CSE 5321

Homework 2

Fall 2019

Problem 1, 2 and 5 are 25 percent credit each. Problems 3-4 are 12.5 percent credit each. 100 percent total.

**HW Discussion schedule - questions only answered according to the following schedule**

**Problems 1 and 2 - before 9/22**

**Problem 3 and 4 - before 9/27**

**Problem 5 - before 10/3**

**Problem 1.**

Use Problem 6 from HW 1 to develop the following.

**Submit the following for this problem:**

1. The test case table - start with this first.
2. Sequence enumeration table
3. List the canonical states - in your pdf or word solution, not the table

Test Case Table

The format of the test case table is the following:



1. States are numbered S0 ... SN. The start event goes to state S0 as shown above.
2. For state S6, the unlock code test, you will need to figure out how to capture this based on the diagram. The inputs in the table above will help give you an idea.

Sequence enumeration table.

1. Use sequence enumeration to develop the canonical states. Show all sequences from length 0 to N. Note that I have pre-filled in the Length 0 response for you below.
2. Capture these in the attached table
3. For the "**Carry to next level**" column in the spreadsheet - use "Yes" or leave blank (for no).
4. Show all outputs for each - there will be no null responses
5. Mark each non-equivalence with a "-" enter in Excel as '-
6. Make a length 6a that is equivalent to State S6 in the test case table.



**Problem 2.**

You have 5 input parameters to a method that is used to compute the cost of a drink ordered in store as specified by the following table. The inputs are **Number of drinks**, **discount type**, **drink type**, **size**, and **payment type**.

|  |  |  |  |
| --- | --- | --- | --- |
| **Discount Type** | **Drink Type** | **Size** | **Payment Type** |
| None | Coffee | Small | Cash |
| Student | Tea | Regular | CreditCard |
| Teacher | Soda | Large | DebitCard |
| Military | Slushy | Grande |  |
| Frequent | Cappuccino |  |  |
| Coupon |  |  |  |
| Employee |  |  |  |

|  |  |
| --- | --- |
| **Size** | **Cost multiplier** |
| Small | 1.0 |
| Regular | 1.15 |
| Large | 1.35 |
| Grande | 1.85 |

|  |  |
| --- | --- |
| **DrinkType** | **Base Cost** |
| Coffee | $1.59 |
| Tea | $0.99 |
| Soda | $1.29 |
| Slushy | $1.19 |
| Cappuccino | $2.59 |

|  |  |
| --- | --- |
| **Payment Type** | **Surcharge** |
| Cash | -3% |
| CreditCard | 3% |
| DebitCard | 0% |

|  |  |
| --- | --- |
| **Discount Type** | **Discount Amount** |
| None | 0% |
| Student | 10% |
| Teacher | 5% |
| Military | 15% |
| Frequent | 20% |
| Coupon | 10% |
| Employee | 25% |

The unit test plan calls for the following values of **Number of Drinks** to be used.

|  |
| --- |
| **Number of Drinks** |
| 1 |
| 4 |
| 7 |
| 12 |
| 17 |
| 20 |
| 31 |
| 50 |

The method takes these four input parameters and returns the value of the following equation:

(number\_of\_drinks \* (1-DiscountType) \* DrinkType \* Size \* (1+PaymentType))\*1.0825

To simplify this problem, carry out the previous equation in Excel **without truncation**. The result is significant to the Cent.

Develop the test to fully test all pair-wise combinations of these four specified parameters. **Please use Excel to develop your expected outputs and supply your Excel test case tables with the homework submission. The function VLOOKUP in Excel could be helpful, but there are many simply ways to solve this problem.**

Your test case table is as follows.



Instructions

1. Download the allpairs tool from the following site <http://www.satisfice.com/tools/pairs.zip> and follow the instructions.
2. Use the following order of variables to define the inputs to the allpairs tool: **Number of drinks**, **discount type**, **drink type**, **size**, and **payment type**
3. Use the previous table (test case table) to develop the test cases needed to test all pairs.
4. Supply item 2 as a single test case table in Excel. You do NOT need to show the output of the allpairs tool - just the test cases it generates and in the order it provides.

**PLEASE MAKE SURE TO SAVE A COPY OF THE TEST CASE TABLE ABOVE AS A TAB DELIMITED TXT FILE. In Excel -> Save As... -> tab delimited txt file. This will allow the GTAs to use WinMerge to compare your test case table with the output. 50% deduction if not supplied.**

**Also, please make sure to show the enumeration values {cash, credit, debit} and not their actual values {-0.3%,0.3%,0.0%}**

**Problem 3.**

Minimize the following expressions using a K-map. Show all work including the K-map.

1. a'b'c + bc + bd + ac'
2. a'b'c'd' + a'b'cd + a'b +ac'd' + acd
3. a'b'c' + a'bd + a'bc' + abd + acd' + acd' + ab'cd
4. a'b'c' + a'b'd + c'd' + ac' + c'd + ab'cd
5. a'c' + abd + a'c + ac

**Problem 4**

For each of the following expressions develop the terms below. Make sure to reduce each to the minimum logical expression before solving. Reduce all answers also.

1. a'b' + c'
2. a'(b' + c)
3. abc + c'd' + a'b'
4. (a'b'c' XOR (a'b)') + abcd
5. The condition coverage, decision coverage, condition/decision coverage terms (one pair per coverage). Write solutions in terms of n-tuples - (FFF, FFT) as appropriate. Clearly indicate your answers for each. **FOR DECISION COVERAGE USE THE FIRST TERM AS FFF or FFFF**
6. The TOFs (Term Omission Faults) and TNFs (Term Negation Faults) for each. Separate each possible answer by a comma.

**Problem 5**

1) Use MC/DC logic and BV testing to determine the minimum test cases for each of the following requirements expressions. For each part, develop a test case table showing test case number, inputs, and expected outputs using the table as shown below.

1. a = (b < 10) || c
2. a = b || (c>=5)
3. a = (b <=5) & (c>=8)
4. a = (b>5) & (b<=15)

Express inputs in terms of numbers (for conditions with logical operators) and Booleans (for logical conditions) - e.g. the inputs are b (int) and/or c (int) when integer expressions are used, otherwise the inputs are Boolean.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Inputs** | | **Expected Outputs** |
| **Test Case** | **b** | **c** | **a** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |

2) Provide the UC MCDC solution for each expression (only 1 solution is needed for each).

1. a'b' + c'
2. a'(b' + c)
3. a + c'd' + a'b'
4. (a'b'c' XOR (a'b)') + abcd

3) Develop the MC/DC solutions for the following expression - 2 UC solutions and 1 Masking solution (that is not a UC solution). Show which are Masking and which are Unique Cause. a' + b'c + d