

文 献 紹 介

A: 数 値 解 析
D: 回 路 お よび 機 器

B: プ ロ グ ラ ミ ン グ
E: オ ー ト マ ト ン

C: 計 算 機 方 式
F: 応 用 そ の 他

B-35. 記号列変換のための公理言語: AXLE

K. Cohen and J.H. Wegstein: AXLE: An Axiomatic Language for String Transformation [CACM, Vol. 8, No. 11, Nov., 1965, pp. 657~661]

この論文は COMIT および SNOBOL と類似の方法で記号処理をする言語 AXLE を提案している。AXLE は記号列変換言語である。AXLE プログラムは記号列の変換および制御の流れを指定する imperative table (命令表) および命令に現われる記号を定義する assertion table から構成される。命令 (imperative) によって処理される記号列データは workspace と呼ばれる。

プログラム例

慣用の算術式のポーランド記法への変換

Imperative table Assertion table

1	$(P \circ Q) = \circ PQ$	$\circ = +$
2	End	$\circ = -$
		$\circ = \div$
		$P = M$
		$Q = M$
		$M = a$
		$\vdots \quad \vdots$
		$M = z$
		$M = \circ PQ$

このプログラムは workspace $((a+b)-c) \times d$ を $\times - + abcd$ に変換する。その作用の仕方を次に示す。

実行される命令 workspace の変化

	$((a+b)-c) \times d)$
1	$((+ab-c) \times d)$
1	$(- + abc \times d)$
1	$\times - + abcd$
1	命令 1 は実行されない。
2	End

記号の定義 $P=M$, $\circ=+$, $Q=M$, $M=a$ および $M=b$ に従って、命令 1 の左辺 $(P \circ Q)$ は workspace の $(a+b)$ を満足するので、それを右辺 $\circ PQ$, すなわち $+ab$ に変換する。したがって workspace は

$((+ab-c) \times d)$ となる。定義 $M=\circ PQ$ によって $M = +ab$, したがって $P=+ab$. 上と同様にして $(+ab-c)$ が命令 1 によって $- + abc$ に変換される。したがって workspace は $(- + abc \times d)$ となる。このようにして、workspace に命令 1 の左辺を満足する部分記号列がなくなるまで命令 1 が実行され、それから命令 2 へ制御が移る。この結果 workspace $\times - + abcd$ となる。

この論文は他に三つの例プログラムをあげ、言語の特徴および応用範囲を示している。 (二村良彦)

B-36. KDF 9 ALGOL におけるリスト処理方式

J.G.P. Barnes: A KDF 9 ALGOL List-processing Scheme [Computer J., Vol. 8, No. 2, July, 1965, pp. 113~119]

ALGOL の言葉で書かれた procedure の群を用意することによって、リスト処理のプログラミングをするという試みが報告されている。実験には KDF 9 ALGOL が用いられた。全体の構成は LISP にならっている。

LISP では計算機の 1 語の半分が左リンク、残る半分が右リンクとして用いられているが、ALGOL を利用するときはそのような詰め合わせはできない。そこでその代りに二つの integer array item および link が定義され、対応する要素の組が item [x], link [x] x 番目の list cell として用いられる。プログラムの上ではリストは list cell の番号を値とする整数型の変数によって代表される。また atomic symbol としては整数だけが許される。リストは外部媒体上では、整数をかっことコンマによって区切ったものであらわされる。そしてそれを integer array item および link の上の適当な構造と対応させるための入出力ルーチンが用意されている。

整数を atom として定義する procedure, LISP の関数 equal, car, cdr, cons に相当する procedure などが、すべて ALGOL の言葉で (code を使わずに) 書かれている。また不要になった list cell を他の目的に使える形になおすための garbage collector

に相当するものも組み込まれている。

このリスト処理方式を使うには、前記 procedure 群をまず自分のプログラムの中に書きうつし、次にこれら的基本 procedure を呼び出して望みの操作をおこなわせればよい。

この方式のよいところは、それがすべて ALGOL で書かれているためプログラム交換に便であり、また数値計算とからんだリスト処理の問題を解くのに有利なことであり、またまずい点は、(1) atom が整数に限られていて文字データなどは直接は扱えないこと。(2) 一つのセルに二つの整数型変数を要するので記憶容量の点で損であること、および(3) プログラムの中にリストを書き込む(たとえば $A:=(1, 2, (3, 4))$ というように) ことができないので、プログラムがやや見にくくなる。

