

Prologコンパイラの翻訳性能に関する一考察

3Y-6

杉崎 由典¹, 中村 三夫²

¹⁾ (株)富士通静岡エンジニアリング, ²⁾ 富士通株式会社

1. はじめに

我々が作成したPrologコンパイラは、高速オブジェクトを生成するためにWarren_baseの中間コードを用いている。これにより実行性能2Mlips(FACOM M780)を実現した。実行性能を向上させるに伴い、コンパイラの実行処理が多く、翻訳時間が増大する結果となった。

そこで、翻訳時間の縮小を目的とした性能改善を行うこととした。ここでは、その際に行った翻訳性能に対する考察を報告する。

2. 翻訳時間を左右する要因

コンパイラの翻訳時間は、入力ソースの形状(性質)によって左右される。コンパイラの翻訳時間の特性を考えるには、この性質的な要素と翻訳時間との関係を明確にすべきである。そのため翻訳時間は中間コード出力までの時間とし、これとソース形状との関係を比較する。

Prologソースの形状を決める要因には以下に挙げるものがある。

(1)ゴールの数

t1:-a1. t2:-a1, a2. tn:-a1, ..., an.

(2)選択枝の数

t1. t2. tn.
t2. .
.tn.

(3)引数の数

t1(1). t2(1,2). tn(1,2, ..., n).

(4)変数の数

t1:-a1(X). tn:-a1(X), ..., an(X).

(5)リストの要素の数

t1([1]). tn([1, ..., n]).

(6)構造体の要素の数

t1(s(1)). tn(s(1, ..., n)).

(7)データタイプの違い

t1(1). t2(X). t3([1]). t4(s(1)).

実際のソースではこの要因が複雑に絡み合ったものとなるが、これらの一要因がどの程度の影響を与えるかを純粋に測る。これにより各要因の翻訳時間への影響量を見ることができる。

3. 翻訳時間と要因との関係

(1)から(6)までの各要因の数を変化させ、これと翻訳時間との関係を表したものが図1、図2である。また(7)についてはそれぞれのデータタイプと翻訳時間との関係を表1として表わした。

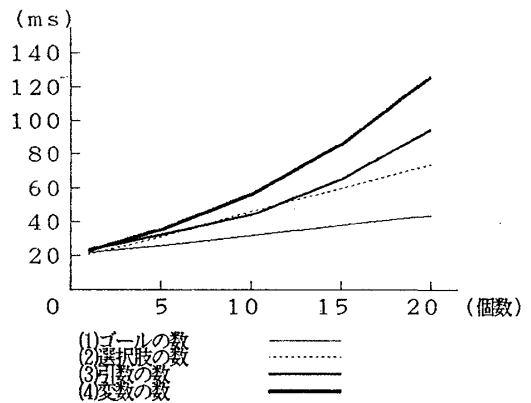


図1 各要因と翻訳時間との比較(1)

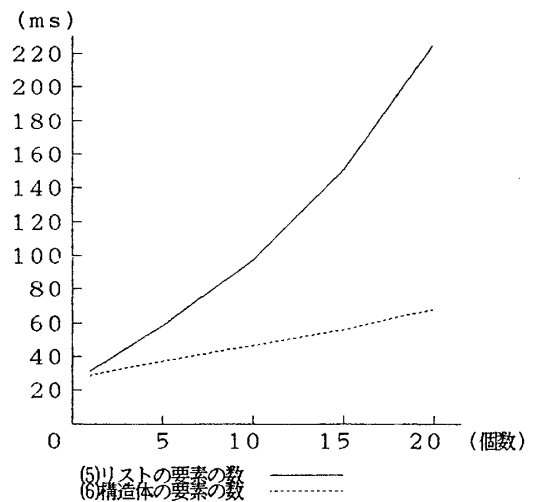


図2 各要因と翻訳時間との比較(2)

A study on performance of compilation in Prolog compiler

Yoshinori SUGISAKI¹, Mitsuo NAKAMURA²

¹⁾FUJITSU SHIZUOKA ENGINEERING Ltd., ²⁾FUJITSU Ltd.

表1 データタイプの違いと翻訳時間との比較

タイプ	定数	変数	リスト	構造体
時間 (ms)	28	26	35	32

4. 考察

翻訳時間と要因との関係を示す図表から次のことが分かる。

(1)図1, 図2より, 翻訳時間と個数との関係が線型であるもの(1)(2)(6)と, 線型とならないもの(3)(4)(5)に分けられる。

それぞれについて分析する。

①翻訳時間と個数との関係が線型のもの(1)(2)(6)

線型を示すものは, 引数の数が変化しないものである。

図1, 図2の線の傾きによって, 各要因が翻訳時間に与える影響量が分かる。処理時間の大小関係は以下のとおりとなった。各要因ごとの処理量から考えて, この大小関係は妥当であるといえる。

ゴールの数 < 構造体の要素の数 < 選択肢の数

②翻訳時間と個数との関係が線型でないもの(3)(4)(5)

非線型となるものは, すべて引数を含んだものである。引数が及ぼす影響がこの特性となって現れるといえる。

引数のあるプログラムでは, すべての引数がレジスタ割り付けの対象となる。レジスタ割り付けは, 最適化によりコード全体のレジスタ状態を考慮するため, 引数が増えれば増えるほど処理が重くなる。そのため, このレジスタ割り付けの影響により線型でなくなる(二乗に比例する)

また変数を含むプログラムでは, グローバル・レジスタ割り付けも行っているため, この影響も受ける。

一般にPrologのプログラムでは, 引数・変数が多く使用されるため, これらの個数による影響は非常に大きい。

(2)図2より, リストと構造体の翻訳時間に違いがあることが分かる。

構造体は要素数分だけ中間コードを生成すれば良いこれに対して, リストは要素数分のリスト・セル生成の中間コードと, 要素数分の中間コードとを生成しなくてはならない。リストと構造体の翻訳時間に差が生じるのは, この翻訳量の違いのためである。

(3)表1より, データタイプによる翻訳時間の大小は,

次のとおりである。これはデータの構造から考えて納得できる。

変数 < 定数 < 構造体 < リスト

5. 今後の課題

以上により, レジスタ割り付けの最適化処理が翻訳時間に与える影響が大きいことがわかった。レジスタ割り付けの最適化は, 出力オブジェクトの実行性能に大きく関与している。また, リストのコード生成処理もリスト処理を高速に実行するために行っているものである。今後は, 実行性能を落とさずにこれらの問題点を改善していくことが課題である。

6. まとめ

本報告では, Prologのソースコードの性質がコンパイラに対して, どのように, どの程度影響があるかを翻訳時間を尺度として考察した。これによりコンパイラの問題となる箇所がはっきりした。これからは, この結果を基に問題となる箇所の改善を進めていく予定である。

末筆ながら, 日ごろ御指導御援助くださる富士通研究所A I研の皆様, 弊社津田課長ならびにProlog関係者諸兄に感謝します。

参考文献

- [1] Warren, D.H.D.: An Abstract Prolog Instruction Set, SRI International, Technical Note 309, (1983)