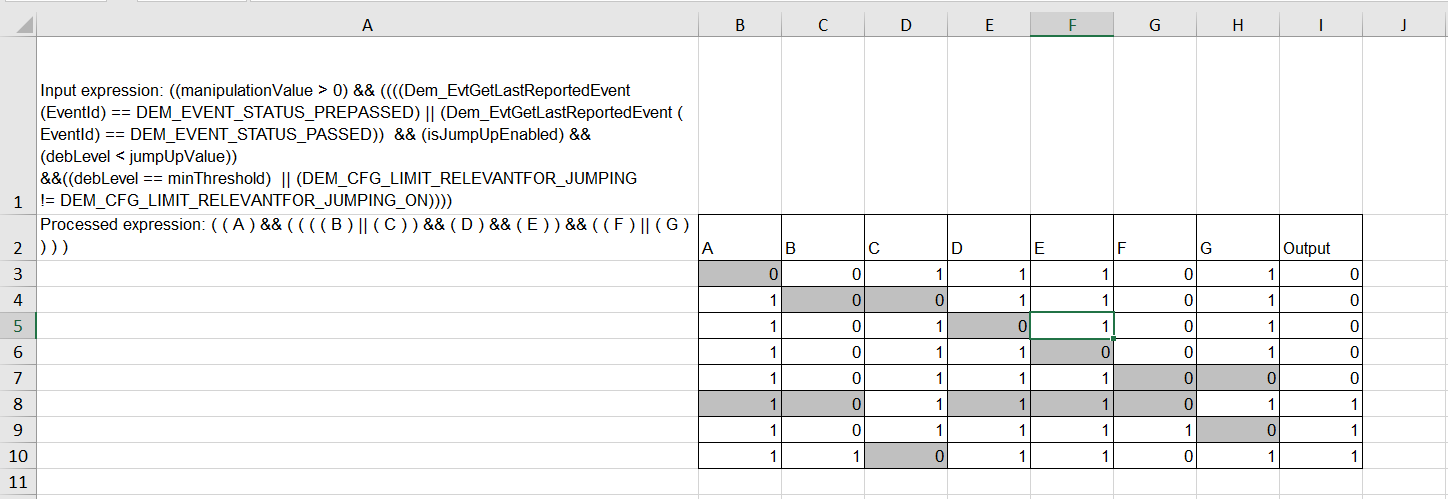
1. What is it ?
2. How does it work ?
   1. A short look-back to MC/DC and Unique Cause
   2. Algorithm base
3. Limitations and blockers
4. What is it ?

If you are doing Unit Testing, you must be very familiar with the MC/DC criteria. Sometimes it costs you many times with complicated and confused condition expression. The more conditions the expression has, the more test cases you must create to get it fully covered.

It will not be a problem with just up to 3 or 4 conditions within a statement, but let us take an example for this statement in *Dem\_Deb.c*:

|  |
| --- |
| **if** ((manipulationValue > 0) && ((((Dem\_EvtGetLastReportedEvent (EventId) == DEM\_EVENT\_STATUS\_PREPASSED) || (Dem\_EvtGetLastReportedEvent (EventId) == DEM\_EVENT\_STATUS\_PASSED)) && (isJumpUpEnabled) && (debLevel < jumpUpValue))&&((debLevel == minThreshold) || (DEM\_CFG\_LIMIT\_RELEVANTFOR\_JUMPING != DEM\_CFG\_LIMIT\_RELEVANTFOR\_JUMPING\_ON)))) |

No need to analyze the logic or design test cases anymore, this tool will do all those stuffs for you. A report shows you the minimum number of test cases you should use with detailed independence pairs for each condition:



Spending just 10 seconds to use this tool **or** more than half an hour to draw the truth table of 7 conditions (27 = 128 lines) and find the results **in desperation**, which do you prefer ?

1. How does it work ?
   1. A short look-back to MC/DC and Unique-Cause
   2. Algorithm base

Let’s take a detail condition expression to explain the algorithm better:

|  |
| --- |
| **( (A || B) && (C || D) )** |

Step 1: Create the truth table for all condition is the expression

* This expr has 4 conditions so the truth table will count from 0 – 15

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Decimal | A | B | C | D | Output |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 2 | 0 | 0 | 1 | 0 | 0 |
| 3 | 0 | 0 | 1 | 1 | 0 |
| 4 | 0 | 1 | 0 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 1 |
| 7 | 0 | 1 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 | 1 |
| 10 | 1 | 0 | 1 | 0 | 1 |
| 11 | 1 | 0 | 1 | 1 | 1 |
| 12 | 1 | 1 | 0 | 0 | 0 |
| 13 | 1 | 1 | 0 | 1 | 1 |
| 14 | 1 | 1 | 1 | 0 | 1 |
| 15 | 1 | 1 | 1 | 1 | 1 |

