

セル構造オートマトンによる並列処理

京大理学部 西尾英之助

II. 序論

セル構造オートマトンは 1) 同一の素子が 2) 多数結合されて
 できる系である。Von Neumann はこの様な系の 逐時的動作 に
 よってオートマトンやキューリング機械の埋込みの問題を扱
 い、自己増殖の論理モデルを作った。¹⁾ 逐時動作というのは、
 ある時某で状態変化をする素子が 1 個であり、かつその素子
 が空間的に隣合った素子に順に移ってゆく動作形式のこと
 である。残りの素子は状態変化をしない。このような静状
 態の素子は、しかるべき時期が来るまで待っていると考
 えれば、やはり処理目的に貢献していることになる。

Von Neumann の仕事の後から始まった Moore, Myhill など
 による“エレンの圈問題”は“斉射撃問題” (同期問題)²⁾ を生んだ。
 これは同時に全素子が同じ状態になることを要求するの
 だから少なくとも最後の瞬間には全素子の 並列動作 が必要
 である。さらに問題は 最短時間解 を求めることになったから、
 並列動作の必要性が高まった。

他方一般に情報処理系については原理的に処理目的の達せ
 られるか否かの問題と同時に所要時間の問題がある。計算

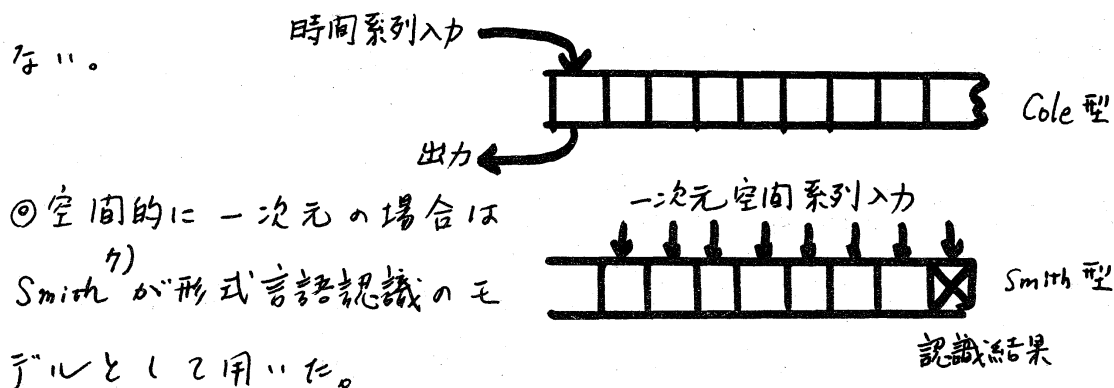
機システムが大型化して多数の装置が組込まれるにつれて、それらを並列動作させるモニターが必要になった。こうしてメモリと処理時間の短かくするために並列処理が必要になったように見えるが、本来処理されるべき情報の空間的拡がりを持ってメモリから並列に処理される場合もある。視覚情報の処理はさうで、人工の装置で逐時処理をするのは、経済性と考え易さのためである。

2. 処理目的とセル構造のモデル

系への入力の子の方と結果の取出し方の違いにより、解析方法や論じよう命題、所要時間の長短など、大いに違ってくる。従来扱われてきたモデルに言及しよからこの臭を整理しよめる。

1) 入力情報の一次元の場合:

①時間的に一次元(空間的に0次元)の方向性を持った、形式言語の子的な情報を空間的に一次元(あるいは n 次元)のセル構造で処理するモデル——これは言語認識モデルとして Code³⁾ が用いたのと、Code型と云ふ。Fischer⁴⁾ が実時間の素数タイピング¹⁾ を発生する系をつくつたし、Byxutab⁵⁾ 也有次⁶⁾ が実時間で計算できる関数を論じた。これら一次元のセル構造の並列処理能力を利用してメモリをどう使うよべき



入力系列は系の初期状態として与えられる、系は自律動作をする。結果はある特定の素子に出る。後述べる“並列・実時間ソート”のモデル⁸⁾では結果は全セルの最終状態として与えられる。一斉射撃や“French Flag問題”⁹⁾もこのような入出力の与え方の例である。まとめ、一次元空間パターンの(実時間)変換の問題と云えよう。Cole型で時間系列の実時間変換を扱った論文は手になん。

2) 入力情報の二次元の場合:

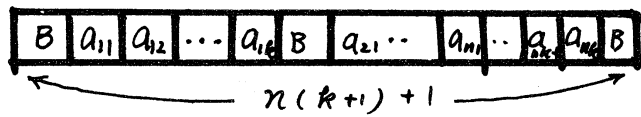
① 時間+空間で二次元の情報はまだセル構造で扱われていない。Hennie¹⁰⁾の iterative array も主な結果は時間的に塊のりのない場合についてである。関¹¹⁾がこのような情報の認識問題を考えているの手に明確な結果が出ていない。

② 空間的に二次元の場合は Smith型²⁾図形認識が扱われていた。Unger¹²⁾, Papert, Minsky¹³⁾, Beyer¹⁴⁾, 稲垣¹⁵⁾など。しかし、処理時間や並列処理の点で系統立つ議論は少ない。Smith¹⁶⁾。しかしこれも認識問題で、変換問題としては、一斉

射撃の二次元への(本質的では無い)拡張があるから...¹⁷⁾
 3.

