

# 生産順序キューマシン命令コード生成手法の提案

川島 祐介 繁田 聡一 B. A. Abderazek 吉永 努 曾和 将容  
電気通信大学 大学院情報システム学研究科

## 1 はじめに

我々は、式評価の中間結果を LIFO (Last in first out) メモリに貯えるキューマシンの研究を行っている。キューマシンは命令に対しオペランドの指定が必要ないためプログラムサイズが小さくなる。また、隣接する命令間に依存関係が発生しにくいいため、命令レベルでの高い並列性が期待できる。しかし、キューから取り出して使用するというキューの特性から、式評価の中間結果を複数回使用するプログラムには不向きである。そこで、我々は参照の概念を含めることでキュー計算方法の制約を緩和する、計算モデルを提案している。その1つが「生産順序キューマシン」である。生産順序キューマシンとは、生産された順序に中間結果をキューに格納し、過去に生産された中間結果を参照することで複数回使用できるようにしたキューマシンである。本稿では生産順序キューマシン用の命令コードを生成する手法を提案する。

## 2 キューマシン

キュー計算モデルとは、演算における式評価の中間結果を FIFO の記憶装置で保持する計算モデルである [1]。この計算モデルの命令を生成するためには、構文木を幅優先で探索し、命令を出力する (図 1)。このキュー計算モデルによって駆動する計算機がキューマシンである。キューマシンでは、全ての演算結果をキューへ格納し終えた結果が正しい順序に並んでいれば、キューの中のデータに対する演算を任意の順序で行えるので、キュー内に常に十分なデータがあれば、CPU の各機能ユニットは待ち状態に陥ることなく、効率のよい並列処理を行うことが期待できる [2]。

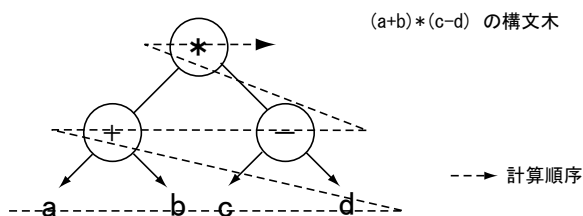


図 1: キュー計算モデルの計算方法

## 2.1 生産消費順序遵守型キューマシン

生産消費順序遵守型キュー計算方法 (以降「生産消費型」と呼ぶ) とは、データの生産順と消費順が Queue Register の並び順と一致するプログラムによる計算方法であり、純粋な Queue 計算方法と一致する。

生産消費型の場合、データに対する生産者と消費者が 1 対 1 の関係になる。一度生産されたデータは一度しか消費できず、複数回消費することができない。これは、同一のデータに対し複数の消費者がいる場合は、データの複製処理が必要なことを意味し、この複製処理はオーバーヘッドになる。

## 2.2 生産順序遵守型キューマシン

キュー計算モデルの条件を緩和し、参照という概念を取り入れた生産順序型キュー計算モデル (以降「生産型」と呼ぶ) を提案している [3]。参照とは、必要なデータを使用するときに、Queue Register のデータの並び順とは無関係にデータを取り出す方法である。参照によるキューの変化の例を図 2 に示す。図 2 において、生産と消費生産型は、データの生産順序を優先して Queue Register に格納する方法で、消費順序が生産順序と一致しないことを許す。

生産型において、参照はデータを消費しない為、データの複数回利用が可能になる。

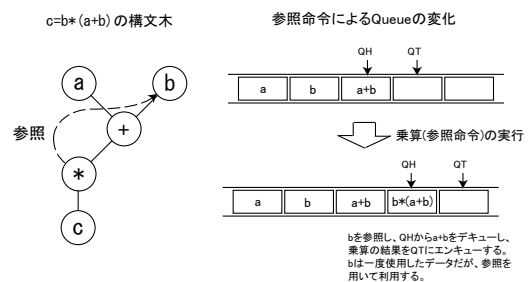


図 2: 参照命令の例

## 3 生産順序型キューマシン命令コードの生成手法

本稿では、生産消費型の命令コードを生産型の命令コードに変換する手法を提案する。生産型キューマシン命令の生成は、生産消費型の構文木の変換をもとに行う。

生産型コードに変更することによって、データの複製を抑制できる。その結果、プログラムサイズの削減、実行時間の短縮を図ることができる。これはデータを複数回使用する生産消費型キュー計算モデルに不向きなプログラムに対して特に有効である。

Proposal of Code Generation Scheme for a Producer-order Queue Machine

Yusuke Kawashima, Soichi Sigeta, B. A. Abderazek, Tsutomu Yoshinaga, and Masahiro Sowa