Step 2: Find all independence pairs

* With each condition, find all of its independence pairs in the decimal form.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Decimal | A | B | C | D | Output | Make A unique | Make B unique | Make C unique | Make D unique |
| 0 | 0 | 0 | 0 | 0 | 0 | x | x | x | x |
| 1 | 0 | 0 | 0 | 1 | 0 | 9 | 5 | x | x |
| 2 | 0 | 0 | 1 | 0 | 0 | 10 | 6 | x | x |
| 3 | 0 | 0 | 1 | 1 | 0 | 11 | 7 | x | x |
| 4 | 0 | 1 | 0 | 0 | 0 | x | x | 6 | 5 |
| 5 | 0 | 1 | 0 | 1 | 1 | x | 1 | x | 4 |
| 6 | 0 | 1 | 1 | 0 | 1 | x | 2 | 4 | x |
| 7 | 0 | 1 | 1 | 1 | 1 | x | 3 | x | x |
| 8 | 1 | 0 | 0 | 0 | 0 | x | x | 10 | 9 |
| 9 | 1 | 0 | 0 | 1 | 1 | 1 | x | x | 8 |
| 10 | 1 | 0 | 1 | 0 | 1 | 2 | x | 8 | x |
| 11 | 1 | 0 | 1 | 1 | 1 | 3 | x | x | x |
| 12 | 1 | 1 | 0 | 0 | 0 | x | x | 14 | 13 |
| 13 | 1 | 1 | 0 | 1 | 1 | x | x | x | 12 |
| 14 | 1 | 1 | 1 | 0 | 1 | x | x | 12 | x |
| 15 | 1 | 1 | 1 | 1 | 1 | x | x | x | x |

* Based on the table above:
* A has 3 independence pairs: (1, 9); (2, 10); (3, 11)
* B has 3 independence pairs: (1, 5); (2, 6); (3, 7)
* C has 3 independence pairs: (4, 6); (8, 10); (12, 14)
* D has 3 independence pairs: (4, 5); (8, 9); (12, 13)
* We have the independence table below:

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | D |
| 1, 9 | 1, 5 | 4, 6 | 4, 5 |
| 2, 10 | 2, 6 | 8, 10 | 8, 9 |
| 3, 11 | 3, 7 | 12, 14 | 12, 13 |

Step 3: Independence pairs Union

* The mission of this step is to find all sets (tập hợp) which contains a number of test cases satisfy MC/DC.
* As each pair from one column satisfies the coverage for one condition, the combination of any four pairs from 4 columns will be a “MC/DC set”.
* To implement this idea, we use a mathematics theory called **SET UNION** (∪)
* For example: (1, 9) ∪ (1,5) ∪ (4, 6) ∪ (4, 5) = (1, 4, 5 ,6, 9) 🡪 this set of Test Cases will satisfies MC/DC
* Another example: (1, 9) ∪ (3, 7) ∪ (8, 10) ∪ (12, 13) = (1, 3, 7, 8, 9, 12, 13) 🡪 this set also satisfies, but has more TCs the above
* Because we do not know when we can find the shortest “MC/DC set”, we must loop this step to find all possible sets.
* We have 4 conditions, each has 3 conditions so the number of MC/DC sets founded will be: 3 x 3 x 3 x 3 = 81 sets

