

文章编号:1673-0062(2016)01-0055-06

基于 MC/DC 的回归测试数据进化生成

韩莹,罗扬*,吴取劲,余童兰

(南华大学 计算机科学与技术学院,湖南 衡阳 421001)

摘要:在合理利用已有测试数据形成优势初始种群的前提下采用遗传算法自动生成回归测试数据是软件测试研究的一个热点.本文通过在已有测试数据的基础上依据 MC/DC 准则演进增补部分用例提升 MC/DC 覆盖率.首先,通过记录每个已有测试数据覆盖的条件组合确定要增补用例的目标条件组合,其次,根据适应度函数从已有测试数据中筛选出部分数据作为初始种群,再次,根据已筛选的部分初始种群所覆盖的条件组合与目标条件组合确定遗传操作分量,最后,演进并判定提取目标数据.理论与实验表明,该方法可以提高回归测试数据生成效率及代码覆盖率.

关键词:回归测试;遗传算法;适应度函数;初始种群;软件测试

中图分类号:TP311.5 **文献标识码:**B

Evolutionary Generation of the Regression Test Tata Based on MC/DC

HAN Ying, LUO Yang*, WU Qu-jin, YU Tong-lan

(School of Computer Science and Technology, University of South China,
Hengyang, Hunan 421001, China)

Abstract: Automatically generating regression test data using genetic algorithms is a hotspot in the field of software testing, the premise of which is to appropriately utilize the existing test data to form the initial population. In this paper, we added some test cases to the existing test data based on the MC/DC coverage criteria to improve the MC/DC coverage. By recording the condition combination coverage of each existing test data, we determined the target condition combinations for supplementation. And according to the fitness function we established the initial population. After that, based on the condition combinations by which the selected initial population covered and the target condition combinations we determined the ge-

收稿日期:2015-05-29

基金项目:湖南省科技厅自然科学省市联合基金资助项目(13JJ9008);湖南省教育厅科研基金资助项目(10CJ144)

作者简介:韩莹(1990-),女,山西运城人,南华大学计算机科学与技术学院硕士研究生.主要研究方向:软件测试.

* 通讯作者.

netic components.Finally,through evolution we judged and extracted the target data.Both the theoretical and experimental results show the benefits of this method to improve the efficiency of both the regression test data generation and code coverage.

key words: regression test;genetic algorithms;fitness function;initial population;software testing

0 引言

回归测试是软件开发和维护的重要组成部分,其数据的自动生成是软件回归测试研究的重点之一.目前经过学者的不断探索,已有许多回归测试用例自动生成的研究成果,如文献[1]通过UML(Unified Modeling Language)类图、顺序图、用例图生成回归测试用例,但由于UML建模设计的复杂性[2],作者对如何处理方法的前置条件和后置条件的变化,如何处理实际应用中一个场景可能对应多个顺序图的问题未予考虑或作了假定[1],是一种较为理想环境下的用例生成技术,文献[3]从另一个角度出发,把路径覆盖与concolic[4]测试技术结合,通过有效减少concolic测试路径,避免其生成和执行大量不能发现缺陷的测试输入[5],文献[6]把路径覆盖与遗传算法结合,利用已有测试数据生成优势初始种群[7],根据修改的程序块确定目标路径,文献[3]和文献[6]是在路径覆盖的基础上进行回归测试数据生成效率的提升.

然而实际程序中如果遇到一个判定中有多个条件时,仅仅基于路径覆盖不能考虑到判定中每个条件的覆盖情况,即很难检测到程序中的副作用函数,因而代码整体覆盖率存在提升空间.针对以上问题,本文提出基于MC/DC(Modified Condition/Decision Coverage)的回归测试数据进化生成,一方面可以有效提高测试数据的生成效率,另一方面可以提高测试数据对代码的覆盖率,增加测试数据发现错误的机率.

1 MC/DC 测试用例生成方法

MC/DC是覆盖率指标之一,适合测试逻辑表达式的逻辑覆盖情况,目前已成为航空航天产品软件测试的重要评价指标[8].一个MC/DC对是一对真值向量,这对真值向量使得判定语句有不同的结果,但结果的不同仅仅是由于真值向量中一个条件值的变化,MC/DC要求测试用例覆盖一组MC/DC对,使判定中每一个条件的所有可能结果至少出现1次,每一个判定本身的所有结果也至

少出现1次,每个入口与出口点至少被唤醒1次[9-10].

