数理解析研究所講究録
259巻 1976年 136-146

半順序グラフと平行プログラムについて

長崎大医　中村剛
植村憲治

非同期式計算機として有名なイリヤック正が発表されて15年余りになり、いわゆる〝非同期〟という思想は広まり、除々に計算機設計にも活かされてきているように思われるが、未だイリヤック正を超える本格的な非同期式計算機が設計されたという話は聞かない。もっともこれは我々の耳に入らないだけで、どこかに存在しているのかもしれない。非同期回路の中でもSpeed-independent回路の長所は、使用する素子の反応速度についてはなんら知らなくて良いということである。すなわち、各素子の速度には無関係に設計者の意図したとおりの正しい動作をすることであり、この為にたとえ使用している素子が動かなくなってしまった(出力が変わらなくなってしまった)場合にも誤動作は生じず、ただその素子と論理的なつながりを持った素子の一部の動作が制限されてしまうだけである。又ひどく速度の違う素子を使用する場合に

は同期式に比べて、はるかに全体としての処理速度が上がることが期待される。一方短所はなんといっても論理設計の困難さが第一である。イリヤック皿の場合はフローチャートなる指図書をもとにカンと経験で設計しつくしたと報告されている。現在でもその状況は変わらず、統一的あるいは普遍的な設計手順で充分実用的なものは開発されていないようである。イリヤック皿を設計したグループのうちのD. E. Millerらにより創始されたセミモジュラー回路理論はその後むしろ数学的理論として発展してきて、最近ようやく、もとの設計図がセミモジュラーの時にはそれをある基本的な論理素子のみで実現できる手法が開発された。しかしながら、その手法により構成される回路では素子の数が多くなり、実用に適するかどうか疑問である。一つにはもとの設計図がセミモジュラーという点にも問題があるように思える。すなわち、セミモジュラーというのは実際の回路設計においては広すぎる概念と思える。実際イリヤック皿で用いられたフローチャートは、一部の特殊なものを除けば、セミモジュラーよりはるかに狭いクラスで記述されてしまう。又最近のLSIの発展をみる時に、回路を構成する最小単位を論理素子にするよりも、より機能の広いオートマトンとしたアプローチもなされねばならないであろう。オートマトンを設計に使用するさいの理

論的な方面からの難点としては、個々のオートマトンを記述する為の手段が状態遷移図であるということである。状態遷移図は直観的にとらえにくくしかも冗長なことが多い。代数的オートマトンの考えにもとづいたコンパクトなオートマトンを想定してみるのが上の難点を克服できる一つの手段かもしれない。ここでは論理設計を単純化しかつ通常必要とされる記述能力を備えた指図書のクラスについての一考察を述べることにする。

§1. Partially Ordered Graph

H を有限集合、$\mathcal{A}$ を $H \times H$ の部分集合とする。H の要素 h をノードと呼び、$\mathcal{A}$ の要素 (h, k) を h を始点、k を終点とする矢と呼ぶ。$\mathcal{A}$ は通常のダイグラフ（向きのつけられたグラフ）である。H の要素の列 $h_1, h_2, \cdots h_m$ で $(h_i, h_{i+1}) \in \mathcal{A}$ $(1 \leq i \leq m-1)$ となるものを $\mathcal{A}$ 上の列と呼ぶ。特に異なる要素からなるものを単純列と呼ぶ。$\mathcal{A}$ が次の P1, P2, P3 を満足するとき Partially Ordered Graph (POG) という。

(P1) 全ての $h \in H$ について $(h, h) \notin \mathcal{A}$

(P2) 特別なノード $S \in H$ が存在して S から S 以外のノードには $\mathcal{A}$ のパスを通って到達可能であるが、S へは他のどのノードからも到達できない。S を START

と書く。

(P3) STARTから始まる単純列において、h_1とh_2($h_1 \neq h_2$)がこの順序で現れたとすると、STARTから始まり、h_1, h_2を含む全ての単純列においてh_1がh_2より先に現われる。

補題1 HをPOGとする。STARTから始まる単純列においてh_1, h_2がこの順序にあらわれる時に$h_1 > h_2$とかき$h_1 > h_2$又は$h_1 = h_2$の時$h_1 \geq h_2$と書くことにすれば、$\geq$は半順序になる。

補題2 HをPOGとする。$h(1) \to \cdots \to h(m) \to h(1)$を$H$のループとし$h(i) \neq h(j)$ $(i \neq j)$とすると、$h(1), \cdots, h(m)$はsimply orderである。又、$h \to k$であったとすると$h > k$又は$h < k$のいづれか一方が成り立つ。$h > k$の時$h \to k$を下向き、$h < k$の時上向きと呼ぶ。

