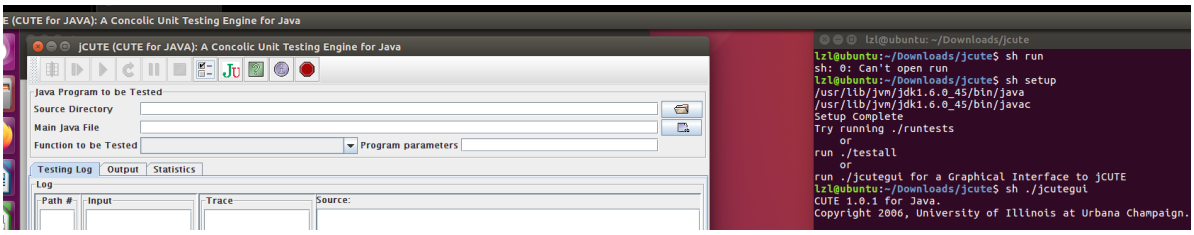
项目说明文档

本项目运行在linux上，通过执行脚本可以看到gui，也可以在命令行进行

```
sh ./run
sh ./jcutegui #启动gui
```



工具集成了两个脚本：jcutec和jcute

jcutec是对待测Java程序的编译器并生成可执行文件

jcute对被测程序的可执行文件进行混合测试

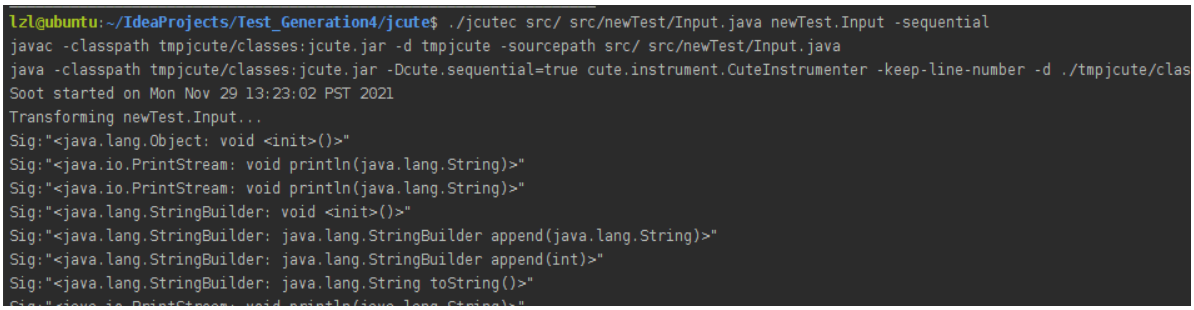
```
./jcutec SOURCE_DIRECTORY MAIN_JAVA_FILE MAIN_JAVA_CLASS -sequential #生成可执行文件
./jcute class -i iterations [-t DEBUG_LEVEL=0] [-m PATH_MODE=0] [-r] #根据参数不同输出不同，可输出约束路径和新测试数据
```

jcutec其后参数分别是源文件所在文件夹， 包含main函数的java文件，包含main函数的对应的class文件

jcute其后的参数分别是class文件 参数-i 迭代次数 参数-t输出debug信息

待测实验对象包括两大类，面向对象SUT和java_examples，在src下新建文件夹newTest与SUT，将提供的_java_examples下的文件移动到newTest文件夹下进行测试，以其中的input.java示例

```
cd jcute
./jcutec src/ src/newTest/Input.java newTest.Input -sequential
./jcute newTest.Input -i 100
```



```
lzl@ubuntu:~/IdeaProjects/Test_Generation4/jcute$ ./jcute newTest.Input -i 100
[Execution Path 1] java newTest.Input  -m 2
-------- In main!

-------- In <Init>(int)! i > 10

-------- In foo! Parameter = -1024
i <= 64

-------- In zoo! Parameters = 1, 2, 1.414
i <= 73
************************** One complete search is over ***********************
```

由于input.java文件输入全部为实际值，所以EGT即Execution-generated testing检测不到符号状态则会原样进行测试运行，这里使用另一个文件

如果需要得到约束路径，则jcute脚本执行语句后面加上参数-t 256

```
./jcute dtests.DExample1 -i 100 -t 256
```

```
[Execution Path 2] java dtests.DExample1 :-t:256
a1
a2
a3
a3
Error
Path Constriant
------------------
0 : pid = 0     mid = 0 nextPid = -1    nextMid = -1    isRace = false
1 : pid = 0     mid = 0 nextPid = 1     nextMid = 0     isRace = false
2 : pid = 0     mid = 1 nextPid = 0     nextMid = 1     isRace = false
3 : null
4 : pid = 0     mid = 0 nextPid = -1    nextMid = -1    isRace = false
5 : null
6 : 1.0 x1 + -2.0 x3 == 1.0
7 : postponed = []      thread = null   isRace = false
------------------
[Execution Path 3] java dtests.DExample1 :-t:256
a1
a2
a3
a3
************************** One complete search is over ***********************
```

本工具支持约束求解器lp_solve,但由于库和包装器针对libstdc++5.so在linux系统上是旧标准不可用了，阴恻需要下载编译lp_solve并编译lp_solve Java Wrapper，同时进行配置build文件，更改路径LPSOLVE_DIR和JDK_DIR,同时第二次g++调用链接到c版本的lp_solve,即得到lp_solve Java Wrapper

/net/mooctest/BPlusTreeTest.java [BPlusTree] - IntelliJ IDEA

build (~/Downloads/lp_solve_5.1_java/lib/linux) - gedit

Open Save

```
# ------------------------------------------------------------------
# This is a build file for the lp_solve Java wrapper stub library
# on linux platforms.
#
# Requirements and preconditions:
#
# - gcc and g++ compiler installed (I used gcc Version 3.3.1)
# - Sun JDK 1.4 installed
# - lp_solve linux archive (lp_solve5.tar.gz) unpacked
#
# Change the paths below this line and you should be ready to go!
# ------------------------------------------------------------------


LPSOLVE_DIR=/home/lzl/Downloads/lp_solve_5.1
JDK_DIR=/usr/lib/jvm/jdk1.6.0_45/

# OK, here we go!

SRC_DIR=../../src/c
INCL="-I $JDK_DIR/include -I $JDK_DIR/include/linux -I $LPSOLVE_DIR -I $SRC_DIR"

g++ -fpic $INCL -c $SRC_DIR/lpsolve5j.cpp
g++ -shared -Wl,-soname,liblpsolve51j.so -o liblpsolve51j.so lpsolve5j.o -lc -llpsolve51 -L/home/
lzl/Downloads/lp_solve_5.1/lpsolve51
```

```
./jcute dtests.DExample1 -i 100 -t 128
```

运行以上命令即运用lp_solve约束求解器进行求解，得到旧输入数据和新输入数据，下面是部分截图

```
a3
Old Input
------------------
Input
------------------
id = 0
 next = 1
 int 1
id = 1
 next = 2
 int 0
id = 2
 next = null
 int 0
------------------
************************* One complete search is over ***********************
New Input
------------------
Input
------------------
id = 0
 next = 1
```