1.1 确定目的条件组合

对于一个具有N个条件的逻辑表达式,测试集合中至少包括N+1组元素[11].可以将逻辑表达式转化成语法树[12],采用文献[9]的MC/DC最小测试用例集快速生成算法,生成满足MC/DC法则的各判定的条件组合的集合D2.

例如表达式A||B&&C,其语法树如图1所示,此时最左节点为A,可以生成单独影响判定结果的MC/DC对,如表1所对应的真值向量号“1”、“2”,此时,B的默认结果为“假”,C的默认结果为“真”;置换A与B,此时最左节点为B,翻转B的默认结果,变成“真”,此时,A的结果应为“假”,C仍为“真”,如表1所对应的真值向量号“3”;将C与左子树置换,变成最左节点,翻转其默认结果变成“假”,而右子树的结果应为“真”,故A与B的结果分别为“真”和“假”,如表1所对应的真值向量号“4”,具体可参见文献[9].

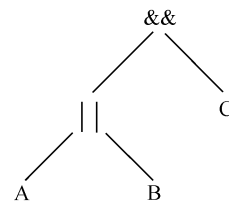


图1 语法树

Fig.1 Syntax tree

表1 满足MC/DC的条件组合

Table 1 The condition combination which meets MC/DC criterion

真值向量号	结果	条件组合			实际条件覆盖		
		A	B	C	A	B	C
1	T	T	F	T	T	N	T
2	F	F	F	T	F	F	N
3	T	F	T	T	F	T	T
4	F	T	F	F	T	N	F

程序在实际运行中,当A取“真”,B的取值不作判断,用‘N’表示,同理,当(A||B)的取值为“假”时,C的取值不作判断,用‘N’表示。(程序中如果遇到循环并且先“真”后“假”,全都用“TF”表示),表1包含了程序实际运行中的条件覆盖。

本文把原始程序p的测试数据作为已有测试数据,把修改后的程序p'所包含的满足MC/DC的条件组合的集合记为D2,把已有测试数据所覆盖的各判定的条件组合的集合D1与D2进行对比,将得到的差集作为目标条件组合的集合D,即 $D=D2-D1$ 。

1.2 构造适应度函数

个体的适应度值反映了个体在问题解空间中的适应能力,是遗传算法引导搜索的主要依据^[13]。本文确定的测试数据进化生成的目标是在已有数据的基础上增加测试数据,提高MD/DC覆盖率。Bueno等人曾指出,相似路径间的测试数据也有相似性^[14]。据此推断相似的判定组间的测试数据也有相似性。先把已有测试数据覆盖的条件组合转化成判定组合,进一步把已有测试数据覆盖的判定组合与目标进化个体覆盖的判定组合的相似度作为适应度函数。以下是相似度的计算方法。

记两个测试数据所覆盖的条件组合的节点序列p、q分别为 $c_{11}, c_{12}, c_{13}, c_{21}, c_{22}, \dots, c_{ij}$ 和 $c_{11}', c_{12}', c_{13}', c_{21}', c_{22}', \dots, c_{i'j}'$,其中,i(或i')表示程序中判定的个数,j(或j')表示程序中判定i(或i')所对应的条件的个数,那么,p、q所包含的节点数

分别为 $P = \sum_{n=1}^{n=i} j$,进一步对上述序列量化,规则为:找到序列p与q所对应的判定,如果相同判定所对应的节点完全相同,记为1,反之记为0,如果该判定所对应的所有变量或函数在之前的判定中已涉及并不在同一分支,记录方式与之前的判定一致,求和算出记录结果,记为sum,适应度函数可表示为:

$$fitness = \frac{sum}{\max\{P, P1\}}$$

1.3 初始种群及遗传操作分量的确定

根据1.2的适应度函数计算出已有测试数据的适应度,取平均值,将大于平均值的数据所对应的测试数据作为部分初始种群d1,d2,d3,d4……dn,其余初始种群随机生成。

考察以d1,d2,d3,d4……dn作为输入运行程

序p',覆盖的条件组合分别为 $p'(t_1), p'(t_2), p'(t_3), p'(t_4), \dots, p'(t_n)$,将这些条件组合与目标条件组合逐一匹配,如果在某节点处不相同,则该节点所对应的变量即可确定为遗传分量,直到匹配结束。

1.4 算法步骤

本文是在已有测试数据的基础上基于MC/DC法则增补部分测试数据使MC/DC覆盖达到最大,以下为算法步骤。

1)设置算法中的各参数;

2)利用已有测试数据运行程序p',记录各数据覆盖的条件组合 $p'(t_1), p'(t_2), p'(t_3), \dots$;