なお、この種のいわゆる host-embedded list processor としては SILP (Comm. ACM, Vol. 6, No. 9, Sept. 1963, 524~544) が有名であるが、本論文のものは FORTRAN ではなく ALGOL を用いている点、したがって procedure の再帰呼出しの可能性がある点、機械語で書かれた基本サブルーチンを要しない点でこれと異なる。

(木村 泉)

C-37. 有限オートマトンの一次配列によって素数を実時間に発生すること

P.C. Fischer: Generation of Primes by an One-Dimensional Real-Time Iterative Array [JACM, Vol. 12, No. 3, July, 1965, pp. 388~394]

Firing Squad Synchronization Problem に用いられた方法により、素数を発生する有限オートマトンの一次元、半無限の配列を構成した。 t 番目のビットが時刻 t に発生し、 t が素数なら 1、素数でなければ 0 となるような 2 進数の列を発生する。この問題は、T. Crowley により提出され、二次元の配列については解かれている。1 からはじまる整数の列を考え、 k 番目のステップでは、 $k^2, k^2+k, k^2+2k+\dots$ と k^2 から始まって k ごとの数を消していく。このステップを $k=2, 3, 4, \dots$ と行なって消されなかつものが素数である。この方法は、普通のエラトステネスの方法よりも能率が悪い。 $k=2$ のステップを行なうには、1 番目と 2 番目のオートマトンの間を時間 2 (時間のスケールの変更が必要) で往復する波をつくり、その波が 1 番目のオートマトンは到達するごとに、0 出力を出すようにし、最初の 0 出力が時刻 4 に出るようにす

る。同様に $k=3$ のステップを行なうには、3 番目と 6 番目のオートマトンの間を往復する波をつくればよい。このような波の往復によって、必要な周期の発振器をつくるために、最初に、発振器製造用の波を出すことにより、それが行なえるようにしておく。

この方法によるとオートマトンの状態数は $37^3 = 50,653$ あれば十分であることが示されている。

(相馬 嵩)

C-38. 連想記憶構造

B.T. McKeever: The Associative Memory Structure [Proc. FJCC, 1965, pp. 371~388]

Associative Structure Computer (ASC) についてシリーズで報告される論文の一つで、ASC の機能についての全般的な特徴とその基本要素の一つである Associative Memory Structure およびプログラム変換、プロセス制御などへの応用例が述べられている。ASC の具体的な装置の構成および ASC の基本的な module の一つである Associative Switching Structure の詳細については 1966 年の IEEE Trans. EC-15 に発表される予定である。

ASC はバッチシステムで製造される素子、特に cryotron などの使用に伴って考えだされた、論理機能と記憶機能とが従来のように明確には分離されていない計算機で、その設計は cellular な方法がとられているが、純粋に cellular なアプローチによるものではなく、すべてが均等とはかぎらない array の array を基本としている。

すなわち、その回路構成の特徴は gate のミクロな array でもなく計算機のマクロな array でもなく、むしろ電話網に近い。

マクロ的な構造は Associative Memory Structure と Associative Switching Structure という二つの module のみを基本とし、両者とも記憶、論理の両方の機能をもつが、前者は記憶を主、後者は論理を主とする。

ミクロ的な構成は、記憶と論理を独立、並列に行ないうる Associative cell と、信号の集中、連結、分配を行なう Accumulator cell の二つから成り、これからユニット関数となる少数の簡単な情報処理演算が金物的に構成され、このユニットを組み合せることにより並列順序論理の機能を与える。論理設計においては、回路設計を行なってから接続パターンを考える従来の方法とは逆に接続パターンから回路設計を行なう

