A-06

C コンパイラ用テストスイート生成システム testgen2

testgen2: Test Suite Generation System for C Compilers

森本 和志 † 内山 裕貴 ‡ 石浦 菜岐佐 † 引地 信之 §
Kazushi Morimoto Yuki Uchiyama Nagisa Ishiura Nobuyuki Hikichi

1 はじめに

ソフトウェアを開発するための基礎ツールとして, コンパイラには高い信頼性が要求される. このためコンパイラの開発に際しては, テストスイートやランダムに生成したプログラムを用いた徹底的なテストが行われる.

コンパイラのテストスイートにはその性格上類似したプログラムが多数含まれるが, これが保守性の点でコードクローンと同様の問題を引き起こす. testgen [1] は, 繰り返し構文を持つテンプレートからテストスイートを生成する方式によってこの問題の解決を試みている. しかし, テストスイート中には testgen のテンプレートでは重複の削減ができないパターンが数多く存在する. また, testgen の生成するテストスクリプトには, テストプログラムが予期せぬ出力をした場合の合否判定や, テストプログラムが無限ループに陥った場合の処理に問題があった.

そこで本稿では, 新たなテンプレート構文を持つテストスイート生成システム testgen2 を提案する [2]. testgen2 のテンプレート構文は, 引数付きマクロの導入により, testgen で困難であったテストプログラム中の重複コードを削減する. testgen2 を実装した結果, テンプレートの記述量は testgen に比べて 53.8% に削減できた. また, テストスクリプトの改良により, PASS/FAIL 判定の正確化と処理時間の監視による無限ループの打ち切り処理を実現した.

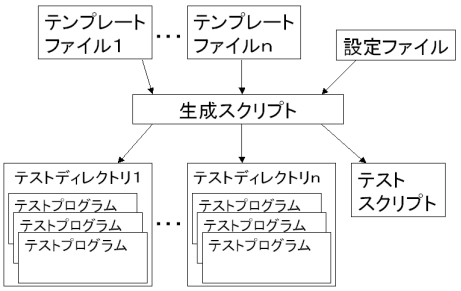

図 1 testgen のテストスイート生成

```
 1: @multiline HEADER
 2: #ifdef UNIX
 3: #include <stnd.h>
 4: #endif
 5: @end_multiline
 6:
 7: @for MODIFIER '' 'static' 'register'
 8: @for VARIABLE 'int' 'short' 'unsigned' 'char'
 9: @file t???_$MODIFIER_$VARIABLE.c
10: $HEADER
11: main(){
12:  $MODIFIER $VARIABLE Variable;
13:  Variable = 1;
14:  if (Variable == 1)  printok();
15:  else printno();
16: }
17: @end_file
18: @end_for
19: @end_for
```

図 2 testgen のテンプレート記述の例

2 testgen テストスイート

テストスイート生成システム testgen は, C コンパイラ用テストスイートとともに GPL で公開されている[*1].

testgen のテストスイート (以下「testgen テストスイート」と呼ぶ) は, 主として開発段階の C コンパイラへの適用を想定したものであり, 約 9,000 本 (総行数約 70 万行) の C プログラム (以下「テストプログラム」と呼ぶ) が 85 のディレクトリに分けて格納されている. 各テストプログラムは, 計算結果と期待値との照合を内部で (1 回以上) 行い, その判定毎に結果が正しい場合には「@OK@」を, そうでない場合には「@NG@」を出力する.

コンパイラのテストスイートには, ある処理が異なる型や記憶クラスの変数に対して行えることをテストするために, 類似するコードが繰り返し出現する傾向があるが, これはテストスイートの保守を考えた場合には, コードクローンと同様の問題を引き起こす. そこで, testgen では, マクロや繰り返し構文等によってテストスイート中のコードの重複をできるだけ排除した「テンプレート」によりテストプログラムの集合を記述し, これを展開することによってテストスイートを生成する (図 1 参照; 1 ディレクトリ分のテストプログラムを 1 つのテンプレートで記述する).

testgen のテンプレート記述の例を図 2 に示す. 1 行目の `@multiline HEADER` は 5 行目までを HEADER と

† 関西学院大学
‡ 株式会社ケイ・オプティコム
§ 株式会社 SRA

[*1] http://ist.ksc.kwansei.ac.jp/~ishiura/testgen/index.html

いう名のマクロとして定義している (この例では 10 行目の $HEADER の部分に展開される). 7 〜 8 行目の @for は繰り返し構文である. この例では二重の @for により $3 \times 4 = 12$ 通りのプログラムが生成される.