3)根据适应度函数,筛选出有用测试数据d1,d2,d3……,dn作为部分初始种群,其余个体随机生成;

4)以d1,d2,d3……,dn作为输入运行p',将其所覆盖的条件组合 $p'(t_1'), p'(t_2'), p'(t_3') \dots p'(t_n')$ 与目标条件组合逐一比对,把不相同的节点所对应的变量作为遗传操作分量;

5)判断算法是否满足终止条件(终止条件是进化达到最大进化代数或者生成的数据覆盖程序p'的目标条件组合);

6)对d1,d2,d3……dn或其子代个体实施遗传操作,为了简化,此处只对个体实施变异操作,对随机生成的个体,进行传统遗传操作,转到步骤5);

7)算法结束,输出测试数据。

2 实验

为了验证本文提出的方法在处理含有多个多条件判定的程序的有效性,将本文的方法与基于路径覆盖的回归测试数据进化生成方法应用于典型基准程序的测试,并且从有用测试数据占有率、有用测试数据个数以及测试数据的覆盖情况三方面对以上两种方法进行对比。程序采用C语言编写,在VC++6.0环境下运行,机器主频为2.80GHz,内存为2GB,测试软件为插件版parasofttest9.2。

2.1 测试程序及遗传策略的设置

本文选取一个典型的基准程序的子函数(sumday)作为测试程序,图2是修改后的程序p'及其CFG图,有3个整数作为输入数据,取值范围均为[0,3000],程序中包括8个主要节点,3个判定,8个简单条件。程序的修改点为'N2',修改前条件语句为if(month>12||month<1||day>

month_day[month]), 修改后' N2' 条件改为: if (month>12||month<1||day>month_day[month] ||day<1), 较之前增加了条件“ day<1”.

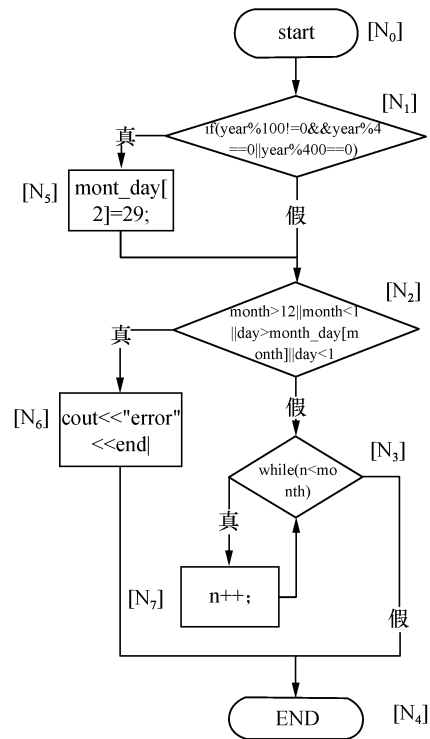
遗传策略: 基于路径覆盖的回归测试数据进化生成是采用节点序列匹配的方法^[15]对已有测试数据进行筛选从而形成初始种群, 利用遗传算法生成覆盖目标路径的测试数据, 即要增补的测

试数据; 本文基于 MC/DC 的回归测试数据进化生成使用的遗传算法, 对部分初始种群 $d_1, d_2, d_3 \dots, d_n$ 采用实数变异, 变异率为 0.2, 对随机生成的初始种群进行传统遗传操作, 其个体采用二进制编码, 遗传操作算子使用轮盘赌算法、单点交叉、单点变异, 交叉和变异的概率分别为 0.7, 0.2.

```

void sumday( intyear, intmonth, int day)
{
    int n = 1, month_day_sum = 0, sum = 0;
    intmonth_day[ 13 ] = { 0, 31, 28, 31, 30, 31, 30, 31,
                          31, 30, 31, 30, 31 };
    if( year%100! = 0&&year%4 = 0||year%400 = 0)
    {
        month_day[ 2 ] = 29;
        cout<<" 闰年" <<endl;
    }
    if( month>12||month<1||
        day>month_day[ month ] ||day<1)
        cout<<" error" <<endl;
    else
    { while( n<month)
      { month_day_sum += month_day[ n ];
        n++; }
      sum = month_day_sum+day;
      cout<<" sum " <<sum<<endl;
    }
}
    
```

a 示例程序 P'



b 该程序对应 CFG

图 2 某示例程序

Fig.2 An example program

2.2 基于 MC/DC 的回归测试数据进化生成

理论分析与实际测试得到已有测试数据对程序 p' 的路径覆盖率为 83%, MC/DC 覆盖率为 94%. 表 2 列出已有测试数据对 p' 的条件覆盖情况, 根据 1.1 的方法得到目标条件组合, 即为“F, N, F, F, F, F, T, N”, 图 3 描述了条件与变量之间的关系. 根据 1.3 所示的方法筛选出的有用测试数据为 (2100, 1, 7), (2100, 2, 30), (1100, 0, 0), 将其作为部分初始种群, 其遗传分量分别为 (month, day), (day), (month, day), 如表 2 所示.

