これからは哲学者，論理学者，数学者や計算機界の人々が取扱って，もっと能率を高めるべきだ．そうすればこの国際語も夢物語には終らないだろう．

（和田　弘）

57. カナダ・ゼネラルエレクトリック社のランプ部門における計算機の使用

A. Leigh : Computer Uses at Lamp Department, Canadian General Electric〔I-2, pp. 8～12〕

現存する事務用計算機の多くは，各種のルーチンをたくさんもっていて，高速度計算機としてよりもむしろ事務的な統合機として有用であるといはれている．ここに，このランプ部門（売る人，造る人，輸送する人全部で 1400 人あまり）の例を述べるが，中形の計算機を事務統合機として使うのがよいようである．

カナダのゼネラルエレクトリックのランプ工場では，約5年の年月をかけて，なるべく人手の介入のいらない，やり直しの少ないやり方で適切なデータを提供する事務情報系を工夫した．使用した GE 225 というシステムは，8,192 語の磁心コア記憶装置（サイクル時間 18 μs，一語は 19 ビットと符号ビット），カードリーダ，ラインプリンタ，カードパンチ，6 個の磁気テープ（15 kc）からなっている．

カナダの人口は約1ダースの都市に集中していて，注文の 90% は六つの支店のストックから 24 時間以内に製品がとどくものとした．また，売れ行きには季節変化もあるし，人為的な変化もある．データとしては記録として義務づけられているものや，事務計画の資料となるものもある．これら数百の要請を考えに入れて工夫した．

日々の計画については，需要と供給の間に振動を起さないように，売り上げや供給の変動をつぎのようにならして入力とした．xを時系列として

平均　$A[x]=\alpha x+(1-\alpha)A[x-1]$

傾向　$T[x]=\alpha(A[x]-A[x-1])+(1-\alpha)[x-1]$

見積り　$R[x]=A[x]+\frac{1-\alpha}{\alpha}T[x]$

$0<\alpha<1$

これらを，α は大としていくつかの項で近似するのである．この他に，入力として，客名，品種名，地方名，セールスマン名，製造状態，勤務時間，原料の使用状態などがある．

1度シミュレートしたあとに作り上げたプログラムのスケッチが図示してあり，それは4個のブロックに分れている．その1は，在庫や売り上げの流れをとらえ，その2は在庫と売り上げの率を調正し，その3では，作業員の動員の必要性を検討し，その4では製造予定と仕入れ計画を作成している．こうして，計算機の価格の3～5倍の調査費を節約したという．

注文の少ない品種は，すぐストックがなくなるなどという結果もでている．

このあと引続いて，原料の制御とか，クリスマスツリー用のランプのように数週間で勝負する品種の製造や供給の問題も計算機にやらせてみる計画があると述べている．

（川合英俊）

58. 軍用通信網 Comlognet におけるプログラムの情報保護の点からみた特徴

A.E. Miller and A.B. Shafritz : Message Protection Features of the Comlognet Program〔X-4, pp. 162～165〕

軍用通信網 Combat Logistics Network はアメリカ空軍の最初の高速ディジタル・データ伝送系である．このシステムは電子交換およびプログラム記憶データ処理装置を使用している．

このシステムでは数百の端末を五つの自動交換センタを持つ通信網を用いて相互につなげる事ができる．交換センタはすべての入出力チャンネルを同時に働かせることができる．交換センタには集中データ処理装置があり，高速磁心記憶装置と基礎処理装置がその中核をなす．その外磁気ドラム，磁気テープ，高速行印刷機，コンソール，集配装置がある．

ある端末から他に情報を送るには，先方のアドレスを付けて情報を送り出す．集中処理装置は送られたメッセージをアドレス共全部記憶装置に読み込んだ後で，宛先へ送り出す．受信が完了すれば確認信号が発信側へ送られる．最も重要な事は，集中データ処理装置に障害が生じた時の対策である．データ処理装置として，オン・ラインとスタンバイの台を置き，切り替えられる．各種の処理状況に関する情報は，磁気テープ・磁気ドラムに重複して残しておき，切り替えられた場合にはまずそれまでのデータがこわされているかどうかを調べ，こわれていれば，その前の段階から処理を繰り返す．

