ECE 322

SOFTWARE TESTING AND MAINTENANCE

Fall 2021

SOLUTIONS

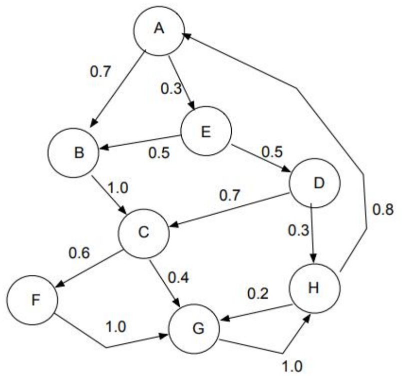
Assignment #4

Due date: Monday, November 1, 2021 by 11:00 PM

Total: 40 points

10 points

1. The behavior of a communication software system can be represented in the form of the finite state machine shown below. The figure shows the corresponding transition probabilities between the states A, B,....



Determine an order in which you would like to test individual states. (Hint: Markov Chain)

Solution

To determine stationary probabilities p_A, p_B... we have to solve a system of linear equations

$P\mathbf{p} = \mathbf{p}$

Where the matrix P is

P = [[0, 0.7, 0, 0, 0.3, 0, 0, 0],
[0, 0, 1.0, 0, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0.6, 0.4, 0],
[0, 0, 0.7, 0, 0, 0, 0, 0.3],
[0, 0.5, 0, 0.5, 0, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 1.0, 0],
[0, 0, 0, 0, 0, 0, 0, 1.0],
[0.8, 0, 0, 0, 0, 0, 0.2, 0]]

and $\mathbf{p}$ =[p_A p_B...p_H]

As a result, we have the following probabilities.

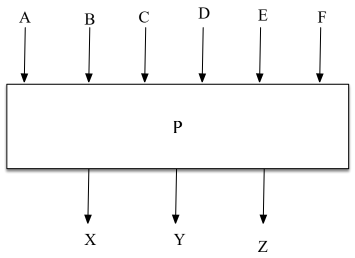
p = [0.15915541 0.13528338 0.15199982 0.02387413 0.04774682 0.09120173 0.1917904 0.19894831]

<u>Testing transitions</u>

Testing order: H, G, A, C, B, F, E, D

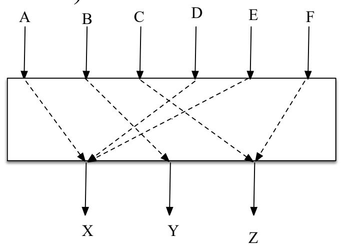
10 points

2. Consider the program P with 6 inputs A, B, C, D, E, F and 3 outputs X, Y, and Z. The input variables assume some values shown in the table below.



input	values
A	0, 1, 2, 3,4
B	A, B, C, D, E
C	100, 200,300
D	7, 8, 9, 10, 11
E	Y, N
F	α, β, γ

Consider the input-output testing strategy. Consider the situation when (i) you do not know functional dependencies among the input and output variables, and (ii) where these dependencies are known (as shown in the figure below).



What is the number of test cases required when using these two strategies. For (ii) propose a collection of test cases. In this strategy, how could you minimize the number of test cases; show an example.

Solution

Solution

(i)if functional dependencies are not known, the number of tests is $5^3*3^2*2 = 2250$.

(ii) for X one has 5*5*2 =50 test cases. For Y one has 5 test cases. For Z one has 3*3=9 test cases. In total, the number of test cases does not exceed 50+5+9 = 64 test cases.

The minimization of test cases can be achieved by selecting suitable values of not relevant inputs for the given output. For instance, for Z one can choose test cases

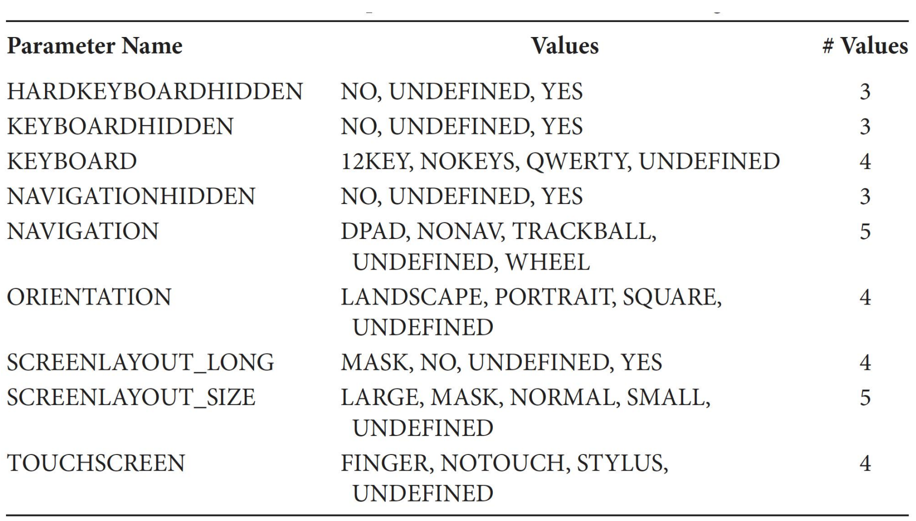
A	**B**	**C**	**D**	**E**	**F**
0	A	100	11	N	α
0	A	100	11	N	β
0	A	100	11	N	γ
0	A	200	11	N	α
0	A	200	11	N	β
0	A	200	11	N	γ
0	A	300	11	N	α
0	A	300	11	N	β
0	A	300	11	N	γ

