C-12

C コンパイラ用テストスイートおよびその生成ツール
Testsuite for C Compilers and Its Generating Tool

内山 裕貴† 引地 信之†† 永松 祐二† 石浦 菜岐佐†
Yuki Ucihyama Nobuyuki Hikichi Yuji Nagamatsu Nagisa Ishiura

1 はじめに

ソフトウェア開発において, テストは非常に重要な工程である. 特にコンパイラは基本ソフトウェアであり, そのテストは極めて重要である. 一般に C コンパイラのテストは C プログラムをコンパイルし実行した結果が期待値と一致するかを確認することにより行うが, このプログラム集合をテストスイートと呼ぶ. コンパイラのテストスイートには類似したプログラムが多く含まれる傾向があるが, これはコードの保守性や可読性がという観点からは望ましくない. また, 実行環境, コンパイルオプション, テストしたい項目をカスタマイズするためには多くのファイルを変更しなければならない場合が多い.

本稿ではこのような問題を解決するための一手法として, テストスイートのテンプレート記述からテストスイートを自動生成する方法を提案する. また, 本手法を SRA で開発された C コンパイラ用テストスイートに適用する.

2 テストスイート

コンパイラのテストスイートとしては GCC に付属の testsuite [1] があるが, これは GCC の C 言語に対する独自の拡張のテストを含んでおり, また入出力ライブラリの存在を仮定しているため, これを他のコンパイラのテストにそのまま用いることは必ずしも容易ではない. これに対し, SRA で開発された C コンパイラ用テストスイート (以下 SRA testsuite) は, 多くの C コンパイラに対して適用することを目的とした汎用的なテストスイートである. 入出力ライブラリがなくてもユーザ定義関数によるテストが可能であり, シンボリックシミュレータを用いればアセンブラがなくてもテストが可能である等, 幅広い実行環境に対応できるため, コンパイラ開発の初期段階, クロスコンパイラや組み込みプロセッサ用コンパイラへの適用が可能である.

SRA testsuite では約一万個のテストプログラムが, テスト項目の種類ごとに約 100 個のディレクトリに配置されている. テストスイートはこの他, テストの設定ファイル, ライブラリ, 実行スクリプト等から構成される. テストは GNU の dejagnu を用いて行う. dejagnu は Expect という言語で記述されており, 設定等も Expect を用いて記述する.

SRA testsuite には, 変数の型や条件分岐の内容のみが異なるテストプログラムが非常に多く含まれており, 保守性は必ずしもよくない. また, コンパイラの指定, シミュレータによる実行等のテスト環境のカスタマイズを行うには多くのファイル (各テストディレクトリに配置されているものもある) を修正する必要が生じる. 更に, テストプログラムが選択できないため, 一部のテスト項目のみを実行することができないという問題もある. 本研究ではこのような問題を, テンプレートファイルと設定ファイルを用いたテストスイート生成手法により改善することを試みる.

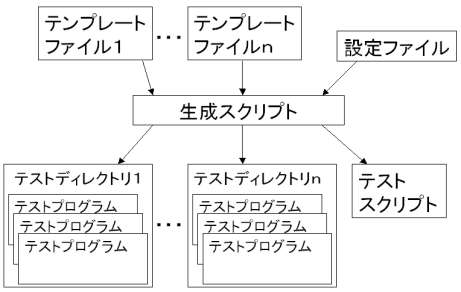

図 1: テストスイート生成の流れ

3 テストスイートの生成

本稿で提案するテストスイート生成の流れを図 1 に示す. テスト生成ツールはテンプレートファイルと設定ファイルを入力として, テストプログラム, およびテストスクリプトを生成する.

テンプレートファイルは意味的にまとまりのある複数のテストプログラムを一つにまとめて記述したものであり, ここから生成されるテストプログラムが一つのテストディレクトリに対応する. テンプレートファイルでは, 変数による共通部分の置換や繰り返し構文等により, 記述量の削減と保守性の向上を図る.

