

談話室



データ・フロー計算機／関数型プログラミング へのコメント†

米澤 明 憲††

現在日本の各界で、データ・フロー計算機に対する関心が高まっているが、データ・フローに基づく“計算概念”に対する誤解あるいは拡大解釈によってデータ・フロー計算機に関する議論の一部に混乱をきたしているように見受けられる。以下この混乱を軽減するために手短かに問題点を述べ、あわせてデータ・フロー計算機の応用分野を再確認しておきたい。

データ・フロー方式の計算について、その研究発展の経緯に触れずに現在明らかになっている点のみに注目して考えてみると、大雑把に言って次の2点に要約される。(1)インストラクションは、それによって処理されるべきオペランド(データ)の準備が完了した時点で直ちに実行される。このため従来のプログラム・カウンタによる順次的な計算ではなく、実行可能となっている複数のインストラクションは並列的に実行してよい。(2)この計算方式に基づいて許される並列性を最大限に引き出し有効に利用するためには、従来のプログラミング言語で日常的に用いられる代入文による変数の値の変化に基づく算法表現でない、例えば関数形式に基づくようなプログラミングスタイルを取らねばならない。

(2)で述べられている“並列性の最大限利用”のための要諦は、『データ・フロー方式で計算を行う場合、そこで処理される対象物(データ)は時間的に変化する内部状態を持つことは許されない』ということを含意する。すると履歴依存(history sensitive)性を持つ情報処理操作をデータ・フロー方式で行うためには、従来対象物の状態の変化によって表現していたことを、状態が変化しない新たな対象物を次々に創生してゆき、その創生の順序を何らかの形で記憶してゆくことによって表現しなければならなくなる。これは多量のデータの記憶更新操作を必要とする応用分野ではデータ・フロー方式を採用することに妥当性が薄いことを意味する。

(2)から起因する上の事実は多くの研究者によって意識はされているが、データ駆動的であるという(1)

の点だけに直観的理解が集中するために、広範囲の問題がデータ・フロー／関数形式的な計算方式によって有効に対処できるという誤解や幻想(?)が払拭されていないようである。またこの誤解に基づいて汎用のデータ・フロー計算機が実用的な文脈の中で実現可能だという論議も散見される。これにははなはだ疑問が多い。

もし汎用のデータ・フロー計算機が実用的なものとして実現されるならば、それは(1)で示唆されている点、すなわちデータ(あるいはメッセージ)の到着によって情報処理操作が駆動されるという点だけに依存するシステムであり、(2)に反して対象物の状態変化を何らかの形で許すものであろう。

関数形式によるプログラミング・スタイルによってどれだけの算法や情報処理操作が実的に表現可能かという問題に対して、我々は現在の研究段階では何の見通しも得ていないという事実も認識されなければならない。問題が表現できてもその表現に従った計算方式が実用的であるという保証はまったくない。さらに、関数形式で書かれた算法表現をマシンによって直接実行するのにデータ・フロー計算機が適しているという保証もまったくない。様々な形式について研究し経験を深めなければならない。

最後にデータ・フロー計算機と関数型プログラミングの双方に深く関連し、かつ従来の計算機機構からあまりかけ離れていない『GCマシン』(ガーベージ・コレクション・マシン)というものを提案しておきたい。関数型プログラミングやデータ・フローによる算法表現を従来の計算機に近いものの上で実行するとき、データの創生複製が日常的に極端な頻度でおこる。すると記憶容量の不足はどんなに記憶素子が安価になっても深刻な問題となるので、ガーベージコレクションが最も重要な機能となる。GCマシンはこれを行うのに適した機構を有した計算機システムである。このマシンはもし実現すればCLU等の対象指向型の言語の処理系作成に極めて有用であることも論を待たない。以上計算機アーキテクチャについての一素人の放言である。ご批判、ご教示を乞う。

(昭和55年2月26日受付)

† Comments on Dataflow Machines and Functional Programming Machines by Akinori Yonezawa (Department of Information Science, Tokyo Institute of Technology).

†† 東京工業大学理学部情報科学科