KL1 抽象命令セットの改良について

6Q-5

木村康則 後藤厚宏 中島克人 近山隆
(財) 新世代コンピュータ技術開発機構 (ICOT)

1 はじめに

KL1 は、ICOT で設計された並列論理型言語である。本稿では、既に提案した KL1 の抽象命令セット [1] の改良案を提案する。主な改良点としては、クローズインデキシングの採用に伴って新たに命令を追加したこと、構造体の要素を参照するために使っていた構造体レジスタを廃止したこと、MRB 管理のための命令およびガーベジコレクション (MRB-GC) 用の命令を新たに追加したことである。

2 クローズインデキシング

従来 KL1 コンパイラでは、候補クローズ群はソースプログラム上現れた順に、クローズ間の情報は考慮にいれずにクローズ単位にコンパイルしてコードを生成していた。そして実行時には、各候補クローズの選択の試みはコンパイルされた順に逐次に行われていた。従って、実行中に同じ変数を何度もデリファレンスしたり、リストの Car、Cdr に分解するといったことが起こっていた。

そこで、実行の高速化のために、同じ述語名のクローズを一纏めにして扱い、各クローズのヘッド引数から各クローズが選択される条件を求め、これに基づいてクローズを分類し、重複した余分な処理を行わないような方式を提案した [3]。本稿では、クローズが分類された後の命令の生成方針および命令セットについて説明する。

2.1 クローズインデキシング用の命令

KL1 コンパイラでは、インデキシングの対象となった引数に対し、以下の様に処理を分類して各々の処理に対応する命令を生成し、その後、連続した命令の統合を行い、命令数の増大を防いでいる。

- 引数のデリファレンスと具体化の確認
- 引数のタイプチェック
- 値一致のチェック

これらの処理のために、インデキシング用の命令セットとして、表 1に示すような種類を用意した。

ここで、"XXX" は、"list"、"vector" などの Ai の取り得るデータタイプを示す。また、引数が未定義のために実行が中断し、次候補クローズのコードへ分岐する場合と、値の不一致 (以下 "失敗" と言う) などにより、次候補クローズのコードへ分岐する場合を分け、前者の分岐アドレスは命令中の "Lab" によって指定し、後者は "try_me_else" 命令で指定するようにした。(従来は、実行の中断および失敗時の分岐アドレスは全て "try_me_else" 命令によって設定されていた。) こうした理由は、インデキシングでは分岐命令が多く使われると予想され、またこの分岐先を全て "try_me_else" 命令で設定すると、"try_me_else" 命令の数が増え、かえって実行時間や静的コードサイズの増大を招くと考えられたからである。

種類	命令	機能
(1)	wait Ai	デリファレンス
(2)	is_XXX Ai, Lab	タイプチェック
(3)	test_XXX C, Ai, Lab	値一致チェック
(4)	wait_XXX C, Ai	(1) + (2) + (3)
(5)	jump_on_non_XXX Ai, Lab	(1) + (2)
(6)	check_XXX C, Ai, Lab	(2) + (3)

表 1: インデキシング用の命令の種類

2.2 コンパイル例

"Append" プログラムをインデキシングをかけてコンパイルした時のコードを示す。

```
a([A|X],Y,Z) :- true | Z = [A|ZZ], a(X,Y,ZZ).
a([],    Y,Z) :- true | Y = Z.
```

```
a/3:    try_me_else a/3/1
        jump_on_non_list A1, a/3/2
        read_element A1, 0, A4
        read_element A1, 1, A5
        put_list A6
        write_val_element A6, 0, A4
        write_var_element A6, 1, A7
        get_list_value A3, A6
        put_value A5, A1
        put_value A7, A3
        execute a/3
a/3/2:  check_atom [], A1, a/3/1
        get_value A2, A3
        proceed
a/3/1:  suspend a/3
```

3 構造体レジスタの廃止

構造体レジスタ (以下 S レジスタ) は、ガード部で構造体の要素を読み出したり、ボディ部で構造体を作る時にその要素を本体に書き込むための構造体の要素位置を指定するために用いられていた。しかし、以下の述べる理由から、S レジスタを廃止することにした。

Improvement of KL1 Abstract Instruction Set
Y.Kimura, A.Goto, K.Nakajima, T.Chikayama(ICOT)

