CSE 4321/5321

Homework 2

Spring 2020

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

**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. It will be very helpful to develop the state table first. Do this from the diagram.
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.
2. Show Boolean inputs and outputs as T or F
3. Show the Display as the text message without the double quote delimiters
4. For State S2 you only need to specify valid combinations not all 32 combinations

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 '-



**SOLUTION**

The test case table follows.



The sequence enumeration follows



The problem also asks students to identify the canonical sequences which are: **S, SN, SNX**

**Problem 2.**

From Problem 2 of HW 1 we need to develop a method that computes the ticket fare.

Your method has the inputs of:

1. LDT (e.g. 12:00:00 " am")
2. Day type (weekday, weekend , or Holiday)
3. Discount type

Fare (expected output) = Base Fare cost per ticket \* Rate Mult Factor \* (1-discount amount).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Base Fare** | | **Rate Mult** | | **Discount** | |
| **Volume** | **Cost/ticket** | **Day Type** | **Factor** | **Type** | **Amount** |
| Low Volume | $2.75 | Weekday | 1 | Standard | 0 |
| High Volume | $3.75 | Weekend | 0.8 | Student | 5% |
|  |  | Holiday | 1.5 | Senior | 10% |
|  |  |  |  | Veteran | 12.50% |
|  |  |  |  | Teacher | 8% |
|  |  |  |  | Worker | 50% |

Fare is a double truncated to the cent (represents currency), make sure to truncate each calculation. Because of the boundary values in the LDT (from Problem 1), the boss wants you to test all the boundary values of LDT as follows:



The boss does not want you to test all 180 possible combinations of unit test parameters. She has given you a challenge to reduce the number of tests to 60 because she has a tight test budget.

Your test case table is as follows.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case Number** | **Inputs** | | | **Exp Output** |
| **LDT** | **Day type** | **Discount type** | **Fare** |

Instructions

1. The allpairs tool is found at the following site <http://www.satisfice.com/tools/pairs.zip>
2. Use the output from the allpairs tool that I have provided **testcases.xlsx**
   1. Use these 60 test cases to develop the expected output for Fare as defined above
   2. a helpful function in Excel is VLOOKUP where you can define a 2-column table where the first column is the field to find and the second is the value to use. Hint: 12:00:00am would have a value of $2.75.
3. Supply the test case table in Excel. You do NOT need to show the output of the allpairs tool.

**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 {weekday, weekend, holiday} and not their actual values {1.0 ,0.8, 1.5}**

**SOLUTION**



**Problem 3.**

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

1. a'bd' + a'c'd+ ac'd + a'cd'
2. a’b’c'd' + ab’c’d + ab’cd + a’bcd + abc'd+ abcd
3. a'b' + a'bc'd + ac'd'
4. abc'd + a'b'c'd' + ab'c'd' + abc'd' + a'bc'd + ab'c'd + a'bc'd'
5. a’b’c’d' + ab’c’d' + ab’cd' + a'b'cd' + abc'd'+ a'b'c'd

**SOLUTION**



****

**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. a + b'c + d
4. (ab XOR cd) + 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.

SOLUTION

1. coverage terms:

**Condition**

**Decision**

**C/D**

**a.**

**a'b + c'**

FTT,TFF

FFF,FFT or FFF,TFT or

FFT,FFF or FFT,FTF, etc

FFF,TTT or FFT,TTF or

FTF,TFT

**b.**

**a'(b + c')**

FFT,TTF

FFF,FFT or FFF,TFF or

FFF,TFT or FFF,TTF,etc

FFF,TTT or FTF,TFT or

FTT,TFF

**c.**

**a + b'c + d**

FFFT,TTTF or FFTF,TTFT

or FFTT,TTFF or

FTFT,TFTF or FTTT,TFFF

FFFF,FFFT or FFFF,FFTF

or FFFF,FFTT or

FFFF,FTFT or

FFFF,FTTT, etc

FFFF,TTTT or FTFF,TFTT

or FTTF,TFFT

**d.**

**(ab XOR cd) + abcd**

**= ab + cd**

FFTT,TTFF or FTFT,TFTF

or FTTF,TFFT

FFFF,FFTT or FFFF,FTTT

or FFFF,TFTT or

FFFF,TTFF or

FFFF,TTFT, or

FFFF,TTTF, etc

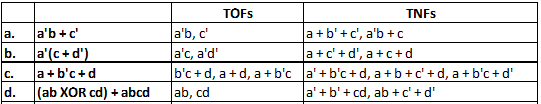
FFFF,TTTT or FFFT,TTTF

or FFTF,TTFT or

FTFF,TFTT ot FTTT,TFFF

**Coverage Levels**

2. TOFs/TNFs:



**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>=4)
3. a = (b <=8) & (c>8)
4. a = (b>=2) & (b<8)

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'(c + d')
3. a + b'c + d
4. (ab XOR cd) + 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. ab'c + d'