3. 2進数の実時間ソート⁸⁾

n 個の k -bit 2進数の列 $\{a_1, a_2, \dots, a_n\}$ が与えられたとき、
 図のように入力型を用いて、大きい順に並べ替える問
 題。ただし数 a_i は



$a_{i1} a_{i2} \dots a_{ik}$ と表現

これ、数の区切り記号 B を用いる。系の素子は n, k に依
 存しない有限オートマトンで、ソート終了時点で系は自動的に
 停止せねばならない。

入力情報の量は $n(k+1) + 1$ だから、 n, k の order の時間でソ
 ートの完了する解を求めろ。

◎ 解の概要:

a) local three-sort T : n 個の 3 組の列の変換 T
 と考へる。

$$\begin{aligned}
 X &= \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ \gamma_0 \end{pmatrix} \begin{pmatrix} d_1 \\ \beta_1 \\ \gamma_1 \end{pmatrix} \begin{pmatrix} d_2 \\ \beta_2 \\ \gamma_2 \end{pmatrix} \dots \begin{pmatrix} d_i \\ \beta_i \\ \gamma_i \end{pmatrix} \dots \begin{pmatrix} d_n \\ \beta_n \\ \gamma_n \end{pmatrix} \begin{pmatrix} \infty \\ \infty \\ \infty \end{pmatrix} \begin{pmatrix} \infty \\ \infty \\ \infty \end{pmatrix} \\
 &= x_{-1} x_0 x_1 x_2 \dots x_i \dots x_n x_{n+1} x_{n+2} \\
 T(x) &= x'_i x'_0 x'_1 \dots x'_i \dots x'_n x'_{n+1} x'_{n+2}
 \end{aligned}$$

T は

$$x'_i = \begin{pmatrix} d'_i \\ \beta'_i \\ \gamma'_i \end{pmatrix} = \begin{pmatrix} \min(\gamma_{i-1}, \beta_i, d_{i+1}) \\ \text{mid}(\gamma_{i-1}, \beta_i, d_{i+1}) \\ \max(\gamma_{i-1}, \beta_i, d_{i+1}) \end{pmatrix} \quad \begin{matrix} i = 0, 1, \dots, n+1 \\ x'_{-1} = x_{-1}, \quad x'_{n+2} = x_{n+2} \end{matrix}$$

$0, \infty$ はそれぞれ数より小、大の数を示す記号。

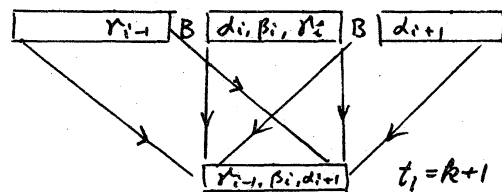
「ソート可能な数列 $\{a_1, \dots, a_n\}$ に対して, $x_i = \begin{pmatrix} a_i \\ a_i \\ a_i \end{pmatrix}$ なる3組に対応させ, その列 $x = x_1 x_2 \dots x_n x_{n+1} x_{n+2}$ とする. x に local three sort T を施すと, やはり同様の3組の列が得られる. この T は π の性質を持つ.

[命題] T を高々 $1.5n$ 回施せば, $\beta_1 \beta_2 \dots \beta_n$ は $a_1 a_2 \dots a_n$ のソート結果になっている. (証明未定)

b) セル構造オートマトンによる T の実現:

初期状態: 各素子は, a_{ij} ($0 \leq i \leq 1$) の3組 $\begin{pmatrix} a_{ij} \\ a_{ij} \\ a_{ij} \end{pmatrix}$, あるいは印記号 B の状態とする. 数 a_{ij} に対応する素子の部分を i -block と呼ぶ.

phase 1: $d_{ij}, \beta_{ij}, \gamma_{ij}$ は単位時間の遅れ. 天を左, 中, 右へ走る. $(k+1)$ 時間後, i -block は $\gamma_{i-1, j}, \beta_{ij}, d_{i, j}$ ($j=1, \dots, k$) に衝突する.



phase 2: bit ごとく大小比較をし, 同一マシンの全素子にその結果を配る. 可及的.

