

C言語コンパイラ構成方式\*

2G-1

漆原誠一 湯川博司 足立佳啓 宇野結 春名修介†  
 松下電器産業株式会社‡ 情報通信関西研究所

1 はじめに

近年、マイクロコンピュータのソフトウェア開発が大規模化し、開発ツールとして高級言語が必要不可欠となっている。同時にマイクロプロセッサの品種が増加しているため、各プロセッサ用高級言語コンパイラの迅速な開発が急務となっている。

本発表は、このような状況を考慮したターゲット毎に再構築の容易なCコンパイラの一構成方式について報告する。

2 構成

本コンパイラ構成方式では、CPU独立部とCPU依存部をモジュール単位で明確に分離し、新規にコンパイラを開発する場合は、CPU依存部のみを作成する。以下に本構成方式の特徴をまとめ、内部構成を図1に示す。

- (1) テーブル駆動方式のコード生成手法
- (2) 汎用化手法の体系的組み込み
  - マクロ定義 … 変数のスタック内オフセットなどの設定値の定義
  - テーブル選択 … アドレッシングモードの選択など
  - 関数の修正 … CPU依存の最適化処理など  
(種類限定、入出力仕様規定)
- (3) 構文解析部の部品化

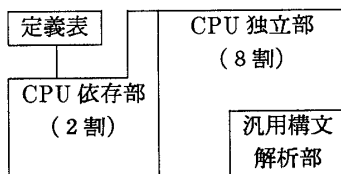


図1: 内部構成

3 構文解析部

コンパイラの構文解析部は対象CPU向けの中間コードを生成するために、その中に対象CPU依存部を含む構成が一般的であった。そのため、各種のCPUへの対応は、構文解析部の中に点在するCPU依存部を直接書き換えることで実現されていた。また、構文解析部は、C言語ソースを解釈して種々のデータを生成する静的解析ツールにも応用できるが、コンパイラの構文解析部をそのまま利用したのでは無駄な処理も多く、構文解析部とツールの接続が容易ではなかった。本システムの構文解析部は、上記の問題を解決するために

- ANSIのC言語規格案に準拠した構文解析処理を行ない、C言語ソースを忠実に反映した中間データを生成する汎用構文解析部(言語仕様依存部)
- 対象CPUの特性をデータとして持ち、汎用構文解析部の生成する中間データを対象CPU向けの中間コードに変換するポストプロセッサ(対象CPU依存部)

を完全に分離する構成とした。これにより、第1に、各種CPUへの対応は、ポストプロセッサの変更だけでよく、コンパイラのインプリメントが従来よりも速やかに行なえるようになった。第2に、汎用構文解析部の静的解析ツールへの応用が、無駄なく容易に行なえるようになった。

4 コード生成部

本システムは、式木を単純なボタンに分解し、それぞれのボタンに対して定義された命令列を出力することで、式木全体の演算を実現するための命令列を構成・出力する、いわゆるテンプレートマッチング方式によってコード生成をおこなう。本システムでは、インプリメンタの負担を軽減するために、次のようなアプローチをとっている。

- 構造化テンプレート
  - 段階的に式木の分析を詳細化して、よりよいコードを出力するように改良を加えられる、構造化テンプレートを導入した。さらに、汎用的なテンプレートをあらかじめ用意し、一通りのコード生成を行なうコンパイラを簡易に作成できるようにした。
- アドレッシングモード制限テーブル
  - 柔軟なアドレッシングモードを許すCPUを想定したアドレッシングモード選択ルーティンを用意し、対象CPUによる制限を、テーブルの形で簡易にインプリメントするためのアドレッシングモード制限テーブルを設けることにより、対象CPU特有のアドレッシングモードを選択するための煩雑な処理ルーティンをつくり直す必要をなくした。

5 まとめ

現在、本構成方式を実際のコンパイラの開発に適用中である。今後は、本構成方式をもとにして、コンパイラ開発における自動化率の向上をはかる必要がある。

参考文献

Aho, A. V., Sethi, R., and Ullman, J. D., *Compilers Principles, Techniques, and Tools*, Addison-Wesley, Reading, Mass., 1988.

\*An Architecture of C Language Compiler  
 †Seiichi URUSHIBARA, Hiroshi YUKAWA, Yoshihiro ADACHI, Munshi UNO, Shusuke HARUNA  
 ‡Matsushita Electric Industrial Co., Ltd.