表 2 优势初始种群及遗传分量

Table 2 The dominant initial population and genetic components

编号	输入数据	条件组合	相似度	GA 分量
01	2014 1 3	T, F, F, F, F, F, F, F	--	--
02	2100 1 7	F, N, F, F, F, F, F, F	0.33	month, day
03	1998 2 3	T, F, F, F, F, F, F, T, F	--	--
04	2100 2 30	F, N, F, F, F, T, N, N	0.33	day
05	1100 0 0	F, N, F, F, T, N, N, N	0.33	month, day
06	1992 2 3	T, T, N, F, F, F, F, T, F	--	--
07	2000 2 30	F, N, T, F, F, T, N, N	--	--
08	1988 13 0	T, T, F, T, N, N, N, N	--	--

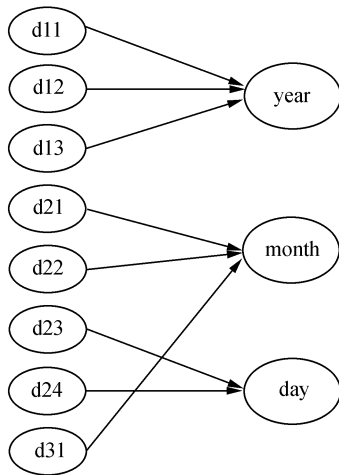


图 3 条件与变量之间的关系

Fig.3 The relationship between conditions and variables

的测试数据生成方法根据修改的条件可设置目标路径为 $N_0, N_1, N_2, N_3, N_7, N_3, N_7, N_3, N_4$, 表 3 记录了 20 次实验分别用两种方法得到的目标测试数据, 两种方法的种群大小均为 20, 表中显示的是种群中第一次出现的符合条件的目标测试数据. 表 4 统计了两种方法得到的测试数据的路径及 MC/DC 覆盖情况, 以及已有测试数据的利用率. 例如实验 1 中, 本文方法生成的测试数据为 (2100, 1, 0), 已有测试数据为 (2014, 1, 3), (2100, 1, 7), (1998, 2, 3), (2100, 2, 30), (1100, 0, 0), (1992, 2, 3), (2000, 2, 30), (1998, 13, 0), 实验得出本文方法生成的测试数据与已有测试数据对程序 p' 的路径覆盖率达到 83%, MC/DC 覆盖率达到 100%, 而对比方法生成的目标数据为 (2014, 3, 3), 它与已有测试数据对程序 p' 的路径覆盖率达到 83%, MC/DC 覆盖率达到 94%.

2.3 结果分析与对比

遗传策略的设置如 2.1 所述, 基于路径覆盖

表 3 两种方法生成的测试数据

Table 3 The test data generated by two methods

		1-10 次实验生成的有效增补数据									
本文方法	2100,1,0	2100,2,0	2100,1,0	1100,1,0	2100,1,0,	2100,6,0	2100,2,0	1100,4,0	1100,3,0	2100,4,0	
对比方法	2014,3,3	1998,3,3	2006,3,0	2100,3,7	1078,3,7	1998,3,3	1998,3,19	2510,3,7	2100,3,3	1998,3,9	
		11-20 次实验生成的有效增补数据									
本文方法	2100,2,0	2100,10,0	2100,2,0	1100,2,0	2100,2,0	2100,1,0	2100,2,0	2100,5,0	2100,8,0	2100,1,0	
对比方法	1998,3,3	2014,3,11	2014,3,0	2100,3,15	1934,3,3	1998,3,7	1994,3,3	478,3,3	1998,3,11	2100,3,7	

表 4 生成数据性能对比

Table 4 Comparison of the two methods

	有用测试数据占有率	有用测试数据个数	路径覆盖率	MC/DC 覆盖率
本文方法	100%	3	均为 83%	均为 100%
对比方法	100%	4	均为 83%	除实验 3,13 为 100%, 其余均为 94%

