

TM-0330他

情報処理学会 第35回全国大会 論文集4-1
(並列推論システム)

©1987. ICOT

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

TM-0330	MRBによる多重参照管理方式－KLI処理系における実時間ガーベジコレクション方式－	木村康則, 近山 隆
TM-0331	PIMの並列キャッシュ－KLIのメモリ参照特性－	松本 明, 中川貴之, 佐藤正俊, 清水 肇 後藤厚宏
TM-0332	PIMの並列キャッシュ－共有バスの一構成法－	中川貴之, 松本 明, 後藤厚宏
TM-0333	KLIの並列処理－蜜結合マルチプロセッサでの並列処理系の評価－	佐藤正俊, 清水 肇, 後藤厚宏
TM-0334	MRBによる多重参照管理方式－KLI処理系における一括型GCの特性評価－	宮内信仁(三菱), 久門耕一(富士通), 木村康則, 近山 隆
TM-0335	小型化版CHIのSUPLOGコンパイラにおけるOR節のコンパイル方式	新 淳, 小長谷明彦, 梅村 護(日電)
TM-0336	ポインタタグ(MRB)による多重参照管理方式	近山 隆, 木村康則
TM-0337	MRBによる多重参照管理方式－KLI処理系におけるキャッシュ特性の評価－	西田健次, 松本 明, 木村康則, 後藤厚宏
TM-0338	並列論理型プログラムの特性解析(1) －動特性解析ツール－	古市昌一, 石塚裕一(三菱), 宮崎敏彦, 滝 和男, 杉野栄二
TM-0339	PSIビットマップ・ディスプレイ装置のカラー化に対するオペレーティング・システムの対応	高橋雄二, 福島克己(三菱), 中澤 修, 寺崎理美子(沖), 近山 隆
TM-0346	マルチPSI第2版のPE間アドレス管理方式	六沢一昭, 市吉伸行, 宮崎敏彦, 近山 隆 滝 和男
TM-0348	並列論理型OS-PIMOS(2) 開発支援KLI処理系	宮崎敏彦, 木村康則
TM-0349	並列論理プログラムの特性解析(2) －小規模プログラムの解析例－	和田久美子(沖電気), 市吉伸行, 寿崎かすみ, 滝 和男

MRBによる多重参照管理方式

20-6

- K L 1 処理系における実時間ガーベジコレクション方式 -

木村 康則 近山 隆

(財) 新世代コンピュータ技術開発機構

1. はじめに

K L 1 処理系において変数セルや構造体データをMRB管理[1]することの一つの長所として、実時間ガーベジコレクションがある。

K L 1 は、副作用を持たない言語であり、実行はヒープ領域への変数セルの割り当て、具体化、参照を繰り返すことによって進む。従って、何らかの方法でヒープ領域の管理を行わないと、'使い捨て'に近い形で領域が消費され、ガーベジコレクションが頻発してしまう。一方、筆者らの予備実験から、具体化した変数セルの参照は1回だけの場合が多いと予想される。そこで、K L 1 の変数セルやデータをMRBで管理し、不要になった時点で積極的に回収することにより実時間ガーベジコレクションが可能となる。

本稿では、逐次版K L 1 処理系に組み込んだ、MRBを用いた実時間GC方式と評価結果について報告する。

2. MRBを用いたGC方式

K L 1 のデータをMRB管理すると、K L 1 の実行中にあるデータへの参照が最後でかつMRBがオフであればそのデータは回収し、再利用できることになる。具体的には、以下の場合にメモリの回収が可能となる。

(1) 変数のデレファレンス

変数のデレファレンス時の間接ポインタは、その間接ポインタセルへの参照が単一で（ポインタMRBはオフ）、この間接ポインタのMRBがオフであれば（セルMRBがオフ）、デレファレンス時に回収できる（図1）。

