

PIMのKL1 並列処理系用 GC 方式

ICOT

第4研究室

後藤厚宏

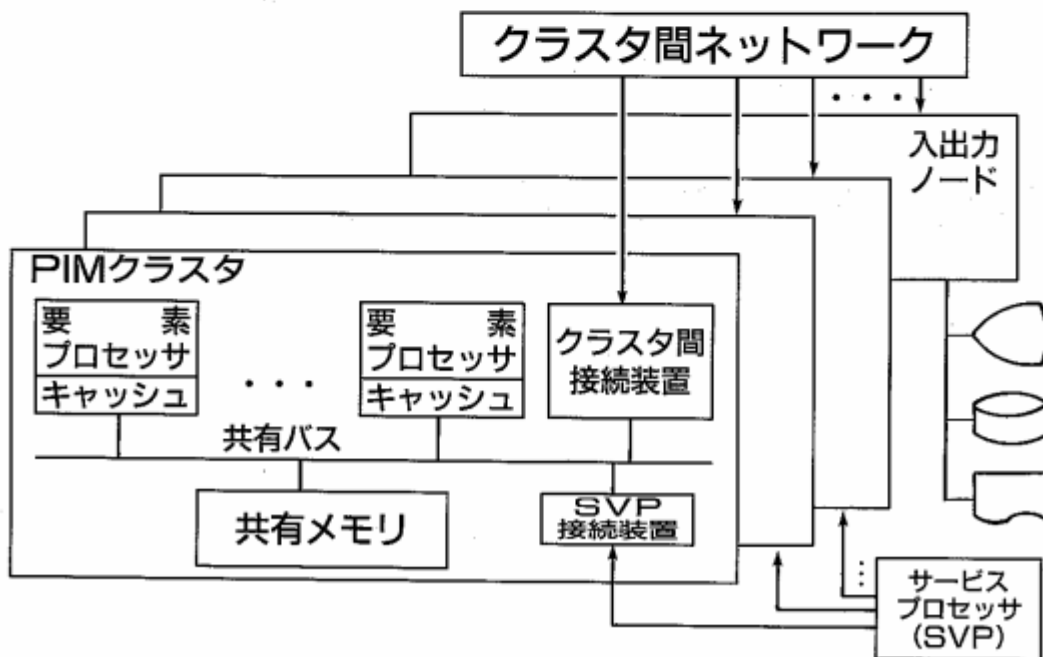
1. KL1の並列処理におけるガーベジコレクション (GC)の重要性
2. 多重参照ビット (MRB)を用いた実時間GC方式
3. シミュレーション評価
4. 実時間GC方式の応用と改良

ガーベジコレクションの重要性

- § 並列論理型言語 KL1 (≈ flat GHC)
 - ⇒ メモリセルの破壊的書き換えを許さない
 - ⇒ 単純に実装するとメモリを急速に消費する
 - ⇒ 効率の良いガーベジコレクション(GC)が必須
 - (cf.) Prologはバックトラックにおいてメモリを回収できる

- § 並列推論マシン(PIM)のGC
 - ⇒ GC操作自体の並列性, 局所性が重要
 - (cf.) 一括型のGCではメモリ参照の局所性が期待できない
 - ⇒ 効率の良いインクリメンタルGC

PIMの全体構成



インクリメンタルGC

要求: 実行時に不要になったメモリ領域を再利用したい

⇒ データオブジェクトへの参照数を実行時に管理

● 参照数カウント方式はコストが高い

- データオブジェクト毎の参照カウンタが必要
- 参照カウンタ更新のオーバーヘッドが大きい
- (循環構造が回収できない)