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [1, 4, 5, 6, 9],  [1, 4, 5, 6, 8, 9],  [1, 4, 5, 6, 9, 12, 13],  [1, 4, 5, 8, 9, 10],  [1, 5, 8, 9, 10],  [1, 5, 8, 9, 10, 12, 13],  [1, 4, 5, 9, 12, 14],  [1, 5, 8, 9, 12, 14],  [1, 5, 9, 12, 13, 14], | [1, 2, 4, 5, 6, 9],  [1, 2, 4, 6, 8, 9],  [1, 2, 4, 6, 9, 12, 13],  [1, 2, 4, 5, 6, 8, 9, 10],  [1, 2, 6, 8, 9, 10],  [1, 2, 6, 8, 9, 10, 12, 13],  [1, 2, 4, 5, 6, 9, 12, 14],  [1, 2, 6, 8, 9, 12, 14],  [1, 2, 6, 9, 12, 13, 14], | [1, 3, 4, 5, 6, 7, 9],  [1, 3, 4, 6, 7, 8, 9],  [1, 3, 4, 6, 7, 9, 12, 13],  [1, 3, 4, 5, 7, 8, 9, 10],  [1, 3, 7, 8, 9, 10],  [1, 3, 7, 8, 9, 10, 12, 13],  [1, 3, 4, 5, 7, 9, 12, 14],  [1, 3, 7, 8, 9, 12, 14],  [1, 3, 7, 9, 12, 13, 14], | [1, 2, 4, 5, 6, 10],  [1, 2, 4, 5, 6, 8, 9, 10],  [1, 2, 4, 5, 6, 10, 12, 13],  [1, 2, 4, 5, 8, 10],  [1, 2, 5, 8, 9, 10],  [1, 2, 5, 8, 10, 12, 13],  [1, 2, 4, 5, 10, 12, 14],  [1, 2, 5, 8, 9, 10, 12, 14],  [1, 2, 5, 10, 12, 13, 14], | [2, 4, 5, 6, 10],  [2, 4, 6, 8, 9, 10],  [2, 4, 6, 10, 12, 13],  [2, 4, 5, 6, 8, 10],  [2, 6, 8, 9, 10],  [2, 6, 8, 10, 12, 13],  [2, 4, 5, 6, 10, 12, 14],  [2, 6, 8, 9, 10, 12, 14],  [2, 6, 10, 12, 13, 14], | [2, 3, 4, 5, 6, 7, 10],  [2, 3, 4, 6, 7, 8, 9, 10],  [2, 3, 4, 6, 7, 10, 12, 13],  [2, 3, 4, 5, 7, 8, 10],  [2, 3, 7, 8, 9, 10],  [2, 3, 7, 8, 10, 12, 13],  [2, 3, 4, 5, 7, 10, 12, 14],  [2, 3, 7, 8, 9, 10, 12, 14],  [2, 3, 7, 10, 12, 13, 14], | [1, 3, 4, 5, 6, 11],  [1, 3, 4, 5, 6, 8, 9, 11],  [1, 3, 4, 5, 6, 11, 12, 13],  [1, 3, 4, 5, 8, 10, 11],  [1, 3, 5, 8, 9, 10, 11],  [1, 3, 5, 8, 10, 11, 12, 13],  [1, 3, 4, 5, 11, 12, 14],  [1, 3, 5, 8, 9, 11, 12, 14],  [1, 3, 5, 11, 12, 13, 14], | [2, 3, 4, 5, 6, 11],  [2, 3, 4, 6, 8, 9, 11],  [2, 3, 4, 6, 11, 12, 13],  [2, 3, 4, 5, 6, 8, 10, 11],  [2, 3, 6, 8, 9, 10, 11],  [2, 3, 6, 8, 10, 11, 12, 13],  [2, 3, 4, 5, 6, 11, 12, 14],  [2, 3, 6, 8, 9, 11, 12, 14],  [2, 3, 6, 11, 12, 13, 14], | [3, 4, 5, 6, 7, 11],  [3, 4, 6, 7, 8, 9, 11],  [3, 4, 6, 7, 11, 12, 13],  [3, 4, 5, 7, 8, 10, 11],  [3, 7, 8, 9, 10, 11],  [3, 7, 8, 10, 11, 12, 13],  [3, 4, 5, 7, 11, 12, 14],  [3, 7, 8, 9, 11, 12, 14],  [3, 7, 11, 12, 13, 14] |

Step 4: Find the shortest MC/DC sets

* From the list of those sets, there are 2 sets with the minimum length: (1, 4, 5, 6, 9) and (2, 4, 5, 6, 10).
* The founded minimum length is 5, which is also the ideal number of test cases need (With N conditions, the minimum test case needed is N+1)
* One in those two set will be chosen and put in the report.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Decimal | A | B | C | D | Output |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 4 | 0 | 1 | 0 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 1 |
| 9 | 1 | 0 | 0 | 1 | 1 |

1. Limitations and Blockers

* The tool can solve the expression with up to 16 conditions. With larger expression, it will take a lot of times to solve cause the step SET UNION uses a lot of PC memory resource. But this limitation is also impractical: No developer would like to use a more-than-sixteens-condition expression in their code.
* With complicated expression, each condition should be put in a bracket so the tool can identify it separately. It cannot solve this expression:

|  |
| --- |
| if ( !Dem\_EvtAllEnableConditionsFulfilled(EventId) ||!Dem\_IsOperationCycleStarted(Dem\_EvtGetOperationCycleId(EventId)) ||Dem\_EvtIsSuppressed(EventId)|| !Dem\_IsEventReportingEnabledByDtcSetting(EventId)) |

* User should spend a little bit efforts to edit it:

|  |
| --- |
| if ( (!Dem\_EvtAllEnableConditionsFulfilled(EventId))||(!Dem\_IsOperationCycleStarted(Dem\_EvtGetOperationCycleId(EventId)))||(Dem\_EvtIsSuppressed(EventId))|| (!Dem\_IsEventReportingEnabledByDtcSetting(EventId))) |