SOLUTION

1)

a.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Inputs** | | **Expected Output** |
| **b** | **c** | **a** |
| 1 | 10 | F | T |
| 2 | 10 | T | F |
| 3 | 9 | T | T |

b.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Inputs** | | **Expected Output** |
| **b** | **c** | **a** |
| 1 | T | 4 | T |
| 2 | T | 3 | F |
| 3 | F | 3 | T |

c.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Inputs** | | **Expected Output** |
| **b** | **c** | **a** |
| 1 | 9 | 9 | F |
| 2 | 8 | 9 | T |
| 3 | 8 | 8 | F |

d. This test requires 4 tests because there are 3 ECPs (we exclude extreme values).

|  |  |  |
| --- | --- | --- |
| **Test Case** | **Input** | **Expected Output** |
| **b** | **a** |
| 1 | 1 | F |
| 2 | 2 or 7 | T |
| 3 | 8 | F |
| 4 | 7 or 2 | T |

2)

a. a'b + c' - **only 1 UC solution is required.**

COI a -> XTT

COI b -> FXT

COI c -> TTX, TFX, or FFX

Base set = COI a U COI b = (TTT,FTT,FFT)

UC1 = (TTT,FTT,FFT,TTF) **or**

UC2 = (TTT,FTT,FFT,FFF)

b. a'(c + d') - **this is a strongly coupled condition with only one solution - see M06 slide 73. We write this in IDNF as a'c + a'd'**

COI c = FXT

COI d = FFX

"base set" = FFT, FTT, FFF

COI a = XTT, XFF (XFT makes both terms false, XTF makes both true)

Solution (note this is NOT UC): FFT,FTT,FFF,TTT,TFF

The strongly coupled condition requires that we use Strong MCDC which counts a' as two conditions and therefore 5 tests.

c. a + b'c + d- **only 1 UC solution is required.**

COI a -> XTTF, XTFF, or XFFF

COI b -> FXTF

COI c -> FFXF

COI d -> FTTX, FTFX, or FFFX

base set = COI b U COI c = (FFTF, FFFF, FTTF)

UC 1 = (FFTF, FFFF, FTTF,**TTTF,FTTT**) **or**

UC 2 = (FFTF, FFFF, FTTF,**TTTF,FFFT**) **or**

UC 3 = (FFTF, FFFF, FTTF,**TFFF,FTTT**) **or**

UC 4 = (FFTF, FFFF, FTTF,**TFFF,FFFT**)

d. reduces to ab + cd

COI a -> XTFT, XTTF, or XTFF

COI b -> TXFT, TXTF, or TXFF

COI a U COI b = (FTFT,TTFT,TFFT), (FTTF,TTTF,TFTF), or (FTFF,TTFF,TFFF)

COI c -> FTXT, TFXT, or FFXT

COI d -> FTTX, TFTX, or FFTX

COI c U COI d = (FTFT,FTTT,FTTF), (TFFT,TFTT,TFTF), or (FFFT,FFTT,FFTF)

UC MCDC = (COI a U COI b) U (COI c U COI d) but the solution must produce 5 test cases only

UC 1 = FTFT,TTFT,TFFT,FTTT,FTTF (first set from each)

UC 2 = FTFT,TTFT,TFFT,TFFT,TFTF (first set from a U b and second set from c U d)

UC 3 = FTTF,TTTF,TFTF,FTFT,FTTT (second term from a U b and first from c U d)

UC 4 = FTTF,TTTF,TFTF, TFFT,TFTT (second term from each)

3) ab'c + d'

COI a -> XFTT

COI b -> TXTT

COI c -> TFXT

COI d -> FFTX, TTTX, TFFX, FTTX, FFFX, TTFX, or FTFX

base set = COI a U COI b U COI c = (TFTT,FFTT,TTTT,TFFT)

There are 3 possible UC solutions - **only 2 are required**

UC 1 = (TFTT,FFTT,TTTT,TFFT,**FFTF**)

UC 2 = (TFTT,FFTT,TTTT,TFFT,**TTTF**)

UC 3 = (TFTT,FFTT,TTTT,TFFT,**TFFF**)

The masking solution uses one of the unused COI d terms - we also know that since COI d has been T in all the base set tests, it must be false here. There are four possible Masking solutions **only 1 is required**

M1 = (TFTT,FFTT,TTTT,TFFT,**TTFF**) or

M2 = (TFTT,FFTT,TTTT,TFFT,**FTTF**) or

M3 = (TFTT,FFTT,TTTT,TFFT,**FTFF**) or

M4 = (TFTT,FFTT,TTTT,TFFT,**FFFF**) or