Some other option could be

A	**B**	**C**	**D**	**E**	**F**
0	A	100	11	Y	α
0	A	100	11	U	β
0	A	100	11	Y	γ
0	A	200	11	Y	α
0	A	200	11	Y	β
0	A	200	11	Y	γ
0	A	300	11	Y	α
0	A	300	11	Y	β
0	A	300	11	Y	γ

10 points

3. For the configuration problem shown below, develop test cases using the strategy of pairwise testing. Compare the number of test cases obtained here with the number of tests required to consider all combinations of input values.
Hint: you can use the python package discussed in the lecture.



Parameter Name	Values	# Values
HARDKEYBOARDHIDDEN	NO, UNDEFINED, YES	3
KEYBOARDHIDDEN	NO, UNDEFINED, YES	3
KEYBOARD	12KEY, NOKEYS, QWERTY, UNDEFINED	4
NAVIGATIONHIDDEN	NO, UNDEFINED, YES	3
NAVIGATION	DPAD, NONAV, TRACKBALL, UNDEFINED, WHEEL	5
ORIENTATION	LANDSCAPE, PORTRAIT, SQUARE, UNDEFINED	4
SCREENLAYOUT_LONG	MASK, NO, UNDEFINED, YES	4
SCREENLAYOUT_SIZE	LARGE, MASK, NORMAL, SMALL, UNDEFINED	5
TOUCHSCREEN	FINGER, NOTOUCH, STYLUS, UNDEFINED	4