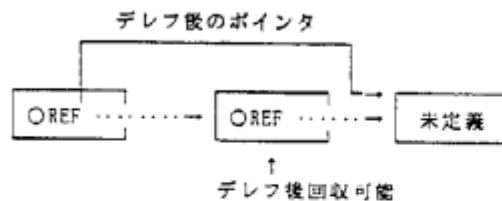


図1. 変数のデレファレンス

(2) 構造体のユニフィケーション

受動部で、ゴールの引数の一つが構造体との単一化に成功したとき、ゴール引数として与えられた構造体はMRBがオフであれば回収できる。例えば以下の様なクローズがあったとき第一引数として与えられたコンスセルはMRBがオフであれば回収できる。

$$p([X|Y]) :- \text{true} \mid b(X), c(Y).$$

(3) ポイド変数とのユニフィケーション

ゴール引数がポイド変数とユニファイしたとき、そのゴール引数のMRBがオフであれば回収できる。

$$p(\text{foo}, X) :- \text{true} \mid \text{true}.$$

(4) 回収される構造体の要素

ある構造体が回収される時、その構造体の要素も各々のMRBがオフであれば回収してよい。例えば、以下のクローズで、第一、第二引数とも同じ構造体を与えられ、構造体同士のユニフィケーションが行われたとき、その構造体は回収され、さらに、要素も再帰的にたぐって（MRBがオフであれば）回収できる。

$$p(X, Y) :- X = Y \mid \text{true}.$$

3. 抽象マシンと命令セット

本方式を実現するためには、先に提案した抽象マシンと命令セット[2]を変更、改良する必要がある。本章では、主に変更点について説明する。

(1) 抽象マシン

まず、MRBのフィールドを各K L 1 データセルに設けなければならない。これにはタグフィールドの1ビットを充てることとした。次に、変数セル、コンスセルのためのヒープ領域は、それぞれフリーリストにより管理することにした。また、前章で示したガーベジセルの回収は、そのクローズが選択されるとわかってから行わなければならない。そこで、受動部の実行中に上記(4)で示した回収可能な構造体要素を一時的に保持しておくためのスタックを新たに導入した（これをガーベジスタックと呼ぶ）。

(2) 命令セット

MRBを正しく管理するために個々の命令の仕様を変更し、新たに幾つかの命令を導入した。具体的には、受動部に現れる変数についてコンパイラでその最終出現をチェックし、最終出現のときには回収用の命令を出すことと、能動部で変数が共有されている時にMRBをオンにすることである。

Multiple Reference Management by MRB - Real Time GC -

Yasunori KINURA and Takashi CHIKAYAMA

Institute for New Generation Computer Technology(ICOT)

以下では、第2章で説明した領域回収を行うための命令のみを説明する。これらの命令は、選択されたクローズの運動部の命令群の前に生成される。なお、現在の命令セットはデレファレンス命令を隔に持っていないので、デレファレンス時の回収は個々の命令中で行われる。

- o collect_list Ai — Ai から指されるリストのMRBがオフであればリストセルを回収する（前章(2)に対応）
- o collect_value Ai — Ai から指されるデータのMRBがオフである限り再帰的に回収する（前章(3)に対応）
- o collect_stack — ガーベジスタックから指されているデータを回収する。（前章(4)に対応）

図2にKL1プログラムとコンパイル例を示す。

```
p([X|Y], X) :- true | q(X).
```

```
p/2: wait_list A1          % 第一引数はリスト?
      read_variable X3      % carを読む
      read_variable X4      % cdrを読む
      wait_reused_value X3, A2 % 第二引数の単一化
      collect_list A1       % リストの回収
      collect_value X4       % ボイド変数の回収
      collect_stack         % ガーベジスタックの回収
      put_value X3, A1
      execute q/1
```