The Graduate School of Information Systems, University of Electro-communications

### 3.1 参照命令への変更

同一のデータを生成している複数の命令がある場合、参照を利用した命令に変更することで同一データを生成している命令を削除できる。例えば図3のような場合、加算を参照命令に変更することで、2回目のaのloadを削除できる。

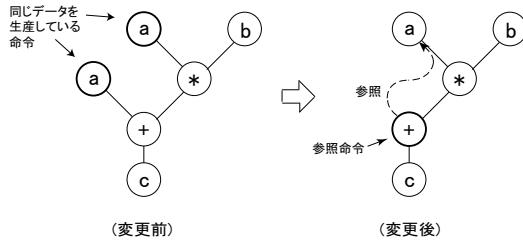


図 3: 参照命令への変更

ほかの変更機会として図4のようなものがある。図4-1と図4-2の場合は後者のload命令を削除し、そのload命令が生成した値を消費する命令を参照命令に変更することによって、生産型化を行うことができる。ただし、データの値が等しいことが条件なので、2つ命令間にデータの変更がある場合には参照は適用できない。また、図4-3において、duplicate命令はデータを複製する命令である。この命令は消費データと生産データは同一の値なので、参照命令を用いることが可能である。

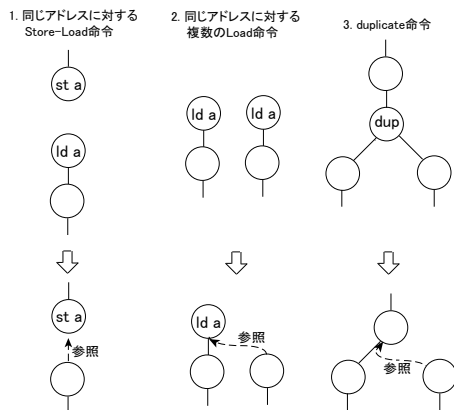


図 4: 参照命令に変換できる命令を発見できるパターン

### 3.2 参照距離の算出

参照命令はオペランドとして、参照場所を必要とし、その値はQHからの相対番地である。これを参照距離と呼ぶ。

参照距離を算出する方法は次のようになる。

まず、構文グラフのアークに番号をつける。番号は図5のように左上から右下へ1レベルずつ下がりながら左から右へと調べながらつけていく。これは、各アークの親によってデータが作成される順番であり、同時にそのデータのオペランドキューの位置と一致

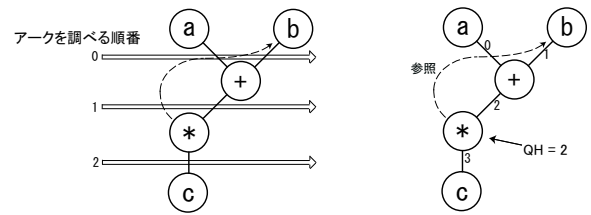


図 5: 参照距離の算出方法

する。よって、参照先のアーク番号を調べることでより、参照データのオペランドキューの位置が求まる。

次に、参照命令が実行される時のQHの位置を求める。これには、先に求めたアークの番号を利用する。入力アーク(親と接続しているアーク)が1つの場合はその入力アークの値になり、入力アークが複数ある場合は番号一番若い値になる。入力アークがない場合は、実行順番的をさかのぼり、一番最後にある現れる入力アークの値に1を足したものになる。

この2つの値の差分をとることで、参照距離を求めることができる。

### 3.3 LQHの操作

参照型キューマシン命令では、一度使用したデータ、つまりQHより前のデータも保持する必要がある。しかし、不要なデータを捨てないとキューが溢れてしまうため、LQH(Live Queue Head)という区切りを設け、LQHまでのデータを保持することで、キューの溢れを防ぐ。生産型では、このLQHを操作する命令が追加される。

LQHを細かく修正すると、命令数が増える。そこで、LQHとQHの距離が間隔をハードウェア機構により一定に保つ命令「AutoLQH」を用いる。QHの動きに合わせてLQHも移動するため、LQHの明示的な操作が必要なくなる。

## 4 まとめ

本稿では、生産消費型の命令コードを、生産型の命令コードを生成する手法を提案した。提案手法の実装と検証が今後の課題である。

## 参考文献

- [1] 前田敦司, 中西正和, "新しい計算モデル キューマシンとその並列関数言語への応用", 情報処理学会論文誌, vol.38, pp.574-583, (1997).
- [2] Ben A. ABDERAZEK, 繁田 聡一, Kirilka NIKOLOVA, 吉永努, 曾和将容, "並列キュープロセッサの基本設計", 電子情報通信学会, 信学技法(CPSY) vol.102, No.478, pp.55-60, (2002).
- [3] A. Markovskij, M. Sowa, S. Shigeta, and B. A. Abderazek, "Producer-order Parallel Queue Processor Architecture Design", 信学技報(CPSY) [掲載予定] (2004).