本文にはオン・ラインのプログラムの障害対等上の仕事，切り替え時の再開プログラムの動作について述べている．（高島堅助）

59. 科学計算センターにおけるアナログ・ディジタル混合システムの利用

A. Debroux, G.P. Del Bigio, A. Gazzano, H. d'Hoop, A. Riotte and A. van Wauwe: Utilization of an Analogue-to-Digital Linkage System in a Big Scientific Computing Center〔Ⅶ-2, pp. 101～104〕

アナログ計算機を用いて一般科学問題を解く場合，問題の数式化，パネルへのセッティング，得られた結果のチェッキングなど，実際の計算時間に比べ準備段階と結果の処理に費す時間が非常に大きい．ディジタル計算機の諸機能をこれに取り入れ，より簡単に正確に問題解決を計ろうという試みがイタリアの Euratom Joint Research Center で始められた．用いられた機種はアナログ計算機は RACE 231-R と ADIOS システム，ディジタル計算機は IBM-7090, 1401, 1620 である．

まず初めは 1401 用として APACHE-1 計画（Analogue Programming And CHEcking）が作られた．その基本的な考え方は次のとおりである．

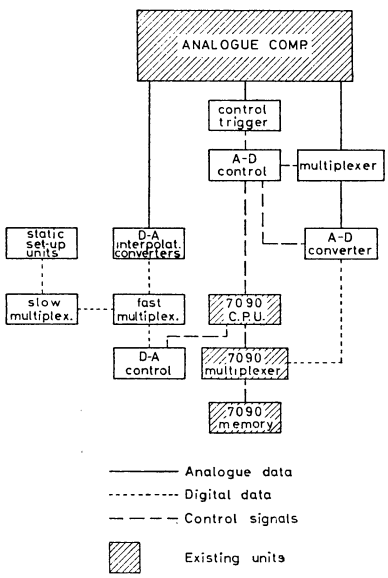
1) 数式は FORTRAN に類似した autocoding により，変数の初期値や最大値各パラメータなどはリストとしてディジタル計算機の記憶装置にいれる．

2) これに基づきアナログ計算機設定に必要な諸データ即ちポテンショメータの係数・プリパッチパネルのセッティングのリストなどを算出する．

3) アナログ計算機の解のチェッキングもディジタル計算機にさせる．

次にこれを基に 7090 用のより複雑な APACHE-2 が完成され，サブルーチンや索引の考え方も取り入れられた．アナログ計算機のサブルーチンは既知の解析された物理系のモデル数式で，これを組み合わせる事により，複雑な物理系により近い数式が得られ，数式化の誤差の減少を可能にし，また索引は初期条件やパラメータに用いられて係数設定の値の交換を簡単にした．このことにより，プログラマはこれらの繁雑さから解放され仕事量はほぼ半減し，またアナログ計算機操作の 50% を占めるこれらの手数が除かれたことによる経済的利益も大きい．以上が APACHE 計画の概略である．

しかしこれまでの方法はいわば両機の off line 的結合で，両機間の情報伝達には人の介入が必要であった．これに対し両機を on-line 結合とし，初めの数式のストアー以外全く人間の介入を除くために計画の第2段として，両機の動的結合（Dynamic Linkage）が現在研究されている．これは ADC および DAC を取り入れて情報伝達の自動化を計り，ディジタル計算機の割り込み機能，各ユニット間の相互独立性を用いて諸種のコントロールおよび計算処理を行ない，またリレー回路によるアナログ計算機素子の内部結合を利用してプリパッチパネルの自動化を行なおうとするもので，アナログ計算機とディジタル計算機の非同期組織（Asynchronous Organization）を作り上げる．ここに注意すべきことは，割り込み動作は各レジスタの内容保存その他で多くの命令語を要し，経済的にも不利である．処理すべきデータの多い事は計算機のスピードでカバーできるので1回の割り込みで処理するデータを多くして，割り込み回数をなるべく減ずるようにすべきである．この観点より ADC も計時制御によって絶えず変換を繰り返すのは好ましくなく，off limit detecter（ここでは Control Trigger と呼ばれる）を設け，ある値を越えた時に変換を行なうという方法がとられ，その変換スピードも計算機の速さより決められる．



