

PROLOGにおけるヒープ領域の固定長 3M-5 ブロック化によるメモリ管理方式

中村光利¹⁾ 松本茂¹⁾ 竹内洋一²⁾ 迫田行介²⁾

¹⁾ 日立東北ソフトウェア(株)

²⁾ (株)日立製作所 システム開発研究所

1. はじめに

PROLOG言語における節は、実行制御のためのプログラムであるが、一方、知識情報を表すデータでもあり、実行中に、追加/削除などが頻繁に繰り返される。

このため、節を記憶するヒープエリアが節の追加/削除に伴って細分化されると、メモリアクセスに要する時間が増し、処理性能が劣化する。ここでは、この問題を解決するためのヒープエリアのブロック化方式を提案する。

2. PROLOGにおけるメモリ管理方式

PROLOG処理系では、実行時のワーキングメモリを、格納するデータの特性により、スタック領域とヒープ領域に分けて管理している。

以前、木構造の項データを内部コードに変換して登録し、高速処理を行うインタプリタについて報告したが^{1), 2)}、今回、その内部コードの特徴に着目し、ヒープ領域を3種類の独立した領域に分割して管理することを検討した。この中の2つの領域は固定長ブロック方式で管理しており、ヒープエリアが細分化しないよう考慮している。

3. 具体的な処理方式

本方式ではヒープ領域に格納するデータは、固定長データと可変長データに分けられる。

固定長データは、主に内部コード化されたPROLOGプログラムを格納している。一方、可変長データは、それ以外のテーブル類などである。

固定長形式で扱われるデータは、PROLOGの節の実行順序や生成、消滅を管理する「管理ノード部」と、節の内容を示す「データ部」から構成される。今回、これらを分離し別の独立した領域に格納する。「管理ノード部」を格納する領域をD-HEAPと呼び、「データ部」を格納する領域をP-HEAPと呼ぶ。

D-HEAPとP-HEAPは共に固定長ブロックに分割し利用するが、ブロック長が異なり、D-HEAPは管理ノードが収まる単位にし、P-HEAPは分割管理によるメモリおよび処理スピードのオーバーヘッドが少なくなる単位に設定している。

一方、可変長データを格納する領域をT-HEAPと呼ぶ。

この3種類の領域は、各々独立したフリーチェーンで未使用ブロックおよび未使用エリアを管

表1 各ヒープ領域の特性

領域名称	対応するデータ	特性		
		圧縮の要否 (細分化の有無)	即時回収 可能性	一括回収 の要否
D-HEAP	管理ノード部	不要(無)	回収不能	必要
P-HEAP	データ部 (PROLOGの節の内容)	不要(無)	多くの場合 回収可能	若干必要
T-HEAP	各種テーブル	必要(有)	全て 回収可能	即時回収の ため不要

Fixed-blocking heap management in PROLOG implementation

Mitsutoshi NAKAMURA¹⁾, Shigeru MATSUMOTO¹⁾, Yooichi TAKEUCHI²⁾, Kousuke SAKODA²⁾

理している。ヒープ領域の管理形態を図示すると図1のようになる。

可変長のT-HEAP領域内にある名前表からD-HEAP領域にある管理ノードをポイントし、管理ノードからP-HEAP領域のデータを管理する。データの生成、消滅が多発するD-HEAP領域とP-HEAP領域に関しては固定長ブロック管理する。

以上のように分離した場合の各領域の特性は表1のようになる。

一般の実用的プログラムでは、PROLOGの節を削除した場合、管理ノード部は即時回収が一般に不可能であるが、データ部は即時回収が可能である場合が多い。したがって、管理ノード部とデータ部とをD-HEAPとP-HEAPとに分けて格納することにより、即時回収の回収量を増すことができる。また、D-HEAP, P-HEAPを固定長のブロック単位で管理することによりメモリの細分化が発生しなくなり、オーバーヘッドの大きい圧縮処理をT-HEAPのみに限定できる。

4. 効果

- (1) T-HEAPとD-HEAP/P-HEAPとを分離することにより、D-HEAP/P-HEAPの細分化を抑止でき、データの圧縮が必要なくなる。
- (2) さらにD-HEAPとP-HEAPとを分離することにより、P-HEAPの即時回収の可能性が高まり、一般の実用的プログラムでは、不要になったデータの多くが即時回収でき、一括回収がほとんど必要なくなる。
- (3) P-HEAPを固定長にすることにより、プログラム格納領域を捜す処理が不要になり、assert処理が高速化される。
- (4) GCのためのマーキング処理がセル単位からブロック単位へと単純化されるため、ヒープに対するGCが高速化される。表2に、事実文p(a, b, c)を10000回assertした後のヒープエリアに対するGCの所要時間を示した。
- (5) T-HEAPに格納する名前表などは、大部分一方的に追加するだけで削除しないので、データ圧縮によるオーバーヘッドは実質的に少ない。

5. おわりに

今回、ヒープ領域を3分割の固定長ブロックによる管理に変更することにより、メモリ管理

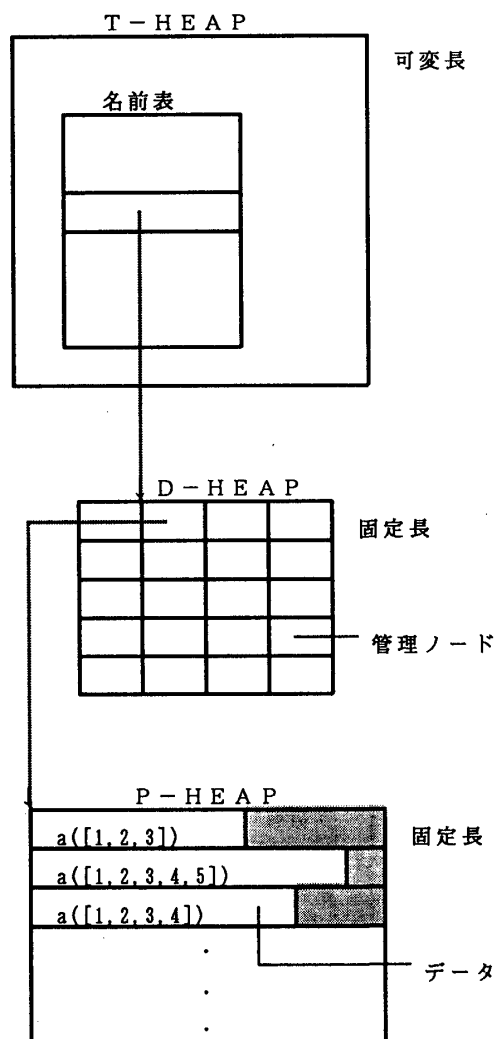


図1 ヒープ領域管理形態

表2 ヒープエリアに対するGC時間

セル単位	ブロック単位
0.63sec	0.18sec

処理の単純化及び高速化を図る手法の検討を行った。

参考文献

- [1] 広瀬他：論理型言語処理系「LONLI」における物理的データ表現方式の提案と評価、情報処理学会第32回全国大会(1986)
- [2] 竹内他：Prolog高速インタプリタの処理特性、情報処理学会第40回全国大会(1990)