$\gamma_{i-1, 1} \dots \gamma_{i-1, k} = x_1 x_2 \dots x_k, \beta_{i1} \dots \beta_{ik} = y_1 \dots y_k, d_{i+1, 1} \dots d_{i+1, k} = z_1 \dots z_k$ とすれば, 例として, $\xi = \xi_1 \dots \xi_k, \xi_i = \phi(x_i, y_i), \xi_i = 0$ ($x_i < y_i$), $\xi_i = 1$ ($x_i > y_i$) の5) をベクトルと見做す. これは1時間ごとく出る. もし $x > y$ ならば ξ_i の最高位0-1 は i の ϕ の桁が1となる. これを i -block

の全素子に配る。同様に、他の2組 $(x-z, y-z)$ の比較結果も並行して全素子に配る。これを $(\tilde{x}, \tilde{y}, \tilde{z})$ とすれば、下記の判定表により、 x, y, z の大小順がわかる。

min	mid	max	
x	y	z	$\tilde{x}, \tilde{y}, \tilde{z}$ が1を含む...とす。
y	x	z	$\tilde{x}=1, \tilde{y} \neq 1, \tilde{z} \neq 1$
x	z	y	$(\tilde{x}, \tilde{y}, \tilde{z}) = (\emptyset \emptyset 1), (\emptyset 1 \emptyset), (001), (0 \emptyset 1)$
z	x	y	$\tilde{x} \neq 1, \tilde{y} = \tilde{z} = 1$
y	z	x	$\tilde{x} = \tilde{y} = 1, \tilde{z} \neq 1$
z	y	x	$\tilde{x} = \tilde{y} = \tilde{z} = 1$

従って各素子 (bit) ごとには新しく d_{ij}, b_{ij}, s_{ij} が取られる。bit ごとには大小比較を数回行う。全素子に配り、上記表を以て操作する。合わせて k 時間経過する。この時間は phase 1 と合わせて、 $k+1+k=2k+2$ である。local three sort の一回分の終了した時刻である。

同期の問題: 各素子は phase 1, 2 の終了時刻ごとの処理に移るわけである。この時刻を一言に知り必要はない。k は任意である。素子内には時計を持つことは出来ない。この一言解題の手法を利用する。区切記号 B のセ〜の両端にあるから、長さ k の各プロセス内では k 時間同期心と取り。これは上述の phase 1, 2 の $k+1$ 時間経過を待つと予備し

なり。最後に、[命題] により、 $1.5n$ 回の T の適用によって
 トが終了するから、この時刻、 $1.5n \cdot 2(k+1) = 3n(k+1)$ を全
 素子を用いた同期問題の解として求めることにできる。

結局、求める素子は、本稿の処理 (phase 1, 2), と 2種類の
 同期部分の直積の形で構成されたことになる。

②この問題の最短時間解は $2n$ 以上だから、 n に近いべき
 法以外にはより良い設計があるかも知れない。local three
 out の拡張 (m-sort) もこの一つの可能性である。Cole 型
 の実時間ソートは考えられかねない。

4. おわりに。

いくつかの例から明らかなように、セル構造のトポロジ
 については、必要な情報をしかるべき位置へ伝達し処理するが、
 伝達に要する時間・総時間・限界を定める。これは反して
 処理装置の1個で補助記憶を持つ系では、必要な情報を10
 分の時間かかる。今後セル構造の並列処理に適したこ
 の問題を発見して、解くことにしよう。この系の特徴をよく
 理解しよう。しかし、並列処理を単に、処理速度を
 高めるという側面のみではなしに、逐時処理の原理による
 状態遷移をよせよという系をも調査を行わねばならぬ。

[文献]

- 1) von Neumann: A.W. Burles, ed. Self-reproducing automata (1966)
- 2) 例之12) Waksman, A: Information and Control (1966)
- 3) Cole, S.N: IEEE C-18 (1969)
- 4) Fischer, P.C: JACM 12 (1965)
- 5) Быхватов, Ю.А.: Проблемы Кибернетики, (1967)
- 6) 有次: 通信学会誌 54-C (1971)
- 7) Smith II, A.R: JCSS 6 (1972)
- 8) 西尾: 通信学会全国大会 S8-8 (1973)
- 9) Wolpert, L: in Towards a Theoretical Biology I. (1968)
- 10) Hennie, III, F.C: Iterative arrays of logical circuit, MIT Press (1961)
- 11) 園: 私信
- 12) Unger, S.H: Proc. IEEE 47 (1959)
- 13) Minsky, M & Papert, S.: Perceptions, MIT Press (1969)
- 14) Boyer, T: PhD dissertation, MIT (1970)
- 15) 箱垣: 九州大学シンポジウム (1972)
- 16) Smith II, R.A: IEEE switching and automata Theory (1971)