しかし, このテンプレート構文では, 2 つの文字列が連動して変化するパターンが出現すると, 重複記述の削減が困難になる. 例えば, 13 〜 14 行目に入力される値が変数の型によって変わると, それを 1 つの記述で表現することはできない. また, 類似した記述が別のディレクトリに存在する場合, その重複の削減は行えない.

testgen のテストスクリプトでは, 出力に @NG@ が出力されない限り PASS と判定しているため, コンパイラに「@OK@」も「@NG@」も出力されない不具合があっても FAIL と判定されない. また, コンパイラの不具合により実行が無限ループに陥った際にそれを止める機能が存在しないことも問題となっている.

3 testgen2 のテンプレート

前節でも述べた通り, testgen のテンプレートで重複記述の削減ができない最大の要因は, 2 つ以上の文字列が連動して変化するパターンが多い点にあった. そこで本稿では, 引数付きマクロを導入したテンプレート構文を持つ testgen2 を提案する.

図 3 は testgen2 のテンプレート記述の例である. 2 行目の @def から 20 行目の @def_ がマクロの定義である. このマクロは引数を持っており, 22 〜 23 行目のように呼び出される (@file は, テストプログラムの生成を意味する). マクロの引数に配列 ([ ] で囲まれたコンマ区切りのリスト) を指定すると, その各要素に対してマクロの呼び出し (と @file によるテストプログラムの生成) が行われる. 22 〜 23 行目のようなマクロ呼び出しにより, 2 つ以上の文字列が連動して変化するパターンをコンパクトに記述することができる.

別テンプレートのマクロ参照には @refer 文を使う. 図 3 の 1 行目は define.tt というテンプレートを読み込み, その中の @def 文の解析を行うことを指示している. これにより, testgen では行えなかった複数ディレクトリにまたがる重複記述の削減が行える.

4 testgen2 の実装

testgen2 は Perl (5.8.0) で実装しており, Cygwin (1.7.5), Ubuntu Linux (10.04), Mac OSX (10.5.8) で動作を確認している.

testgen2 のテンプレートは testgen のテンプレートから変換し, テンプレートから生成される全てのテストプログラムが testgen テストスイートのものと一致することを確認した. testgen2 のテンプレートの特長を生かした重複記述の削減は現在手動により進めている. 表 1 に書き換えを行った 53 ディレクトリに対して, テンプレート記述の行数を比較した結果を示す. 元のテストスイートの記述量を 100% とした場合の行数が testgen では 32.7% であったが, testgen2 ではこれを 17.6% に削減できた.

```
 1: @refer define.tt @refer_
 2: @def $macro($TYPE, $VARIABLE)
 3: $HEAD()
 4: main()
 5: {
 6:  $TYPE             Variable;
 7:
 8:  itest = NO;
 9:  Variable = $VARIABLE;
10:  itest = Variable;
11:
12:  if (itest == $VARIABLE)
13:  printok();
14:  else
15:  printno();
16:
17:
18:  return (0);
19: }
20: @def_
21: @dir gcc.1-1
22: @file >??.c $macro(int, [INT_MIN, INT_MAX]) @file_
23: @file >??.c $macro(char, [CHAR_MIN, CHAR_MAX]) @file_
24: @dir_
```

図 3 testgen2 のテンプレート記述の例

表 1 テストスイートとテンプレートの行数比較

directory	testsuite	testgen	testgen2
gcc.1-*	10,049	3,109 (30.9%)	1,869 (18.6%)
gcc.2-1-*	62,696	21,266 (34.0%)	11,619 (18.5%)
gcc.2-2-*	33,066	10,452 (31.6%)	5,644(17.1%)
gcc.2-3-*	64,857	20,938 (32.3%)	10,893 (16.8%)
total	170,668	55,765 (32.7%)	30,025 (17.6%)

また, テストスクリプトの改良により, PASS/FAIL 判定の正確化と無限ループの監視・打切り処理を実現した.

5 むすび

本稿では, 新たなテンプレート構文を持つ testgen2 を提案し, これにより testgen では難しかったテストスイート中の重複コードの削減を実現した.

今後の課題として, テスト実行時間の短縮や testgen テストスイートを K&R 規格から C89, C99 に変更することが挙げられる. なお, testgen2 は今後 GPL にて公開する予定である.

謝辞

testgen2 のテンプレート記述圧縮の作業にご協力頂いた元関西学院大学の小渕慎二氏に感謝します. また, testgen2 の開発に関してご協力, ご討議頂いた関西学院大学石浦研究室の諸氏に感謝します.

参考文献

[1] 内山, 引地, 石浦, 永松: “C コンパイラ用テストスイートおよびその生成ツール testgen,” 信学技報 VLD2006-95 (Jan. 2007).

[2] 内山裕貴: “C コンパイラ用テストスイート生成のためのテストテンプレート記述,” 関西学院大学大学院理工学研究科修士論文 (Mar. 2008).