● 実際のプログラムの実行では, 参照数が1または2の場合が多い

⇒ ハッシュ表方式

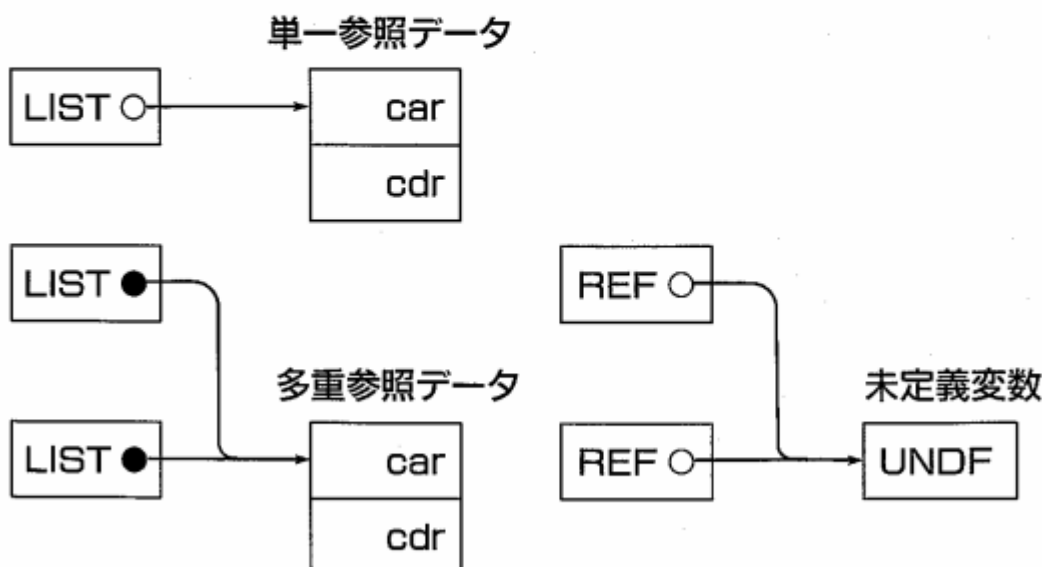
⇒ 多重参照ビット(MRB)方式

⇒ 遅延参照カウント(LRC)方式

多重参照ビット(MRB)による実時間GC方式

● ポインタにおいて参照数情報を管理

● 1ビットの情報(MRB)で単一参照・多重参照を表現



KL1における参照数の更新

- KL1プログラムのクローズ毎にデータへの参照数の増減をコンパイラで検出
p(X) :- true | q(X), r(X). % データのコピー(増)
p(X) :- true | q(X). % データの転送(変化なし)
p(X) :- true | true. % データの消費(減)
- MRBのオン・オフを行なう機能を各ユニケーション命令が持つ
- コンパイラが回収するための命令を陽に生成する

