

SOFTWARE TESTING [CS394]

FINAL PRESENTATION

Prof. Frankl
Spring 2013
NYU-POLY

TEST SUITE IMPLEMENTATION

Presented by
Omar English
Taylor

OVERVIEW

- Test suite introduction
 - Coverage test
- Describe application to be tested
- Describe test suite implementation
- Slide demonstration
- Conclusion

TEST SUITE INTRODUCTION

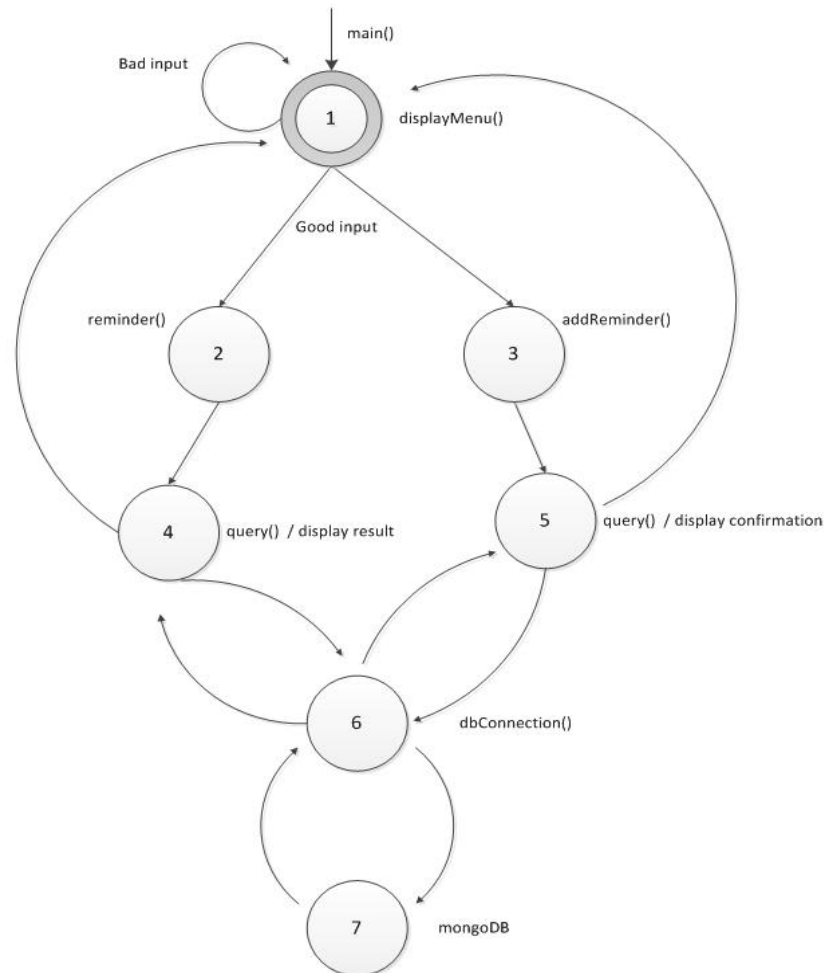
- Python test suite
- Node coverage
- Edge coverage
- Edge pair coverage
- Correctness of system behavior

APPLICATION TO BE TESTED

- Reminder Application
 - Calendar application similarity
 - Online auto-pay service similarity
 - Command line application
- Store and retrieve notifications
- MongoDB
 - Pymongo library

APPLICATION TO BE TESTED CONT.

Reminder App Flow



TEST SUITE IMPLEMENTATION

Test suite Layout

```
3 import insertDocument
4 import main
5 import query_pymongo
6 import addReminder
7 import db_connect
8 import getReminder
9 import menuDisplay
10
11 def _test_1():
12
13 def _test_main():
14
15 def _test_query_pymongo():
16
17
18 def _test_addReminder():
19
20
21 def _test_db_connect():
22
23
24 def _test_getReminder():
25
26
27 def test_menuDisplay():
28
29
30 if __name__ == '__main__':
31     for func in (val for key, val in vars().items()
32                 if key.startswith('test')):
33         func()
```

How to run test suite

- Import application functions
- Write test functions prefixed with “test”
 - Execute functions prefixed with “test”
 - Ignore, not prefixed with “test”

TEST SUITE IMPLEMENTATION CONT.