第2図

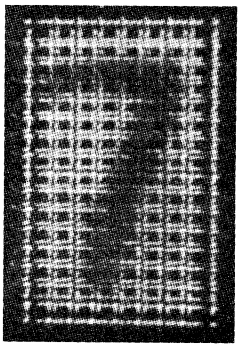
以上に述べた非同期組織のまず第一の利点は問題設定と結果のチェッキングのスピードで APACHE より

数段高い経済性を持つ．第2に多くの計算技術が可能になる．すなわちディジタル計算機がアナログ計算機の制御をし，自動スケーリング，浮動点動作，自動最適調整，繰り返し動作などを可能にする．またこれによりアナログ計算機のみでは実行不可能であった収益上の最適解を求めることも可能になる．1961年1月から始められたこの一連の研究は現在進行中であるが，今後の成果が期待される．（浜野桂吉）

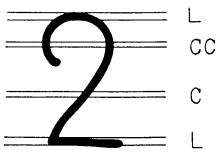
60. 文字のアナログディジタル読取

M. Nadler(Bull): Un Systèm Analogique-Digital pour la Reconnaissance de Caractères〔XIII-5, pp. 212～215〕

この論文の著者は，文字を判読する場合まず問題になる「有効な情報を雑音の中から抽出する」という点に関して独自の立場を表明しようとしている．



8×16 の基本ゾーンと数字 "7"



コードの例

すなわち従来しばしばみられたような，走査装置から出た video 信号をただちに沪過し座標点として認識する方法では，なまの video 信号に含まれている有益な情報をかなり無視してしまうので補正が必要となる．だから文字に関するなまの信号はそのままアナログ的にあつかい，有効な情報をできるだけ保存すべきだと主張する．

陰極管と光電増幅管とで構成された走査装置によって，各基本ゾーン中の文字のふちの切線の方向（ベクトル）をあたえる．こうしてスカラー的な認識ではなく，ベクトル的認識が可能になる．

認識された特長をもとにして，読みこまれた文字が何であるかを弁別する理論としては（商業的な機械においてはスカラー的情報から線の輪郭を再現することがむずかしいためあまり採用されていないが），topographique の論理の方がすぐれている．

現実の実験として数字を認識するためにつかった便宜的な水平方向のコードを紹介している．すなわち，C-走査と直交，L-走査と平行，CC-走査と二度交叉，の三種類の方向(ベクトル)をコードとする．このコードは走査の間に得られるベクトルを分析し，一つの仮定を与え，それがつぎの走査で決定されてさだまる．

基本ゾーンを横 8×縦 16 ならべて1字を読み取り 780 字/秒の速度が実験的に得られた．

この方式は Bull 社の光学文字読取装置に採用されている．（小田小枝）

61. 手書き文字の読取

L.D. Earnest: Machine Recognition of Cursive Writing〔XIII-6, pp. 216～219〕

光ペンで書いた語はただちに 100×170 の Bool 行列として計算機に入る．線の跡切れを埋めて，上下から水平線ではさみこむ．この線より上へのとびだし（*d* など）の数，下へのとびだし（*g* など）の数，（*t* の）横棒の有無，（*e* のように)中央で閉じている輪の数，中央の水平線を交叉する回数が調べられコード化される．これによって約1万語の入っている辞引をひき単語をきめる．以上の過程が MIT の TX-2 で約5秒/語の速さである．

実験してみると，18% とか 60% とかの成功率で唯一の単語が確認できる．その他の場合は複数個の候補単語がでてきてきまらない．現在のプログラムには学習能力はないが，皮肉なことに実験をつづけるうちに人間の方が機械の癖を学習する傾向がみられた．

TX-2 はいろいろと特異な計算機であって，たとえば金物で行列やリストが扱えるので，非常にうまく使うことができる．そこで読取システムはつぎのような構成になっている．

親プログラム-読取ルーチン
- Bool 行列プロセッサー-パタン行列
- リストプロセッサー-辞引などのリスト

（西村恕彦）