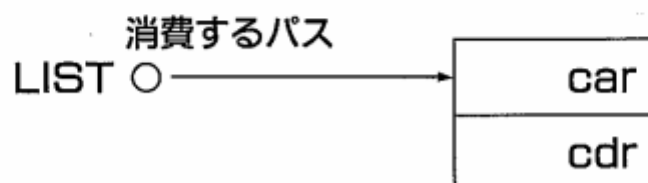
MRBによるメモリ領域の回収

- デレファレンスにおける回収



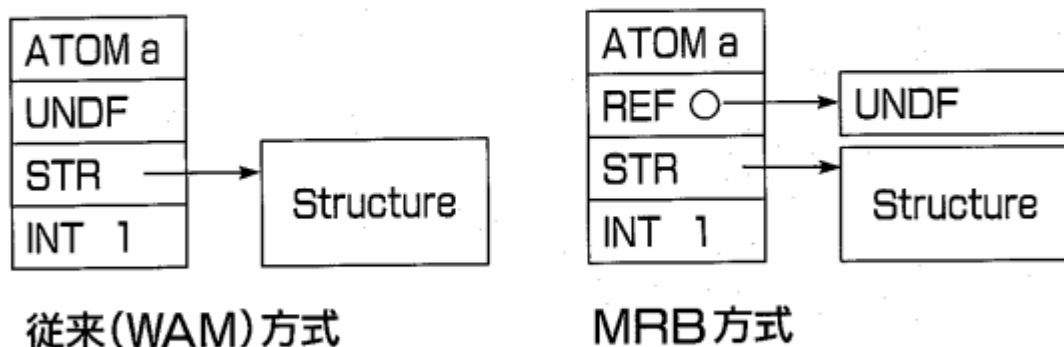
- ユニフィケーションによる参照の消費

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



構造体中の変数とフリーリストによるメモリ管理

- 構造体の外に未定義変数セルを置く
⇒ メモリ使用量が増える

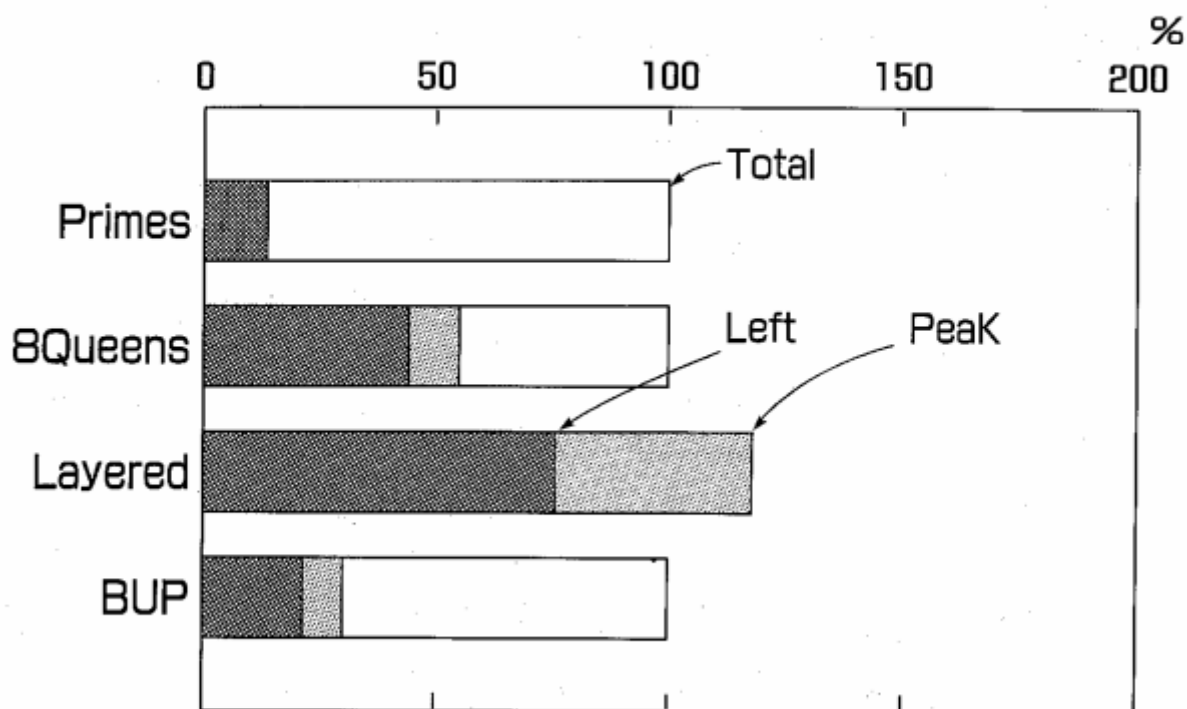


- フリーリストによるメモリ領域の管理
⇒ メモリ割当てのコストが増える

シミュレーション評価

- メモリの使用特性
 - 実行効率は良い(メモリアクセス増は5~20%程度)
 - 節約できるメモリ量はプログラムによって差が大きい(一旦、多重参照になったものは回収できないため)
- キャッシュ特性
 - 単一プロセッサではヒット率向上(メモリ参照の局所性)
 - マルチプロセッサでは並列キャッシュ間のデータの共有率が高まる

メモリの使用特性



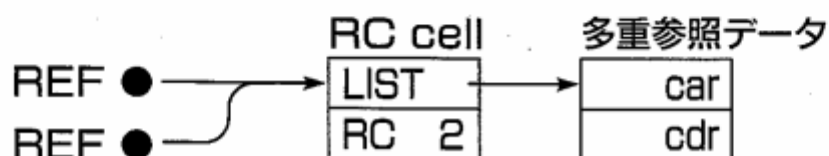
実時間GC方式の応用と改良

§ 参照数管理の副産物

- コピーをしないで単一参照されている構造体の要素を書き替える
- ストリームマージを効率的に実現する

§ 実時間GCの改良・遅延参照カウント(LRC)方式

- ポインタタグによって多重参照と単一参照を識別
- 参照カウント付き間接参照セルを用いて被参照数を表現



ま と め

- KL1の並列処理におけるガーベジコレクション(GC)の重要性
- 多重参照ビット(MRB)を用いた実時間GC方式
- シミュレーション評価
- 実時間GC方式の応用と改良

今後の予定

- 実時間GC方式の改良
- 実時間GCを組み込んだKL1の並列処理系の実装