図 2 にテンプレートファイルの記述を示す. @ではじまる行はテンプレートで使用する特殊な命令であり, それぞれが命令の終わりを示す @end で始まる命令に対

†関西学院大学 理工学部
††SRA 先端技術研究所

```
 1 @comment
 2 変数の代入のテスト
 3 @end_comment
 4
 5 @multiline HEAD
 6 #ifdef UNIX
 7 #include <stnd.h>
 8 #endif
 9 @end_multiline
10
11 @for MODIFIER '' 'static' 'register'
12 @for VARIABLE 'int' 'short' 'unsigned' 'char'
13 @file t???_$MODIFIER_$VARIABLE.c
14 $HEAD
15 main(){
16   $MODIFIER $VARIABLE Variable;
17   Variable = 1;
19   if (Variable == 1) printok();
20   else               printno();
21 }
22 @end_file
23 @end_for
24 @end_for
```

図 2: テンプレートファイルの例

応している. 1 から 3 行目はコメントであり, 生成されるテストプログラムには影響しない. 5 行目から 9 行はグローバルな変数の定義である. ここでは HEAD という変数を定義しており, 14 行目の $HEAD は 6 から 8 行目の内容で置換される.

11, 12 から 23, 24 行目の@for から@end_for はその間に記述されたコードを, 変数の置換を行いながら繰り返し生成することを提案する. この@for はネストが可能であり, この例では 12 通りの異なったプログラムが生成される.

11 から 19 行目の @file から @end_file で囲まれた記述は一つのテストプログラムに対応する. @file の後にはファイル名を記述し, "???" はその桁数の番号が自動的に生成される. ファイル名は, 例えば "t001_static_int.c" となる.

```
 1 CC:m32r-unknown-elf-gcc
 2 TARGET:m32r-gcc
 3 SIMULATOR_NAME:m32r-unknown-elf-run
 4 TOOL:gcc
 5 LIBS:
 6 LDFLAGS:
 7 FILSET:FILSET
 8 OBJ:.
 9 LOG:log
10 TESTDIR:m32r-testsuite
```

図 3: 設定ファイルの記述例

図 3 は設定ファイルの記述例である. 1 行目から 3 行目で, 使用するコンパイラやシミュレータの名前等を記述する. 4 行目はテストを行うディレクトリの名前の指定である. 5 行目から 7 行目ではテスト実行時のオプションやライブラリ等の指定を行う. 8 行目から 10 行目ではテストスイート, ログ等を出力するディレクトリを指定する.

4 ツールの実装

以上に述べたテストスイート生成システムを Perl 5 で実装した. 表 1 は元の SRA testsuite のテストプログラムの行数と, 同じテストスイートを生成するテンプレート記述の行数を比較したものである. 全体としては行数が約 60%に減少した.

テスト項目のカスタマイズは, テストスイート生成時に必要なテンプレートファイルだけを選択すること, あるいはテスト実行時に特定のテストディレクトリを指定することにより行える. また, 実行時にテストスクリプトを指定することにより, コンパイルのみのテストやコンパイルオプションの変更等も容易に行える.

表 1: テンプレートファイルの適用前と適用後

ディレクトリ名	元のテストスイート (行)	テンプレート記述 (行)
gcc.1-1	1,822	965
gcc.2-1-01	3,351	999
gcc.3-1	6,080	5,623
gcc.4-1	2,364	2,149
全体	485,232	284,619

5 むすび

本研究ではテンプレート記述からのテストスイート生成と, その SRA testsuite への適用について述べた. 今後の課題として C 言語の種々のバージョン (K&R C, ANSI C, C99) への対応, テスト項目追加による SRA testsuite 自身の増強等が挙げられる.

なお本研究は, IPA 2004 年度オープンソースソフトウェア活用基盤整備事業のプロジェクト「C コンパイラ向けテストスイート生成ツールの開発」(契約番号 2004 情財第 0825 号) に採択されたもので, テストスイートおよび生成ツールは [2] で公開している.

参考文献

[1] http://gcc.gnu.org/

[2] http://ist.ksc.kwansei.ac.jp/~ishiura/gcc/ipa/