方法がとられている。

Associative Memory Structure は、情報蓄積を行なう associative cell とワードコントロールを行なう logical accumulator の 2 種の cell から成り、一致検出線と不一致検出線の両方を用いることにより AND, OR などの論理を含む探索、書き込み、読み出を行なう桁処理と各語に二つのフリップ・フロップを用いて各種の選択、分配を行なう語処理の 2 種の built-in 演算が作られている。アクセスのしかたには直接アクセス、条件つきまたは間接アクセス、逆アクセスおよび並列アクセスの四つの mode がある。結論として、Associative Memory structure は多用途の情報蓄積および処理機能をもった計算機用のブロックで探索のみに限定されないこと、ユニット関数を組合わせることにより built-in のマクロ命令よりもはるかに flexible かつ並列な演算処理が可能であることが示される。
(小林正明)

D-39. 最悪ケースを考慮した論理回路の設計

W. Bongenaar and N.C. Detroye Worst Case Considerations in Designing Logical Circuits [IEEE, Trans. EC, EC-14, No. 4, Aug., 1965, pp. 590~599]

最悪ケースを考慮した場合の論理回路の設計について述べられている。本論文では、特に温度変動と抵抗のばらつきに注目し、直流条件における最適化の方向づけを行なっている。本論文で検討する論理回路の例としては、TTL, DTL, ETL の三つである。

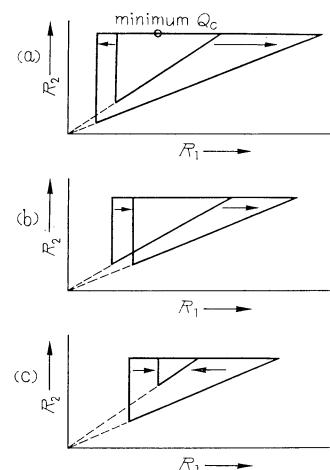
電子計算機においてあらゆる機能は一つの基本論理回路で達成される。その際論理動作は、どんな条件下でもあらかじめ決められた値を満足しなければならない。すなわち最悪ケースにおいても論理回路が正しく動作する必要がある。以上のことを考慮して所要ファン、アウト数、最大電力消費量など基本的要件を与えたときに、論理回路が正しく動作する領域が回路定数をパラメータとしてえられる。これを二、三の回路形式について求めるのが本論文の目的である。

まず検討に先立ち各回路定数、例えば供給電圧 V_p 、回路抵抗 R_i 、on 状態におけるベースエミッタ間電圧 V_{beon} などが温度変動するとする。さらに抵抗 R_i については抵抗自体のばらつきを考慮する。

(a) TTL circuit の検討

基本的要件としては、off 状態における雑音余

裕度、on 状態における負荷を考慮した動作条件、電力消費量等の条件を与える。つぎに off 状態、on 状態における最悪ケースの回路状態を考え、上の要求条件がこの回路状態においても、満足しなければならないとして満足すべき不等式を設定する。これよりパラメータの一つである供給電圧 V_p と固定すれば、 $Q_c = R_c/R_1$ を変数として、各不等式の満すべき領域がえられる。また Q_c を固定すれば R_1, R_2 に対して領域を決定することができる。このときパラメータ Q_c, V_p, F_{out} などをふやしたときに R_1, R_2 に対する解の領域がどのように変化するかを図示できる(第1図(a), (b), (c))。これらの性質が、最適化への方向づけとなる。



第1図 The boundaries of the area of solution for the TTL circuit as affected by increasing Q_c (a) increasing V_p (b) and increasing F_{out} (c).

(b) DTL circuit, (c) ETL circuit についても同様の検討を行なっている。これらの検討より各回路形式に特長的な性質がみい出される。

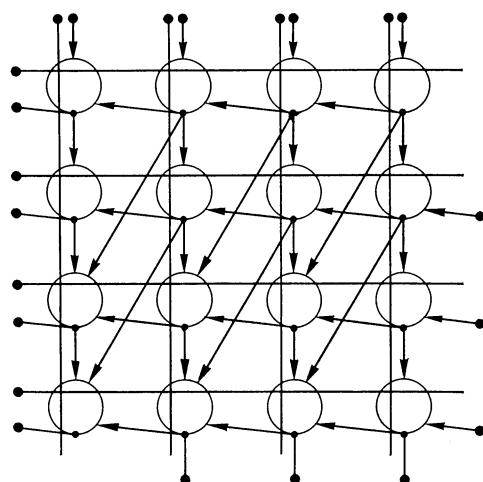
結論として、本論文で行なった最悪ケースを考慮した設計方針は論理回路が正しく動作するという基本的要件のみを考慮に入れた検討であるが、これはあらゆる設計方針の出発点である。すなわちこの考え方から、パラメータのとりうる組合せが満足する領域を決定することができるとしている。
(寺島信義)

