变异测试

下午1时9分

提纲

- 变异测试概述
- 变异测试流程
- 变异算子
- 变异测试工具
- 变异测试的成本
- 变异测试的研究问题

下午1时9分

提纲

- 变异测试概述
- 变异测试流程
- 变异算子
- 变异测试工具
- 变异测试的成本
- 变异测试的研究问题

下午1时9分

1. 变异测试概述

- 变异测试是一种对测试的充分性进行评估的技术，以创建更有效的测试用例集合
- 人为在被测系统中加入缺陷（变异），检查测试用例发现这些缺陷的能力，从而判断测试用例的有效性
- 与路径测试或数据流测试不同，没有测试数据的选择规则
- 与传统测试技术结合，可以提高测试效率

下午1时9分

1.1 基本思想

- 变异测试是一种 fault-based 的软件测试技术
- 原代码段

```
if(a&&b)  c = 1;
else c = 0;
```

变异后代码段

```
if(a || b)  c = 1;
else c = 0;
```

- 为了发现这个变异（杀死变异），测试数据必须对变异前后程序可能的不同状态覆盖，且c的值应该传播到程序输出，并被测试检查。
- 测试用例分析：
  - a=0, b=0和a=1, b=1都不能发现（杀死）这个变异
  - a=1, b=0或a=0, b=1可以发现（杀死）这个变异

下午1时9分

1.2 程序变异概念

- 给定一个程序P和一个测试用例集T，通过变异算子为P产生一组变异体MP={Pi, i=1,…,n}，对P和MP都使用T进行测试，如果某Pi在某个测试用例上与P产生不同的结果，则Pi中的变异被杀死；若某Pi在所有测试用例上都与P产生相同的结果，则Pi为活的变异体。
- 形式化描述：若Pi为P的变异体，T为测试用例集：
  - 对于某个测试用例t，P(t)≠Pi(t)，则称t杀死(kill)Pi中的变异；
  - 若T中的所有测试用例t，都使P(t)=Pi(t)，则称Pi是活的(live)变异体。

下午1时9分

例：求两个整数中的较大者

```
int max(int x, int y)
{   int mx;
    if(x>y)  mx=x;
    else mx=y;
    return mx;
}
```

原始程序P

```
int max(int x, int y)
{   int mx;
    if(x<y)  mx=x;
    else mx=y;
    return mx;
}
```

变异体Pi

测试用例x=3,y=4能够杀死变异体；

测试用例x=3,y=3不能杀死这个变异体。

下午1时9分

1.2 程序变异概念

- 活的变异体是否一定说明测试用例不够充分呢？

原始P

```
for(int i=0; i<10; i++)
{  sum+=a[i]; }
```

变异体P1

```
for(int i=0; i!=10; i++)
{  sum+=a[i]; }
```

- 若在P的输入域中，不存在测试用例t，能够使得P(t)≠Pi(t)，则P与Pi等价；
- 需要对活的变异体进行分析，检查其是否与P等价；
- 对不等价于P的变异体，需要设计新的测试用例，进行进一步测试，直到满足充分性度量（变异分数）。

下午1时9分

1.3 程序变异举例-1

原始P

```
int min(int x, int y)
{  int mx;
   mx=x;
   if(y<x)  mx=y;
   return mx;
}
```

变异体P1

```
int min(int x,int y)
{  int mx;
   mx=y;
   if(y<x)  mx=y;
   return mx;
}
```

变异体P2

```
int min(int x,int y)
{ int mx;
   mx=x;
   if(y<mx)  mx=y;
   return mx;
}
```

- **变量间替换变异，即用另外一个语法上合法的变量来替换当前变量。**
- **分析发现，P2是一个等价变异体。**

下午1时9分

1.3 程序变异举例-2

原始P

```
int min(int x, int y)
{  int mx;
   mx=x;
   if(y<x)  mx=y;
   return mx;
}
```

变异体P3

```
int min(int x,int y)
{  int mx;
   mx=x;
   if(y>x)  mx=y;
   return mx;
}
```

运算符间替换变异，即用另外一个语法上合法的运算符来替换当前运算符。

下午1时9分

提纲

- 变异测试概述
- 变异测试流程
- 变异算子
- 变异测试工具
- 变异测试的成本
- 变异测试的研究问题

下午1时9分

2 变异测试流程

下午1时9分

提纲

- 变异测试概述
- 变异测试流程
- 变异算子
- 变异测试工具
- 变异测试的成本
- 变异测试的研究问题

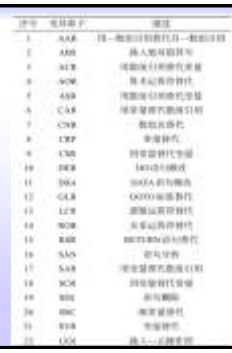
下午1时9分

3. 变异算子

- 变异算子：在符合语法规则前提下， 变异算子定义了从原有程序生成差别极小程序（即变异体） 的转换规则。
- 经验研究表明，程序员容易犯简单的非技术性错误。如将x<y+1错写成x<y等。
- 变异算子就是用来模拟典型的程序员编码错误，以及强制使某些表达式满足一定的条件（如使表达式的值为0）。
- 合理的变异算子有利于提高变异测试的效率

