LRCによる多重参照管理方式

後藤 厚宏　　近山 隆
(新世代コンピュータ技術開発機構)

1 はじめに

KL1はフラットGHCに基づく並列論理型言語であり，並列処理システムの実現にとって不可欠である同期や通信等の基本概念が自然に記述できる。ただし，KL1のようなメモリセルの破壊的書き換えを許さない並列論理型言語を単純に実装すると，メモリ領域を単調かつ急速に消費してしまう。この結果，メモリ参照の局所性が悪い一括型GCを頻発してしまい，キャッシュのミスヒットやページフォールトが多くなる。一方，ICOTにおいて開発を進めている並列推論マシンPIMのように，並列マシンによってKL1を並列処理する場合，プログラムの実行を並列に行なうだけでなく，GCも並列に実行する必要がある。このため，KL1の並列処理には，インクリメンタルGCのように，不要になったメモリ領域を各要素プロセッサが実行時に回収し再利用する機構が重要となる。

実行時のGC方式としては，各データオブジェクトにそこへの参照数を示すカウンタを設ける参照カウンタ方式が一般的である。しかし，この方式では，原理的にポインタの一語長分の参照カウンタ領域が必要であり，参照カウンタ更新のためのオーバヘッドも大きい。

これに対し，実際のプログラムの実行では，データオブジェクトへの参照数が1または2の場合が多いことを利用した方式が幾つか提案されている。

多重参照ビット(MRB: Multiple Reference Bit)方式[1, 2]では，データオブジェクトではなく，ポインタ上の1ビットのフラグ(MRB)を用いて参照カウンタに相当する情報を表現する。これによって，メモリ資源を節約し，操作を簡素化している。ただし，1ビット分のMRBでは一度多重参照になったデータオブジェクトは参照数が0となっても回収できないため，メモリの再利用という点では完全ではない。このため，MRB方式は一括型のGCと併用する必要がある。

本稿では，遅延参照カウント(LRC: Lazy Reference Count)と呼ぶGC方式を提案する。LRC方式[3]は，2ワードからなる参照カウント付き間接参照セルを用いて参照数を表現し，多重参照か単一参照かはMRBと同様のポインタタグによって識別する。これにより，MRB方式の長所を活かしたまま，MRB方式における回収の不完全さを補うことができる。

Multiple Reference Management by LRC
Atsuhiro Goto, Takashi Chikayama (ICOT)

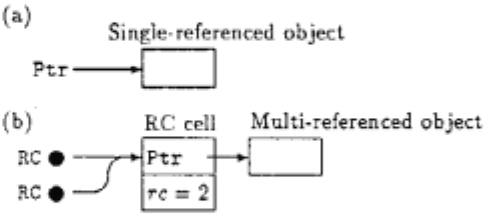

図1: RC付きセルよる多重参照

2 遅延参照カウント方式

2.1 RC付きセルによる多重参照

LRC方式では，ほとんどのデータオブジェクトが単一参照であると考え，普通のデータオブジェクトには参照カウンタ領域を持たせない。もし，多重参照されるオブジェクトが必要になったときは，その時点で，図1(b)に示すような参照カウンタを持つ2ワードの間接ポインタ(RC付きセルと呼ぶ)を利用して多重参照を表す。この結果，多重参照されるデータオブジェクトへの参照パスが合流するノードはRC付きセルのみに限られる。ここで，RC付きセルの1ワードめ(データ部)は，多重参照されている構造体へのポインタ，アトム型の値，および多重参照さている未定義変数セルであり，2ワードめ(rc部)は，そのノードの被参照数を示す。

プログラムの実行において，多重参照されているオブジェクトへの参照パスを消費したり生成したときは，その参照パスで最初に現われたRC付きセルのカウントを更新する。一方，単一参照であったオブジェクトへの参照パスを追加する場合は，参照パスの途中にRC付きセルを持たないため，新たにRC付きセルを挿入する。

2.2 多重参照ポインタ

LRC方式では，ポインタのタグによって，参照パスが合流するRC付きセルを識別する。多重参照ポインタ(RC●)はそれが直接指しているセルがRC付きセルであることを示す。つまり，それ以降のオブジェクトは多重参照されている可能性がある。一方，その他のポインタ(REF, LIST, VECT)は単一参照ポインタであり，RC付きセルでない変数セルまたは構造体データの本体を直接指す。つまり，その直ぐ先のセルは，そのポインタによってのみ参照される。ただし，先のセルが未定義変数の場合は，もう一本の参照パスがありうる(2.3参照)。