D-40. くもの巣状のセルラー・アレイ

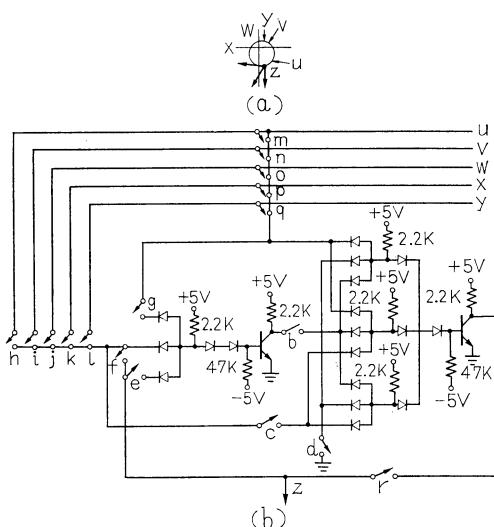
R.C. Minnick: Cobweb Cellular Arrays [Proc.

FJCC, 1965, pp. 327~341]

著者は前にカットポイントアレイによる論理回路の構成を発表したがこれには(1)ある種の論理回路、特にパラレル演算の回路では素子数が非常に多くなる、(2)アレイの側辺同志を結びつける信号線が必要である、(3)素子の各々の孤立化が不十分で故障対策に不便である、などの欠点があった。この論文はカットポイントセルに若干の修正を加えてこれらの欠点を改善したカブウェブアレイについて述べてある。その修正とは(1)各素子に新られた14個のカットポイント



第1図 Structure of the cobweb array.



第2図 Diode-transistor realization of cobwell cell.

トを追加することにより信号の方向変換、あらゆる端子間の短絡を可能にする、(2)垂直方向のバスラインを追加することによりアレイの下端から上端へ信号を逆送できるようにする、(3)水平方向の信号の逆送ができるように素子間に右から左への信号線を追加する、(4)2個のカスケード論理回路が交差できるよう桂馬跳び(a knight move)の信号線を追加することである。カブウェブアレイの構造を第1図に、素子の回路図の一例を第2図に示す。第2図においてx, y, u, v, wが入力線でzが出力線であるが、これらの信号線はアレイ上で交差している(第1図参照)から、これを集積回路で実現するためにはy, v, wを含む層とx, uを含む層との2層構造をとる必要がある。

パラレル演算の場合カットポイントアレイでは被演算数の桁数の2乗のオーダで素子数がふえるが、カブウェブアレイでは桁数の1乗のオーダでしかふえない。たとえばnビットのパラレル加算器ではカットポイントアレイの場合 $(2n+1)^2$ 個の素子が必要であるが、カブウェブアレイの場合には $(9n+3)$ 個の素子が必要なだけである。

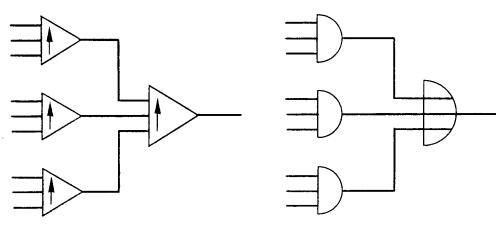
アレイの中の幾つかの素子が故障している場合の対策にはカットポイントアレイの場合とは異った考慮が必要である。ここでは4個ずつの素子を1組にして、これに1個の素子と同一の機能を持たせることによって4個に3個ずつの割合で遊んでいる素子を作り、その4個の内の働いている素子がこわれた場合には他の3個のどれか一つに切換えるという方法を考えている。もちろん必要な素子数は4倍になる。

カットポイントアレイとカブウェブアレイの比較例として12ビットのシリアルな掛算器の構成が示してある。この場合必要素子数はふえるが、アレイ側辺の結合線には大幅の減少が見られる。(三上 啓)