Expanded test suite function

```
def test_menuDisplay():
    #displayMenu(inputFromUser="", testMode=False):
    # 0 : success
    #1 : Unknown input
    value = menuDisplay.displayMenu("1", True)
    print "Returned Function Value:", value
    assert(0 == value), "main() failed"
```

- Testing functions depend on return value from tested function
- Return value indicate status of executed function
- displayMenu()
 - Returned value passed to assert function
 - Two return values 0(Success) 1(Unknown Input)
 - Enable test mode
 - testMode=False

displayMenu() Implementation

```
def displayMenu(inputFromUser="", testMode=False):

    if(testMode==False):
        print("\t\t***WELCOME TO REMINDER APP***\n")
        "(1) Check Today's Reminder\n"
        "(2) Add Reminder"
        userInput = raw_input("Enter menu option: ")

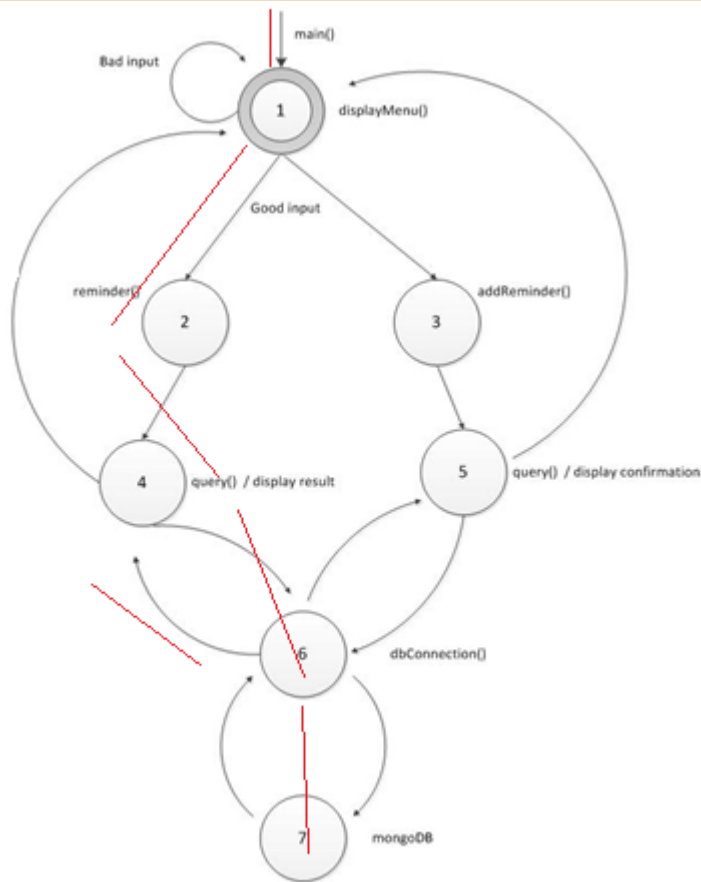
        while (userInput != "1") and (userInput != "2") and (userInput != "0"):
            print"ERROR: Unknown entry, try again \n"
            userInput = raw_input("Enter menu option: ")
        return userInput, 0

    if(testMode==True):

        print "Executed Function: displayMenu() \n"
        if(inputFromUser != "1"):
            return 0
        if(inputFromUser != "2"):
            return 0
        if(inputFromUser != "0"):
            return 0
        else:
            return 1
```

DEMONSTRATION 1

Test path for main()



- Test main()
- Test Paths= [{0, 1, 2, 4, 6,7,6, 4}]
- Edge pair coverage test requirement
 - TR = {[2,4,6]}
- Node 0 represents execution of main()

DEMONSTRATION 1 CONT.

Regular mode execution

```
$ ./main.py
          ****WELCOME TO REMINDER APP****
(1) Check Today's Reminder
(2) Add Reminder
Enter menu option: 1
Enter category name: school
Checking database

      _Reminder_
Description: CS Midterm
2013-04-02 04:00:00
          ****WELCOME TO REMINDER APP****
(1) Check Today's Reminder
(2) Add Reminder
Enter menu option: _
```

Test function for main()

```
def test_main():
    #main(displayUserInput="", categoryName="",nameReminder="",summaryReminder="",dateRemind="",testMode=False):
    # 0 : success
    # 1 : Reminder function returned bad value
    # 2 : addReminder function returned bad value

    value = main.main("1", "school","", "", "", True)
    print "Returned Function Value:",value
    assert(0== value), "main() failed"
```

DEMONSTRATION 1 CONT.