図2. KL1プログラムとコンパイル例

4. 評価

実時間GCの効果を調べるために、汎用計算機上にC言語でエミュレータを作成し、のべメモリ使用量(TOTAL)、使用メモリ領域の大きさ(PK), 実行後残ったゴミの量(LEFT)の計測を行った(図3)。用いたテストプログラムは、素数生成(PRIME)、8クイーン、ボトムアップパーザ(BUP)である。素数生成の様な素直なストリーム通信を行うものについては、実時間GCをやらない場合(TOTAL)と比較して、15%程度のメモリ量(PK)で実行でき、実行後には、メモリは、すべて回収されている。8クイーン、BUPでは、それぞれ6割、3割程度のメモリ量で実行でき、実行後には、5割、2割のメモリ量(LEFT)がゴミとして残っていた。以上、プログラムにより回収率に差があるものの実時間GCの効果は大きいと言える。

また、図4に全回収データの命令別の回収比率を示す。まず、各命令に分散しているデレファレンス時の回収が多い。これは、具体化した変数は一つの参照パスしか持たないことが多いことを示していると思われる。命令別では、collect_list命令による回収が多い。これは、KL1でのゴール間のデータの受け渡しはリストを用いたストリーム通信によって行われていることによるものと考えられる。一方、collect_stack命令では全く回収できていない。これは、KL1のユニフィケーションが単方向であり、受動部では、複雑なユニフィケーションがあまり行われていないことを示していると思われる。

5. まとめと今後の課題

MRBを用いた実時間GC方式について報告した。テストプログラムを用いた実験では、十分なGC効率を期待できることがわかった。今後、MRBを用いた実時間GCを並列マシンに実装するにあたって検討すべき項目をまとめると以下の様になる。

- (1) 現在は回収用の命令を隔に出しているの、動的な命令実行数が増えている。テストプログラムでは約2割の増加となっているが、実行命令数をおさえるような命令体系の検討が必要である。
- (2) 個々の命令自身がMRB管理のために操作が重くなる。MRB管理を高速に行うハードウェアの検討が必要である。
- (3) 変数セル、コンスセルをフリーリスト管理することは実行時のオーバーヘッドが大きい。MRB方式に適したメモリ領域の高速な獲得、解放機構の検討が必要である。

謝辞

日頃御指導頂くICOT4研、内田俊一室長に感謝します。実時間GC方式の実装、評価に関し、貴重なコメントを下さった後藤厚宏、宮内信仁、西田健次の各氏に感謝します。

参考文献

- [1] T.Chikayama and Y.Kimura: "Multiple Reference Management in Flat GC", Proc. of ICLP'87
- [2] Y.Kimura and T.Chikayama: "An Abstract KL1 Machine and its Instruction Set", Proc. of SLP'87

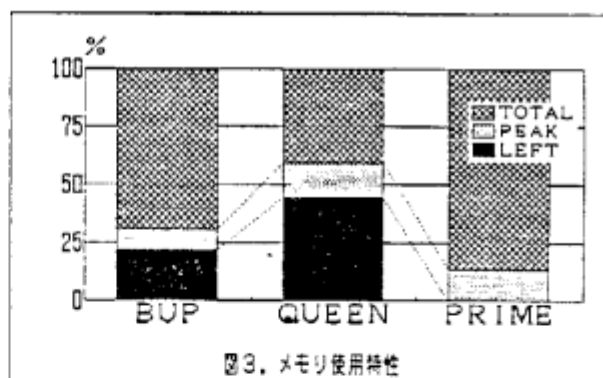


図3. メモリ使用特性

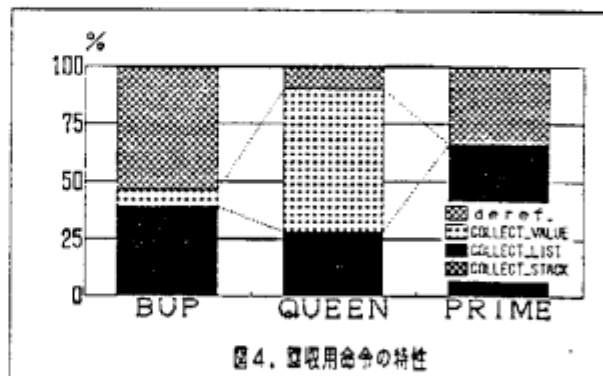


図4. 回収命令の特性