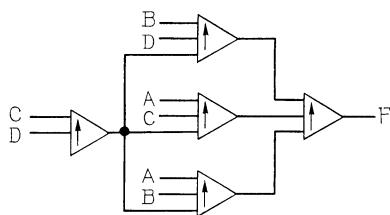
E-41. NAND または NOR 素子による組合せ論理合成

D.T. Ellis: A Synthesis of Combinational Logic with NAND or NOR Elements [IEEE Trans. EC, EC-14, No. 5, Oct., 1965, pp. 701~705]

NAND (NOR) 論理の合成法は種々発表されているが、最も primitive なものは、Quine-McCluskey の方法による積和(和積)2段合成法の転用である。Q-M 法によって主項を求める最小の組合せを選んだのち、選ばれた各主項の因子に相当するリテラル



(a) NAND 2段構成 (b) 等価の AND-OR 構成
第1図



第2図 応用例

を第1図(a)のような2段NAND構成の入力とすれば、AND-OR合成と同様の手法で最小化ができる。ただし、現実的には、NAND回路網の入力を肯定変数のみとするために、第1図(a)の左に否定用の1入力NANDを付加する。

Ellisは、この入力段の1入力NANDの個数を削減するための三つの規則を導き、その応用例を示している。これらは、主項をMcCluskeyの表記法で列記した表に関するもので、入力段の1入力NANDを複数個まとめて1個の複数入力NANDで置きかえる手順を示したものである。この手順によって、たとえば

$$\begin{aligned} F &= AB\bar{C}D + ABC\bar{D} + ABC\bar{D} + A\bar{B}C\bar{D} + \bar{A}\bar{B}\bar{C}D \\ &= BCD + ACD + ABC \end{aligned}$$

は第2図のように実現される。NOR論理についても、これと双対の方法で合成できる。

超小形技術の方向として、論理素子(基本回路)総数最小化という在來のcriterionはかなり大幅に修正されねばならないことが明らかである。しかし、論理設計の自動化を考慮する場合、論理網の冗長度に関する検討は基本的に必要なものであるから、その意味で、Ellisの一見自明な結論も注目される必要があろう。

(苗村憲司)

E-42. オートマトンの自己同型写像群と 入力同値類の集合

B. Barnes: Groups of Automorphisms and Sets of Equivalence Classes of Input for Automata

[JACM, Vol. 12, No. 4, Oct., 1965, pp. 561~565]

さきに、Fleckは、オートマトンのそれ自身の上への自己同型写像を定義し、この全ての自己同型写像の集合が群をなすことを示した。

またWeegは、オートマトンの自己同型写像群とその入力半群の同値類との間の関係を調べ、強く結合されたオートマトンの各自己同型写像群は、入力半群の同値類の群のひとつのものに同型であるという結果を得ている。

本論文は、逆に、オートマトンが無意味でない自己同型写像群を持つための十分条件が、同値類に対するどんな条件となっているのかという問題について論じ、強く結合されたオートマトンの各状態に対して、群をなす、入力半群の同値類の集合の部分集合が存在するならば、この群はオートマトンの自己同型写像群である。という結果を得ている。

また最後に、得られた結果が、オートマトンの自己同型写像群を決定するalgorithmをも与えていることを指摘している。

(村上国男)

E-43. オートマトンの自己同型写像群について

A.C. Fleck: On the Automorphism Group of an Automaton [JACM, Vol. 12, No. 4, Oct., 1965, pp. 566~569]

オートマトンの持つ代数的性質について、いくつかのことが明らかにされている。

\tilde{I} をオートマトンの入力の全ての同値類の集合とし、 $G(A)$ をオートマトンの自己同型写像群とする。

まず、 n 状態を持つ強く結合されたオートマトンを A とするとき、 J_2 が J_1 の正規部分群であり、剩余群 J_1/J_2 が $G(A)$ に同型であるようなCharacteristic semigroup \tilde{I} の部分群 J_1, J_2 が存在する。という証明がなされ、次に A を状態数 n 、 $G(A)$ のorderが n の強く結合されたオートマトンとするとき、 $H \cap K = 1$ かつ $HK = G(A)$ であるような $G(A)$ の部分群 H, K が存在するための必要十分条件は、 A が $A/H \times A/K$ と同型となることである。という結果が得られている。

(村上国男)

F-44. オートマトン理論とその応用

V.M. Gluskov: Automata Theory and its Application [Proc. IFIP Cong., 1965, pp. 1~8]