下午1时9分

Offutt和King针对Fortran77定义的22种变异算子：



下午1时9分

Lin Deng提出的针对Android APP定义的17种变异算子：

Category	Android Mutation Operator	Acronym
Event-based	Intent Payload Replacement	IPR
	Intent Target Replacement	ITR
	OnClick Event Replacement	ECR
	OnTouch Event Replacement	ETR
Component Lifecycle	Activity Lifecycle Method Deletion	MDL
	Service Lifecycle Method Deletion	SMDL
XML-related	Button Widget Deletion	BWD
	EditText Widget Deletion	TWD
	Activity Permission Deletion	APD
	Button Widget Switch	BWS
	TextView Deletion	TVD
Common Faults	Fail on Null	FON
	Orientation Lock	ORL
	Fail on Back	FOB
Context-aware	Location Modification	LCM
Energy-related	WakeLock Release Deletion	WRD
Network-related	Wi-Fi Connection Disabling	WCD

Lin Deng. MUTATION TESTING FOR ANDROID APPLICATIONS. PhD dissertation,2017

几个Android APP变异例子

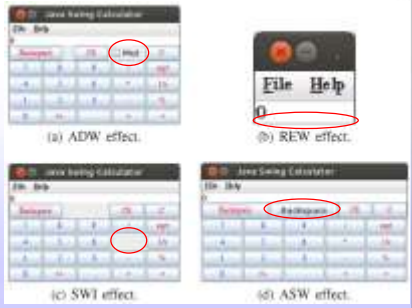
下午1时9分

Rafael A.P. Oliveira提出的针对GUI定义的18种变异算子：

Rafael A.P. Oliveira, etc. Definition and Evaluation of Mutation Operators for GUI-level Mutation Analysis. ICSTW 2015

几个GUI变异例子

原始界面



下午1时9分

提纲

- 变异测试概述
- 变异测试流程
- 变异算子
- 变异测试工具
- 变异测试的成本
- 变异测试的研究问题

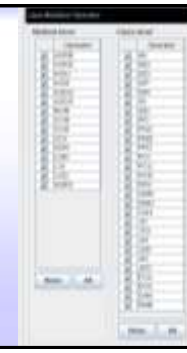
下午1时9分

4. 变异测试工具

- 研究人员针对不同语言，开发了不同的变异测试工具
  - Fortran：Mothra
  - C：Proteum, MILU, Insure++
  - Java：μJava, MuClipse, JESTER, JUMBLE, Javalanche
  - SQL：SQLMutation

下午1时9分

μJava的变异算子：



开源工具***μJava*(也叫muJava) 网址：**
https://cs.gmu.edu/~offutt/mujava/

下午1时9分

μJava中显示变异前后的程序：

下午1时9分

μJava中显示测试用例的变异分数：

下午1时9分

提纲

- 变异测试概述
- 变异测试流程
- 变异算子
- 变异测试工具
- 变异测试的成本
- 变异测试的研究问题

下午1时9分

5. 变异测试的成本

- 有研究表明，一个软件模块所能产生的变异体数量正比于程序中数据对象数量与数据对象的引用次数之乘积
- 由于每个变异体都至少要执行一个测试用例，一般需要运行多个测试用例，需要巨大的计算开销。这也阻碍了变异测试在工业界的广泛应用

下午1时9分

提纲

- 变异测试概述
- 变异测试流程
- 变异算子
- 变异测试工具
- 变异测试的成本
- 变异测试的研究问题

下午1时9分

6. 变异测试的研究问题

- 等价变异体检测问题：
  - 从生成的变异体中识别出等价变异体
  - 通过设计变异算子和分析被测程序特征，在变异体生成过程中避免等价变异体的生成。
- 等价变异体检测是一个不可判定问题，因此需要测试人员借助手工方式予以完成。等价变异体在语法层次上有微小的差别，但是在语义层次上是一致的。有研究人员发现，在生成的大量变异体中，等价变异体所占比例一般介于10%~40%。

下午1时9分

6. 变异测试的研究问题

- 变异测试分析优化：如何从生成的大量变异体中选择出典型变异体
  - 如果某些变异体可以被相似测试用例检测到，则只选择出典型变异体，其他变异体则被丢弃
  - 对变异算子进行约简来大规模缩小变异体数量，从而减小变异测试和分析开销
- 面向变异的测试用例生成技术，以提高测试用例的生成效率。
- 与工业界测试流程的结合：目前的一些原型工具并未考虑与软件企业已有的开发和测试流程紧密结合

下午1时9分

The End

Any Question?