補題3 HをPOGとする。$h_1 \mid h_2$ ($h_1 \geq h_2$, $h_2 \geq h_1$のどちらも成りたたないことを示す)でh_1からh_2にいたるHの列、$h_1 = h(0) \to \cdots \to h(m) = h_2$が存在したとする。ある$0 < k < m$について、$h(k) > h_1$かつ$h(k) > h_2$が成り立つ。

Hをダイグラフとする。Hのノードh, kについ

て $n \overset{*}{\to} k$ により n から k へ到達できるか又は $n = k$ なることを示す。$n \overset{*}{\to} k$ かつ $k \overset{*}{\to} n$ なるとき $n \overset{*}{\leftrightarrow} k$ と書く。$\overset{*}{\leftrightarrow}$ は同値関係になる。M のノードの集合 H の $\overset{*}{\leftrightarrow}$ による同値類で n を含むものを $C(n)$ で表わし n の連結成分と呼ぶ。

補題4　M を POG とする。任意のノード n について $C(n)$ は最大ノードをもつ。又、$n' \to n$, $n' \notin C(n)$, となるのは n が $C(n)$ の最大ノードの時に限られる。

§2. 平行動作

この節では M のノードを実行命令、M を実行順序を指定した指図書とみなす。まず M の頂点分離（Vertex Separator）の概念を定義する。M の V.S. とは M の全ての矢の上に白丸をおいたものであり、M_0 で示す。この定義は非常に直観的であるが分り易いし、ここでは十分厳密性を保てる。勿論厳密な定義は可能であるが非生産的に思えるのでこの直観的な定義を採用する。矢 $n \to k$ 上の白丸を n_0k で示す。M のノードの実行される順序を定義するのに M_0 にポインターの概念を導入する。ポインターは最初は START から出ている全ての白丸 s_0n の上に分離して位置する。一般にポインターが白丸 h_0k の上にあり、更に k に入っている〝必要な矢〟の上の白丸にポインターがそろうと、k が実行される。すると k に入って

いる矢の上の白丸のポインターは全てなくなり、kから出ている全ての矢の上の白丸に分離して位置する。正確に述べると次の様になる。

(EP1): はじめはポインターはSTARTから出ている全ての白丸の上にのみ存在する。

(EP2): ポインターがhok上にあり、kからでている矢がない時にはそのポインターはhokで停止する。

(EP3): kから出ている矢があるものとする。kに入っている矢のうち以下で述べる"必要な矢"の上の白丸にポインターがそろうと、kに入っている矢の上の白丸のポインターは全てなくなり、kから出ている全ての矢の上の白丸に分離して位置する。必要な矢とは:

(a) kに入っている矢がh以外にないときは $h \to k$

(b) kからでている矢の上の白丸にまだ一度もポインターがいたことがない時は、全ての下向きの $h \to k$

(c) kにすでにポインターがいたことがある時は全ての上向きの$h \to k$と、下向きの$h \to k$のうちである$h' \geq h$が存在してh'が上向きの

矢を受けているもの。

補足　hに入っている必要な矢の上の白丸にポインターがそろうと、hで表わされている実行命令が実行されるが、実行時間は未知であり、いつ終わるか予測することはしない。ただいつかは終わり、終わったならばhに入っている矢の上のポインターがhから出ている矢の上に移動する。H上のループ$h(1)\to h(2)\to\cdots\to h(m)\to h(1)$で$h(i)\neq h(j)$ $(i\neq j)$なるものを単純ループと呼ぶ。H上のパス$h(1)\to\cdots h(m)$に対してH_0上のパス$h(1)\circ h(2),\cdots,h(m-1)\circ h(m)$が対応する。もとのパスが単純列の時単純列、単純ループの時単純ループという。又簡単の為にポインターが"$h\circ k$上にいたことがある"というかわりに、"ポインターがhにいたことがある"という。hに入っている必要な矢の上の白丸にポインターがそろっている時、hがexciteしているという。同時にいくつものノードがexciteしうる。従って現在のポインターの位置だけでなくノードの実行における相対的な速さも又次のポインターの位置に影響する。我々はノード実行の速さについては知らないわけであるから、起こりうるすべてのポインターの動作を考えることになる。あるポインターの動作にお