オートマトン理論は大きく分けて、次の三つの分野からなっている。i) 抽象オートマトン理論、ii) 構造オートマトン理論、iii) 自己組織および学習システムの一般理論である。i) は内部状態の集合 α 、入力信号の集合 χ 、出力信号の集合 τ および一つの写像 $\alpha \times \chi \rightarrow \alpha$ 、 $\alpha \times \chi \rightarrow \tau$ によって記述される離散的オートマトンをあつかう。ii) では α 、 χ 、 τ は有限個の基本信号（普通は2値の信号）の組合せでコード化され、オートマトンは一つの回路網としてあらわされ、基本信号を記憶したり変換したりするものとして扱う。iii) では入出力によって応答が変化するようなオートマトンのはたらきについて研究する。このような変化は、抽象的なレベル（オートマトンの内部状態の変化）と回路網の構造の変化を許すレベルとから考察する。

各分野はさらにそれぞれこまかく分けられる。抽象オートマトン理論は、有限オートマトンとある種の無限オートマトン理論に分けられる。また半群論のある部門やアルゴリズム理論ばかりでなく、形式言語論や形式文法論もこれと密接に関係がある。構造オートマトン理論には、符号化理論、論理函数および組合せ回路の理論等があり、数学的論理学（特に命題論理）や情報論理、信頼度の理論等はこれと関係が深い、自己組織および学習システムの理論としては、図形認識、ニューロン回路網の理論や、パーセプトロンの理論がある。これに関する分野としてゲームの理論、統計的意志決定理論、マルコフ過程の理論、発見的なプログラミングや定理証明プログラミングの問題などがある。

オートマトン理論は最初簡単な論理回路の解析、合成、簡易化に応用されたが、最近の発展（とくに無限オートマトン理論）によって電子計算機の形式的設計の基礎を与えた。とくにマイクロプログラムの簡易化に用いられる可能性があり、また ALGOL 型の言語を翻訳する計算機の構造設計にも用いられる。また言語翻訳の形式化、符号化、復号化の問題、離散的外挿法の応用の問題が考察されている。図形認識の理論は視覚および音声信号の認識の問題や、プロセス制御において発生する状況の認識の基礎となっている。また言語における図形認識の理論の結果が将来計算機の論

理的構造に大きな影響を与えるだろう。自己組織システムの理論の生物学への応用も重要な問題であろう。たとえば動物や人間のいろいろな行動の型をシミュレートすることによって、頭脳の働きや生物体の進化の問題を研究する手がかりが得られるだろう。

以上のべた問題につき、最近の研究例をいくつか具体的にあげて説明してある。
(相馬嵩)

F-45. 大容量ランダムアクセスファイルを持つ計算機センタの共同利用

J.M. UNK: General Purpose External Memory System for Data Base Processing (ISAR BASE)
[Proc. IFIP Cong., 1965, pp. 267~271]

オランダのハーグにある、大容量のランダム・アクセス記憶装置を持つインフォメーションセンタを、いくつかの会社がいろいろな業務に通信回路を通して共用するシステムについて述べている。

インフォメーションセンタは、大容量のランダムアクセス記憶装置と、データ処理用の計算機2台、ファイル処理および端末制御用の計算機2台、各種交換機3台、その他の装置から成っている。ファイル処理および端末制御用の計算機がこのシステムの主な部分で (ISAR base—Information Storage And Retrieval 一と呼ばれている)、いろいろな会社の端末装置から来る種々のフォーマット、種々の伝送速度のメッセージを処理して、大容量ファイル、データ処理計算機に定まったフォーマット、定まった速度のメッセージを送るようになっている。

端末装置には利用会社に固有の番号が与えられていて、メッセージにはハードウェアにより自動的にその番号が附加されるので、他の会社のデータを読み出したり、故意に変更したりすることはできない。この番号の処理は、ISAR base により自動的に行なわれる。

データ処理用計算機に対するプログラムは、ISAR base により、メッセージのフォーマットが標準化されていること、ファイルへのアクセスを他のメッセージによるものと関係なくプログラムが組めること、多くの基本的なプログラムが準備されていることなどによって、他の同様なシステムに比べて簡単にできるようになっている。
(関栄四郎)