Test suite execution result

```
$ ./testSuite.py
Executed Function: main()

Executed Function: displayMenu()

Executed Function: reminder()

Executed Function: query()

Executed Function: dbConnection()

Access db: mongoDB

Executed Function: dbConnection()

Executed Function: query()

    _Reminder_
Description: CS Midterm
2013-04-02 04:00:00
Returned Function Value: 0
```

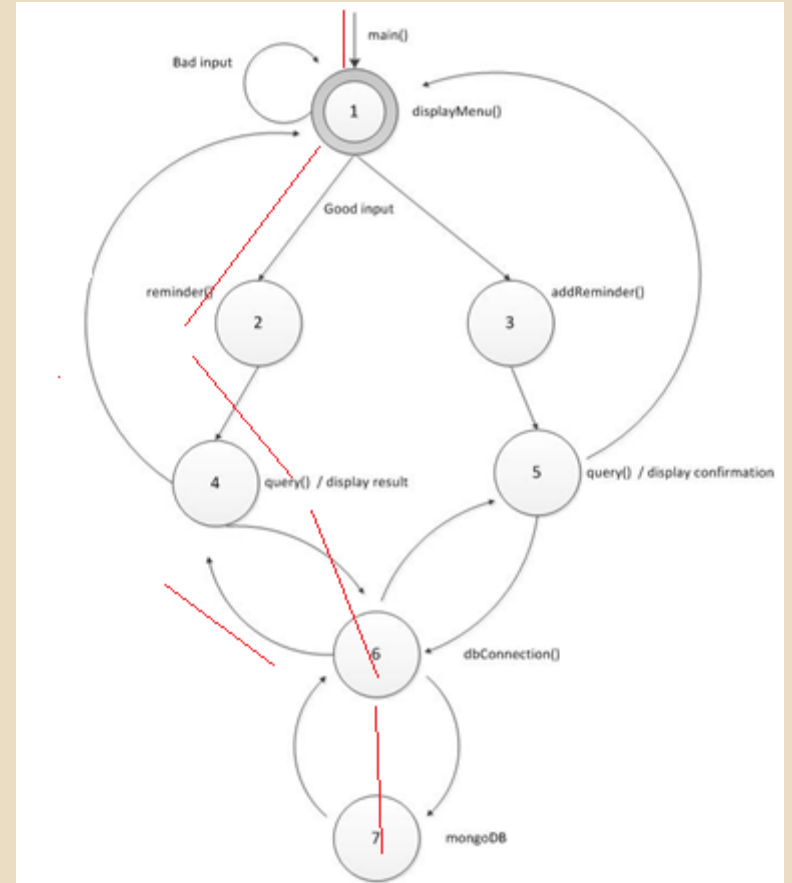
■ Test Paths

- {0, 1, 2, 4, 6, 7, 6, 4}

■ Test Requirement

- {2, 4, 6}

Test path for main()



DEMONSTRATION 2

reminder() implementation

```
def reminder(categoryName, testMode=False):
    if(testMode==True):

        print "Executed Function: reminder()\n"
        remind_doc={ "name": categoryName}
        executeStatus = query_pymongo.query(remind_doc, "1", testMode)

    if(testMode==False):
        category = raw_input("Enter category name: ")

        remind_doc={ "name": category}
        executeStatus = query_pymongo.query(remind_doc, "1", testMode)

    return executeStatus
```

test_getReminder() implementation

```
def test_getReminder():
    #reminder(categoryName, testMode=False):
    # 0 : success
    #1 : bad db connection passed into query()
    #2 : "Error IOError"
    #5: "query Type unknown"
    value = getReminder.reminder(2, True)
    print "Returned Function Value:", value
    assert(0== value), "getReminder() failed"
```

- Output for false assertions
- reminder()
 - Retrieve or add a reminder notification
 - Expects category name (string) of reminder
 - School, doctor, grocery etc.
 - Return four status codes
- test_getReminder()
 - pass an int value instead of string
 - Expects status code 5 to be returned
 - Unknown query type

DEMONSTRATION 2 CONT.

Test suite execution result

```
query Type unknown
Returned Function Value: 5
Traceback (most recent call last):
  File "./testSuite.py", line 78, in <module>
    func()
  File "./testSuite.py", line 65, in test_getReminder
    assert(0== value),"getReminder() failed"
AssertionError: getReminder() failed
```

- Return value is 5
 - Query Type Unknown

CONCLUSION

- Created a python test suite
 - Checks Edge pair coverage
 - Compare trace results to control flow graph
- Refine test suite
 - Check for statement coverage
 - Test multi-tier projects

END

THANK YOU