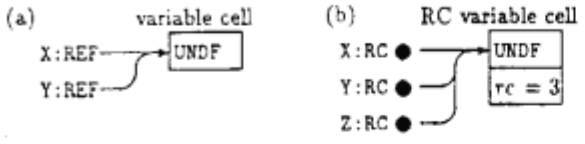

図 2: 未定義変数の表現

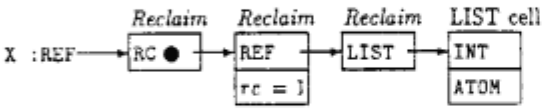

図 3: デレファレンスによる GC

2.3 変数の生成

KL1 の未定義変数には，通常，それを具体化する参照パスと具体化した値を読み出す参照パスの両者がある。そこで，参照パスが 2 以内の未定義変数セルは通常の間接参照ポインタ (REF) によって参照できることにする (図2(a))。

```
p :- true | q(X), r(X).        (1)
p :- true | q(X), r(X), s(X). (2)
```

クローズ(1)の場合は，ボディ部で新たに割り付ける未定義変数 X の出現回数が 2 なので，図2(a) のような REFのポインタによって参照される変数セルを割り当てる。一方，クローズ(2)では，3 回 (以上) 現れているため，図2(b) のような RC 付きの未定義変数セルを割り当て，RC ● で参照する。

2.4 実行時におけるメモリ領域の回収

KL1 の実行では，変数のデレファレンス，ユニフィケーション，および構造体要素の参照等の操作によってデータへの参照パスを消費した時，そのデータが多重参照されていた場合は，RC 付きセル中の参照数を減算する。一方，その参照パスが最後のもの (例えば，ポインタが単一参照ポインタ，または多重参照ポインタRC ● で参照カウントが 1 の時) であれば，そのメモリ領域は回収し再利用できる。具体的には，以下のような場合がある。

a. 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あれば，デレファレンス時に回収できる (図3)。

b. 構造体のユニフィケーション

受動部で，ゴール引数が構造体とのユニフィケーションに成功した場合，ゴール引数として与えられた構造体 (下線部) が単一参照であれば回収できる。

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

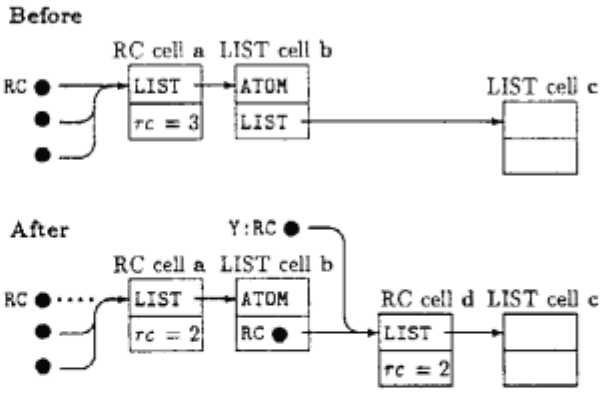

図 4: 構造体要素への参照パスの追加

一方，構造体が多重参照されていた場合，構造体から要素への参照パスがそのまま残り，要素への参照パスが一つ増える。このため，構造体要素への参照パスが単一参照であったならば，RC 付きセルの遅延割り当てを行う必要がある。例えば，図4の LIST セルbの cdr 要素 (LIST セルcへのポインタ) を Y に読み出す場合，RC 付きセルdを挿入して，元の構造体 (LIST セルb) と Y からのポインタを合流させる。

3 検討

LRC 方式では，MRB 方式と同様に，ポインタのタグによって多重参照を検出できる。このため，単一参照である限りメモリセルの回収のコストは小さく，MRB 方式と等価である。ただし，単一参照を多重参照に変換するコストは LRC 方式の方が大きい。LRC 方式のもっとも大きい利点は，回収残しがほとんど無いことである。MRB 方式によるインクリメンタル GC の評価[2] では，ベンチマークプログラムによって多少異なるが，1 回のゴールリダクション当たり 1 ワード程度の回収残しが生じる。このため，MRB 方式の GC だけでは不十分であり，一括型 GC を併用する必要がある。これに対して，LRC 方式では，参照カウントによって原理的に回収できないループ構造とメモリ空間のフラグメンテーション以外の理由で一括型 GC を必要とすることはない。

参考文献

[1] T. Chikayama, et al. Multiple Reference Management in Flat GHC. In *Proc. of the 4th ICLP*, 1987.

[2] 木村 他. KL1 の多重参照ビットによる GC 方式. データフローワークショップ, 1987.

[3] 後藤 他. 遅延参照カウントによる並列推論マシン向き実時間 GC 方式. LPC'88.