いて、hokの上にポインターが存在し、kから矢がでているのにそこにいつづけたままでいる時、kをillegal halting nodeといい、全てのkがそうでない時にHにillegal haltは生じないという。又あるポインターの動作において、hok上にポインターがあり、同時にkがexciteしている時kにmulti callが生じたという。全てのノードkにmulti callが生じない時にHにmulti callが生じないという。POG Hが次の条件をみたす時TaillessPOG (TPOG)という。(T) H上のループ $k(1)\to\cdots\to k(m)\to k(1)$ とその一員 $k(i)$ が存在して $k(i)\to k$ ならば $k \overset{*}{\to} k(i)$ である。

定理1. HをPOGとすると、Hにillegal haltingは生じない。又Hにmulti callの生じない必十条件はHがTPOGなることである。

§3. TPOGの蓄積グラフ

$H=\{h_1,\cdots,h_n\}$ とし、HをH'$=\{{}_{a_1}h_1,\cdots,{}_{a_n}h_n\}$ をノードの集合としてもつTPOGとする。Zを非負整数の集合 $\{0,1,2,\cdots\}$ とし Z^H によりHからZへの写像の全体を示す。$a\in Z^H$, $h_i\in H$ について $a(h_i)\in Z$ を a_i で示す。aをポインターの蓄積実行状態(あるいは簡単に状態)といい、ノード h_i a_i回実行されたことを示す。ポインターがSTARTにい

る時は $a=(0,\cdots,0)$ であり、ポインターの移動とともにノードが実行されるごとに対応する成分がカウントアップされていく。ノード h_i の連結成分が2つ以上のノードからなる時 h_i をループエリメントという。$h_i \gg h_j$ により $h_i > h_j$ かつ $h_i > h > h_j$ となる h が存在しないことを示す。

補題5　M をTPOGとし、ポインターが M 上を移動していくとする。ある時点においてポインターの蓄積実行状態が a であったとする。ノード h_j が excite している為の必十条件は次の(イ)又は(ロ)が成り立つ時である。

(イ)　$a_j = 0$ の時　全ての $h_i \gg h_j$ について

$$a_i = 1$$

(ロ)　$a_j \geq 1$ の時　h_j はループエレメントであって全ての $h_i \gg h_j$ なるループエレメント h_i に対して

$$a_i = a_j + 1$$

又全ての上向きの $h_i \to h_j$ について

$$a_i = a_j$$

記号　状態 a で excite しているノードの集合を $E(a)$ で示す。　$a' \in Z^H$ を

$$a'_i = a_i \qquad h_i \notin E(a)$$
$$= a_i + 1 \qquad h_i \in E(a)$$

と a' を定義すると、a' も又蓄積状態である。a' を a

の目標状態という。さて $h_i \in E(a)$ であり、状態が a から b に移り $a_i = b_i$ であったとする。すなわち、ノード h_i は a で excite していたが、他のノードが先に実行されて状態が b に変わった時を考える。ポインターは消えてしまうことはないから、ノード h_i は b でも excite している。すなわち $b_i = a_i$ である。これは蓄積状態の集合（今後 V で示す）が Semimodulo になることを示している。実際には次の補題が成り立つ。

補題6. H を TPOG とし、ポインターが H 上を移動していくとする。ポインターの蓄積状態の集合を V で示す。V は distributive になる。すなわち次の条件をともに満足する。

(1) $(0, \cdots, 0) \in V$

(2) $a, b \in V$ ならば $a \vee b \in V$

(3) $a, b \in V$ ならば $a \wedge b \in V$

(4) $a, b \in V$ で $a \ll b$ ならば、ある $i \in H$ について $b = a + \delta^i$ となる。

ここで $a \vee b$, $a \wedge b$ は次のように定義される。Z^H の点である

$$(a \vee b)_i = \max\{a_i, b_i\}$$

$$(a \wedge b)_i = \min\{a_i, b_i\}$$

定義　Z^Hの点a, bと自然数Pについて、

$a_i \equiv b_i \pmod P$ for all $i \in H$ となるとき $a \equiv_P b$ と書く。Z^Hの部分集合VとVの点aについて、

$Va = \{ b - a , b \in V \}$ と定義する。

Vが次の性質をもつときp進ディジタルという。

(pd)：　$a, b \in V$ で $a \equiv_P b$ ならば $Va = Vb$.

定理2　MをTPOGとしその蓄積状態の集合をVとする。Vは2進ディジタルなディストリビューティブ集合である。

附記　定理2のVは実際は非常に特殊なディストリビューティブ集合になっている。それにもかかわらずTPOGはイリヤック II で使用された指図書を、特殊な一部のものを除いて、記述できる。定理2よりTPOG Mを実現する論理回路の存在が帰結されるが、より簡潔な回路の実現をオートマトンを利用する観点から考察中である。