Topics in Software Quality

Assignment Sheet 3

Input Domain Testing and Combinatorial Testing

Question 1:

Derive input space partitioning tests for the LinkedList class with the following methods:

void LinkedList() – constructs an empty list

void add(int index, Object e) - adds the object e to the specified position in the list, throws exception if the list is too small

Object get(int index) – returns the element at the specified position in the list, returns Null if the list is too small

Boolean removeFirstOccurrence(Object e) – removes the first occurrence of the object e from the list, returns false if the object is not in the list

void clear() – removes all elements from the list

Int size() – returns the number of elements in the list

Propose possible input domain model(s) for the parameters and the state of the list.

Define a base case.

Create a test suite with base case coverage.

The test suite should check the correctness of the three methods add(i,e), get(i), and removeFirstOccurrence(e).

You can assume that the methods LinkedList(), clear() and size() have already been tested and are bug-free.

Question 2:

Given a function with ten Boolean inputs:

C_i is the set of all Boolean vectors of length 10 with exactly *i* 1s.

a. What is the coverage level of the test suite $C_0 U C_9$?
b. What is the coverage level of the test suite C_8?

Question 3:

What is the smallest theoretically possible test suite (minimum number of test cases) for a pairwise covering test suite of a function with 3 Boolean inputs, one parameter with three values and one parameter with 5 possible values?

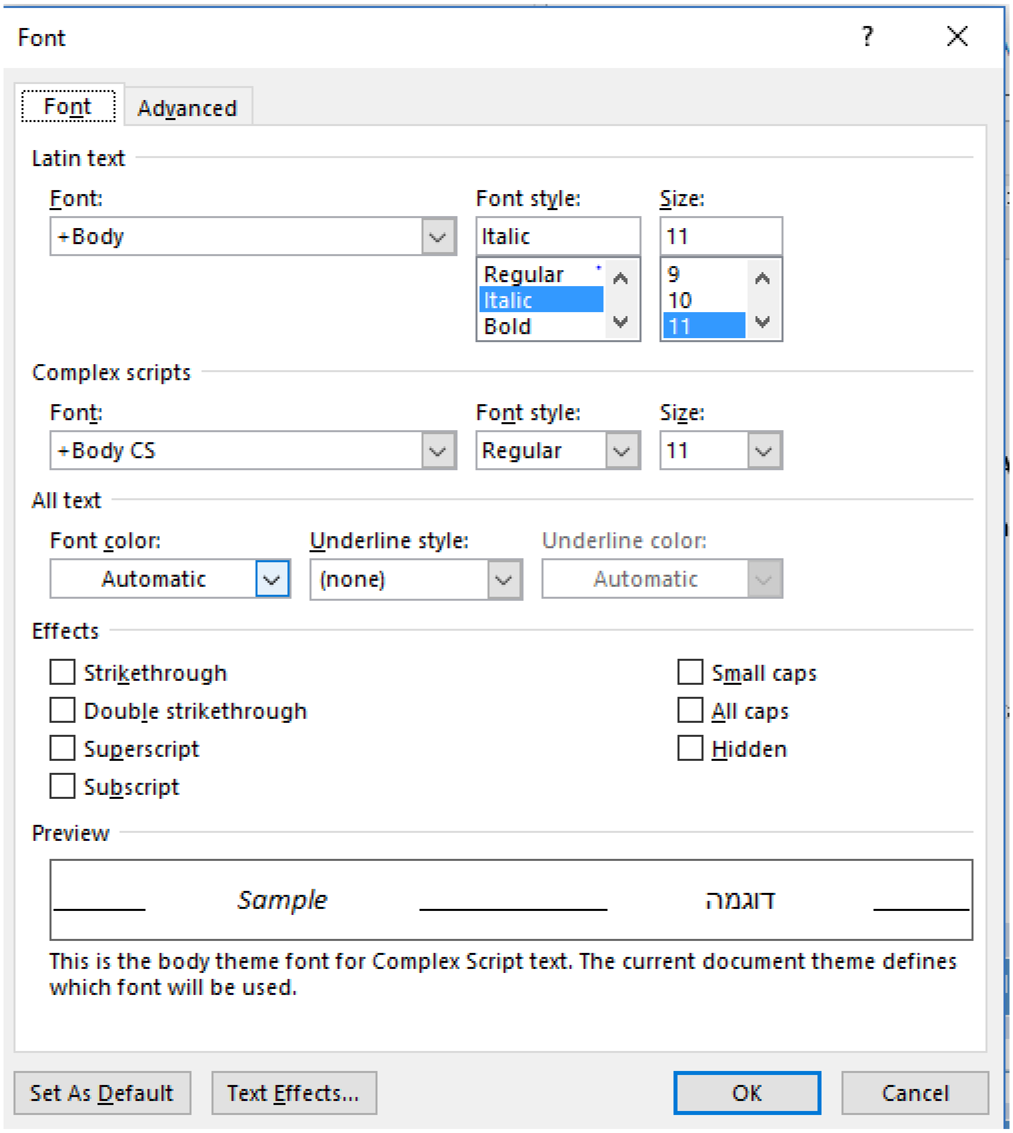
Can you create a test suite that achieves this minimum size?

Question 4:

Use the pairwiser site to create a set of 25 pairwise test cases for the illustrated user interface for editing fonts in MS Word. Suggest input partitions for the fields Font:, Font style: Size: Font color: Underline style: and Effects: (Ignore the section on Complex Scripts).

Your testing budget is at most 25 test cases. Describe the coverage you achieved (using the analysis provided by pairwiser). You should also use the constraints feature to describe the restrictions on the effects.

e.g. It is impossible to have both subscript and superscript at the same time.



Latin text

Font: +Body

Font style: Italic

Regular
Italic
Bold

Size: 11

9
10
11

Complex scripts

Font: +Body CS

Font style: Regular

Size: 11

All text

Font color: Automatic

Underline style: (none)

Underline color: Automatic

Effects

□ Strikethrough

□ Double strikethrough

□ Superscript

□ Subscript

□ Small caps

□ All caps

□ Hidden

Preview

Sample דוגמה

This is the body theme font for Complex Script text. The current document theme defines which font will be used.