前章で述べたクローズインデキシングを行う際、KL1コンパイラはヘッド引数に現れた構造体の要素もインデキシングの対象としている。この時のコンパイラの操作としては、構造体を一時に分解して、要素をレジスタに展開する方式と、必要になった時点でインデキシングの対象となった要素をレジスタに読み出してくる方式が考えられる。前者は、コンパイラの処理は簡単であるが、レジスタを多く使う、以後の処理で使われない要素まで読み出してしまうことがある、と言う欠点がある。そこで、我々は、後者の方式を採用し、引数位置をオペランドとして指定するような命令を新たに用意した。表2に代表的な構造体操作命令の新旧対比を示す。

旧命令	新命令
read_variable Ai	read_element V, I, Ai
write_variable Ai	write_var_element V, I, Ai
write_value Ai	write_val_element V, I, Ai

表 2: 新旧命令の仕様

ここで、"V" と "Ai" は各々構造体へのポインタ、読み出す / あるいは書き込む構造体要素をもつレジスタである。"I" は構造体要素位置を示す即値である。

4 MRB 管理用の命令

KL1 コンパイラのもう一つの特徴は、コンパイル時にクローズを解析し、MRB-GC と呼ばれるインクリメンタルなガーベジコレクションを可能にする命令列を生成することである [2]。

MRB-GC 用命令には、(1) クローズが選択された時点で、回収出来るかも知れない KL1 データ (変数、構造体) に対して生成される "回収用の命令"、と (2) 能動部の実行に際して、受動部から受け取ったデータの参照数が増えたり、新たに割り付けたデータに対して正しく MRB を付ける "マーク用の命令" の二種類がある。ここでは、後者の命令に関するもののうち、構造体要素マーク用の命令について説明する。

"マーク用の命令" は、一旦レジスタに読み出したデータの MRB をオンにするための命令である。しかし、図 1に示すようにベクタの要素をガード部で取り出し、ベクタ本体と取り出した要素の両方をボディ部で使う場合には、ベクタ要素の MRB もオンにする必要が生じる。

命令	操作
mark_element V,I	ベクタ V の I 番目の要素の MRB をオンにする。

表 3: 構造体要素マーク用の命令

図 1では、このクローズが選択された時には、変数 "E" の MRB とベクタ "V" のレジスタ "I" で示される要素 MRB の両方をオンにしなければならない。なぜならば、ここで取り出された要素の変数は、"E" とベクタ本体の両方から指されているからである。この時、変数 "V" の MRB をオンにする方法もあるが、こうするとベクタ本体の MRB がオンになってしまい、ベクタの本体が回収できなくなってしまう。これは、MRB-GC の効果を低下させてしまうので避けるべきであろう。図 2に、図 1のプログラムのコンパイル例を示す。

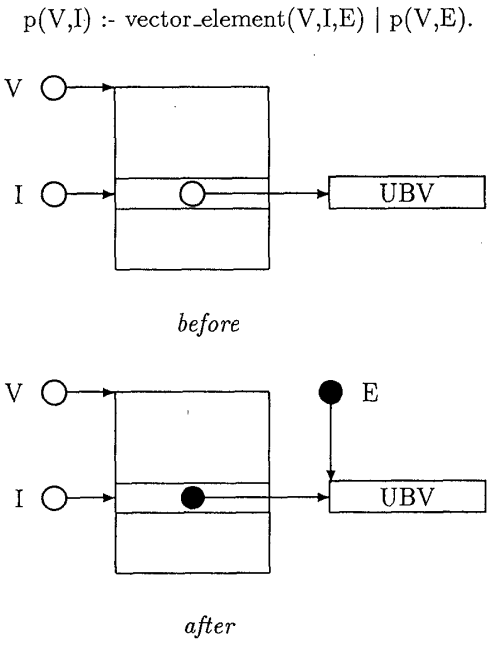

図 1: 構造体要素マークの例

```
p/2:   try_me_else p/2/1
       vector  A1
       integer A2
       vector_element A1, A2, A3
       mark_element A1, A2
       put_value A3, A2
       execute p/2
p/2/1: suspend p/2
```

図 2: vector_element コンパイル例

5 おわりに

KL1 の抽象命令セットの改良案について説明した。今後評価を進め、さらに改良を進めていく予定である。

日頃御指導いただく ICOT 第4研究室内田俊一室長に感謝します。また、関田大吾、平野喜芳、中越靖行の各氏を始めとする ICOT 並列処理検討会メンバ諸氏に感謝します。

参考文献

[1] Y.Kimura and T.Chikayama : An Abstract KL1 Machine and Its Instruction Set, Proc. of SLP'87

[2] T.Chikayama and Y.Kimura : Multiple Reference Management in Flat GHC, Proc. of ICLP'87

[3] 西崎 他: KL1 クローズインデキシング方式の評価, 本大会予稿集