6X-5

より現実システムに近い非決定性モデルについて

岩間一雄 (京都産大)

1. まえがき

非決定性TM (NTM) やオルタネートTM (ATM) を純粋に数学モデルとする考え方もあるが，木構造でプロセスを生成していく並行システムのモデルとする見方もある[1,2]. その場合のモデルの最大の特色は，生成された数多くのプロセスはいったん生成されたあとは一切相互干渉を持たないという点であろう. このような少ない通信量で，ATMの場合，T(n)時間でおよそT(n)領域量の直列計算を模擬できるという並行システムの基準ともいえる性質を有しており，モデルとしての優秀さは広く認められている. しかしこの特色は，別の (より現実的な) 観点からは，並行システムとしては強すぎる制限にもなり，特に低いレベルの複雑さの具体的問題に対するアルゴリズム設計を考えるときは (もともとTM自身がそのような場合にあまりなじまない点を割引いて考えても) 大きな障害になっているようでもある. 以上の動機から，本稿では，ATMやNTMにおいてプロセス間通信を許すようなモデルの拡張を考える. 通信機能としては最も原始的と考えられるバス通信をとりあげる. さらには，ATMから離れ，そのようなバス通信自体の並行システムにおける貢献についても考察する.

2. バス通信を許すATM

ATM (又はNTM) で生成されるすべてのプロセスが1ビットバスを通して通信できるモデルを考える. すべてのプロセスは計算の各ステップで0又は1をバスに送出し，その結果 (バスの状態) はその次のステップですべてのプロセスから参照できる. バスと各プロセスの接続はいわゆるwired-orを仮定する. 即ち1つでも1を送出するプロセスが存在すればバスの状態は1になる. 状態遷移関数は，入力テープ記号，ワークテープ記号，内部状態，バス状態の組から，ワークテープ記号 (書き換えの為の)，ヘッドの動き，次内部状態，バスへの送出記号の組を決める. 様相は通常の内容 (状態，ヘッドの位置，ワークテープの内容) の他にバス状態が加わる. 時刻 t のバス状態 $B(t)$ は時刻 $t-1$ において少くとも1つのプロセスが1を送出すると1になる (すべてが0なら0). 受理条件は，通常のATM (又はNTM) のそれを用いる.

以上の様に増強されたATM (NTM) をATMSB (NTMSB) と呼ぶことにする. 直観的には，各プロセス間の同期をとることが可能になった. [3] でATMの非正規言語に対する領域量の下限が $\log\log n$ であることを示したが，ATMSBでは領域量 $\log\log\log n$ で受理できる非正規言語 ([4] におけるDBLBIN) が存在することが証明できる. さらに，その為にはATMの受理条件は本質的でなくNTMのそれでも十分なこと，さらに領域量の定義に関してもすべてのプロセスに関して最悪の場合をとるという最も強い (実際的な) ものでも結果に変化が出ないことが示せる.

On Bus-Connected Parallel Models

Kazuo Iwama
Kyoto Sangyo University

［定理1］ オンラインNTMSBによって領域量$O(\log\log\log n)$で受理できる非正規言語が存在する．(証明略)

3. バス通信自体の能力

前節で述べたように，バス通信の能力はたとえ1ビットといえどもある局面ではATMの(巧妙な)受理条件を必要としなくなる程大きい．これはバス通信が自然にOR演算を実現することも大きな要素になっている．この現象は共通記憶を通じて通信するPRAMモデルにおいてもみられ，同一アドレスに対する複数個のプロセスによる同時書込みを許すか(必然的に何らかの演算が導入されなければならないが著者が知っているすべての場合において書き込んだ内容ではなくプロセスの番号に関するルールを採用している)許さないかによって能力が大きく異ってくることが多くの論文で論じられている(e.g. [5])。

1ビットバスは自然にOR演算を実現するが，それをkビットバスに拡張する場合の演算の自然な拡張は最大値演算であろう．つまり各プロセスはkビットの2進数を送出し，バスの状態はその中の最大値になるというルールで，回路的実現も容易である：バスとの接続部のドライバの入力iと出力o（バスの状態）を比較し，$i=0$かつ$o=1$ならそのドライバも含めそれより下の桁のドライバの出力を高インピダンス（又はドライバの入力を強制的に0）にすればよい．複数個のRAMをこのようなkビットバスで接続したシステムをPRAMAXと呼ぶことにする（kはRAMの命令で扱われる1ワードのビット数以上と仮定）．これはかなり現実性の高いシステムであり，1ステップ動作でプロセサ個数までの数の最大値が得られるという利点を有す．以下では一例として，n節点のminimum spanning tree (MST)をnプロセッサ，$O(n)$時間で求めるアルゴリズムを与える．なお，本モデルは通信幅1のPRAMで同時書込みは最大値を書込んだプロセッサが優先されるというモデルと同等であるが，著者が知る限りにおいて過去にはとり上げられていない．又，MSTに対するnプロセッサ，$O(n)$時間のアルゴリズムも過去には知られていない．

入力としてはプロセサi ($P(i)$)に接点iと隣接するすべての節点jと枝(i,j)の重みを与える．Primのアルゴリズム（$O(n^2)$直列時間）を用いる．詳細は略すが，次の各処理が1ステップで実行できることが基本になる．(i) $P(i)$はiをバスに流してiと隣接する接点$j_1, j_2, \cdots$に対応する$P(j_1), P(j_2), \cdots$をアラートする．(ii) $P(j_1), P(j_2), \cdots$はそのステップまでに構築された木Tと自分自身の節点をむすぶ最小重みの枝をみつける．(iii) $P(j_1), P(j_2), \cdots$は(ii)で得た枝の重み 補数をバスに同時に流すことによりTにつながる枝の最小重みを知る．(iv)その重みの枝が2つ以上ある場合はプロセサ番号最大のものの担当枝をとりその端点をTに加える．なお，本アルゴリズムはnより少ないプロセサ数に対しても適用できる。

［定理2］ n節点のMSTはPRAMAXによりn/pプロセッサ，$O(np)$時間でもとめることができる（$p \geq 1$）．

MSTに対しては，n^2プロセサ，$O(\log^2 n)$時間等のアルゴリズムが数多く知られているが，モデル（無制限の通信幅）やプロセサ数の点で本アルゴリズムの方がより現実的といえよう。さらに本手法は最短経路問題等にもほとんどそのまま応用でき，さらに他のいくつかのn^2直列時間アルゴリズムの本原理による並行化も可能である．

文献

[1] Chandra他 JACM 28,1 [2] Savitch他，JACM 26,1
[3] Iwama, Res. Rept. KSU/ICS 86-01. [4] —, KSU/ICS 86-02 [5] Fich他 85 STOC.