Solution

Using the software, the generated list of test cases is shown below:

```
0: ['NO', 'NO', '12KEY', 'NO', 'DPAD', 'LANDSCAPE', 'MASK', 'LARGE', 'FINGER']
1: ['UNDEFINED', 'UNDEFINED', 'NOKEYS', 'UNDEFINED', 'NONAV', 'PORTRAIT', 'NO', 'MASK', 'FINGER']
2: ['YES', 'YES', 'QWERTY', 'YES', 'TRACKBALL', 'SQUARE', 'UNDEFINED', 'NORMAL', 'FINGER']
3: ['YES', 'UNDEFINED', 'UNDEFINED', 'NO', 'UNDEFINED', 'UNDEFINED', 'YES', 'SMALL', 'NOTOUCH']
4: ['UNDEFINED', 'NO', 'UNDEFINED', 'YES', 'WHEEL', 'UNDEFINED', 'UNDEFINED', 'UNDEFINED', 'STYLUS']
5: ['NO', 'YES', 'NOKEYS', 'UNDEFINED', 'WHEEL', 'SQUARE', 'YES', 'UNDEFINED', 'UNDEFINED']
6: ['NO', 'UNDEFINED', 'QWERTY', 'YES', 'UNDEFINED', 'LANDSCAPE', 'NO', 'LARGE', 'UNDEFINED']
7: ['UNDEFINED', 'YES', '12KEY', 'NO', 'NONAV', 'SQUARE', 'MASK', 'SMALL', 'STYLUS']
8: ['YES', 'NO', 'NOKEYS', 'UNDEFINED', 'TRACKBALL', 'LANDSCAPE', 'MASK', 'MASK', 'NOTOUCH']
9: ['NO', 'NO', 'QWERTY', 'NO', 'NONAV', 'PORTRAIT', 'YES', 'NORMAL', 'NOTOUCH']
10: ['YES', 'YES', '12KEY', 'YES', 'DPAD', 'PORTRAIT', 'NO', 'UNDEFINED', 'NOTOUCH']
11: ['UNDEFINED', 'UNDEFINED', '12KEY', 'UNDEFINED', 'DPAD', 'UNDEFINED', 'UNDEFINED', 'NORMAL',
'UNDEFINED']
12: ['UNDEFINED', 'UNDEFINED', 'QWERTY', 'UNDEFINED', 'WHEEL', 'LANDSCAPE', 'MASK', 'SMALL', 'STYLUS']
13: ['NO', 'YES', 'UNDEFINED', 'UNDEFINED', 'UNDEFINED', 'PORTRAIT', 'UNDEFINED', 'LARGE', 'STYLUS']
14: ['UNDEFINED', 'NO', 'UNDEFINED', 'NO', 'TRACKBALL', 'SQUARE', 'NO', 'SMALL', 'UNDEFINED']
15: ['YES', 'YES', 'NOKEYS', 'YES', 'NONAV', 'UNDEFINED', 'UNDEFINED', 'MASK', 'UNDEFINED']
16: ['UNDEFINED', 'UNDEFINED', 'NOKEYS', 'NO', 'UNDEFINED', 'SQUARE', 'YES', 'LARGE', 'NOTOUCH']
17: ['NO', 'UNDEFINED', 'UNDEFINED', 'NO', 'TRACKBALL', 'UNDEFINED', 'MASK', 'UNDEFINED', 'FINGER']
18: ['YES', 'YES', 'UNDEFINED', 'NO', 'WHEEL', 'LANDSCAPE', 'UNDEFINED', 'NORMAL', 'NOTOUCH']
19: ['YES', 'NO', '12KEY', 'YES', 'UNDEFINED', 'PORTRAIT', 'MASK', 'NORMAL', 'UNDEFINED']
20: ['YES', 'YES', '12KEY', 'YES', 'WHEEL', 'PORTRAIT', 'YES', 'MASK', 'STYLUS']
21: ['NO', 'YES', 'NOKEYS', 'NO', 'DPAD', 'SQUARE', 'YES', 'MASK', 'STYLUS']
22: ['YES', 'YES', 'QWERTY', 'YES', 'DPAD', 'UNDEFINED', 'NO', 'SMALL', 'STYLUS']
23: ['YES', 'YES', 'UNDEFINED', 'NO', 'NONAV', 'LANDSCAPE', 'YES', 'UNDEFINED', 'FINGER']
24: ['YES', 'YES', 'UNDEFINED', 'NO', 'DPAD', 'UNDEFINED', 'UNDEFINED', 'LARGE', 'STYLUS']
25: ['NO', 'YES', 'NOKEYS', 'NO', 'TRACKBALL', 'PORTRAIT', 'YES', 'SMALL', 'STYLUS']
26: ['YES', 'YES', 'QWERTY', 'NO', 'WHEEL', 'PORTRAIT', 'NO', 'NORMAL', 'STYLUS']
27: ['YES', 'YES', 'QWERTY', 'NO', 'UNDEFINED', 'PORTRAIT', 'UNDEFINED', 'UNDEFINED', 'FINGER']
28: ['YES', 'YES', 'UNDEFINED', 'NO', 'WHEEL', 'PORTRAIT', 'UNDEFINED', 'SMALL', 'FINGER']
29: ['YES', 'YES', 'UNDEFINED', 'NO', 'WHEEL', 'PORTRAIT', 'UNDEFINED', 'MASK', 'STYLUS']
30: ['YES', 'YES', 'QWERTY', 'NO', 'WHEEL', 'PORTRAIT', 'UNDEFINED', 'MASK', 'STYLUS']
31: ['YES', 'YES', 'NOKEYS', 'NO', 'WHEEL', 'PORTRAIT', 'UNDEFINED', 'NORMAL', 'STYLUS']
32: ['YES', 'YES', '12KEY', 'NO', 'TRACKBALL', 'PORTRAIT', 'UNDEFINED', 'LARGE', 'STYLUS']
33: ['YES', 'YES', 'UNDEFINED', 'NO', 'WHEEL', 'PORTRAIT', 'UNDEFINED', 'LARGE', 'STYLUS']
34: ['YES', 'YES', 'UNDEFINED', 'NO', 'UNDEFINED', 'PORTRAIT', 'UNDEFINED', 'MASK', 'STYLUS']
35: ['YES', 'YES', 'UNDEFINED', 'NO', 'NONAV', 'PORTRAIT', 'UNDEFINED', 'LARGE', 'STYLUS']
```

When testing all combinations, we require $3^3 * 4^4 * 5^2 = 172,800$ test cases.

10 points

4. Propose test cases to carry out syntax-based testing for the following grammar

<expr>::= <id>|<num>|<expr><op><expr>
<id>::=<letter>|<letter><id>
<num>::=<digit>|<digit><num>
<op>::=*|-|/ |+
<letter>::=a|b|c|…|z|A|B|C|...|Z
<digit>::=0|1|2|…|9

so that the following coverage criteria are satisfied: (i) terminal symbol coverage, (ii) production coverage, and (iii) derivation coverage
How many test cases are required to meet the above coverage?

Solution
terminal symbol coverage: 52 for a, b, c,…z, A, B,...Z; 10 for 0, 1,2,..,9, and 4 for +,-, *, /. Overall 66 test cases
production coverage: 66 + 7 = 73
derivation coverage: infinite; see for instance <id>::=< letter><id> and <num>::=<digit><num>