从表 3 可以看出, 本文方法和基于路径覆盖的方法均可在最大进化代数内找到目标测试数据; 从表 3 可以看出, 两种方法生成的测试数据均来自于已有测试数据, 所以有用测试数据占有率都为 100%; 本文方法从已有测试数据中筛选的有用测试数据有 3 个, 占已有测试数据的 38%, 基于路径覆盖的方法从已有测试数据中筛选的有用测试数据有 4 个, 占已有测试数据的 50%; 两种方法进化生成的测试数据加上已有测试数据对 p' 的路径覆盖率均为 83%; 本文方法得到的测试数

据加上已有测试数据对 p' 的 MC/DC 覆盖率均为 100%, 基于路径覆盖的方法进化生成的测试数据加已有测试数据对 p' 的 MC/DC 覆盖率除了实验 3 和实验 13 达到 100%, 其余均为 94%.

由上分析可知, 两种方法都可以充分利用已有测试数据提供的有用信息, 引导进化种群生成目标测试数据^[6], 而本文方法可以保障程序的 MC/DC 覆盖率达到最大. 所以, 当程序中存在多个判定中有两个或者两个以上的条件时, 使用本文方法可以提高测试数据对程序的覆盖率.

3 结 论

本文提出的基于 MC/DC 的回归测试数据进化生成,可以根据适应度函数利用已有测试数据筛选出有用测试数据作为部分初始种群,减少遗传代数.该方法考虑了程序中一个判定有多个条件的情况,解决了基于路径覆盖的数据进化生成方法没有考虑到的副作用函数的问题,有效提高了代码的整体覆盖率,进而增加了测试数据发现错误的机率.此外,采用基于 MC/DC 的回归测试数据进化生成,可以根据测试数据覆盖的条件确定遗传操作分量,减少个体需要改变的遗传操作分量的个数^[7],进一步减少遗传代数,提高测试数据生成效率.

本文提出的基于 MC/DC 进化生成回归测试数据时,默认 MC/DC 覆盖率可以达到 100%,但在某些特殊情况下,代码的 MC/DC 覆盖率不能达到 100%,所以快速检测程序中无法覆盖的条件组合是进一步研究的目标.

参考文献:

- [1] 姜承兵.基于 UML 设计的回归测试用例辅助生成研究[D].南京:南京航空航天大学,2007.
- [2] 李传煌,王伟明,施银燕.一种 UML 软件架构性能预测方法及其自动化研究[J].软件学报,2013,24(7):1512-1528.
- [3] 殷鹏川,贲可荣.基于路径引导的回归测试用例集扩增方法[J].计算机工程与科学,2014,36(11):2159-2163.
- [4] 王欣,郭涛,董国伟,等.基于补丁比較的 Concolic 测试方法[J].清华大学学报(自然科学版),2013,53(12):1737-1745.
- [5] 崔展齐,王林章,李宣东.一种目标制导的混合执行测试方法[J].计算机学报,2011,34(6):953-964.
- [6] 巩敦卫,任丽娜.回归测试数据进化生成[J].计算机学报,2014,37(3):489-499.
- [7] Maaranen H, Miettinen K, Penttinen A. On initial populations of a genetic algorithm for continuous optimization problems[J]. Journal of Global Optimization, 2007, 37(3):405-436.
- [8] Fang C R, Chen Z Y, Xu B W. Comparing logic coverage criteria on test case prioritization[J]. Science China Information Sciences, 2012, 55(12):2826-2840.
- [9] Kandl S, Chandrashekar S. Reasonability of MC/DC for safety-relevant software implemented in programming languages with short-circuit evaluation[J]. Computing, 2015, 97(3):261-279.
- [10] 张宇,张波,王俊杰,等.基于二叉树满足 MC/DC 测试用例设计方法[J].微计算机信息,2010,26(3):171-173.
- [11] 段飞雷,吴晓,张凡,等.MC/DC 最小测试用例集快速生成算法[J].计算机工程,2009,35(17):40-45.
- [12] Chilenski J J. An investigation of three forms of the modified condition decision coverage(MCDC) criterion[J]. Computer Programming and Software, 2001, 18(4):214-219.
- [13] 李龙澍,李森,廖敏,等.基于多种群遗传算法测试用例优先级技术研究[J].计算机技术与发展,2011,21(4):112-119.
- [14] Bueno P M S, Jino M. Automatic test data generation for program paths using genetic algorithms[J]. International Journal of Software Engineering and Knowledge Engineering, 2002, 12(6):691-709.
- [15] 谢晓园,徐宝文,史亮,等.面向路径覆盖的演化测试用例生成技术[J].软件学报,2009,20(12):3117-3136.