Test cases for Decision Coverage and Modified Condition / Decision Coverage

Zalán Szűgyi

Eötvös Loránd University,
Faculty of Informatics

2008

Table of Contents

1 Introduction

Coverage refers to the extent to which a given verification activity has satisfied its objectives. Coverage measures can be applied to any verification activity, although they are most frequently applied to testing activities. Appropriate coverage measures give the people doing, managing, and auditing verification activities a sense of the adequacy of the verification accomplished. [1]

The code coverage analysis contains three main steps [2], such as: finding areas of a program not exercised by a set of test cases, creating additional test cases to increase coverage and determining a quantitative measure of code coverage, which is an indirect measure of quality. Optionally it contains a fourth step: identifying redundant test cases that do not increase coverage.

The code coverage analysis is a structural testing technique (white box testing), where it compares test program behavior against the apparent intention of the source code. This contrasts with functional testing (black box testing), which compares test program behavior against a requirements specification. Structural testing examines how the program works, taking into account possible pitfalls in the structure and logic. Functional testing examines what the program accomplishes, without regard to how it works internally.

In this study we concern to structural testing methods, especially which are related to Decision Coverage (DC), and Modified Condition / Decision Coverage (MCDC). These coverage metrics are discussed in the next chapter. We analyze several projects – written in Ada programming language – in subprogram level, and estimate how many test cases are needed to satisfy the 100% of DC and MCDC coverage. At last we answer to the question: how many test cases need more to satisfy MCDC then DC.

In the second chapter we describe the most frequently used coverage metrics. In the third chapter we give a detailed description about how we analyzed the source codes of projects. Then we discuss the results of our analysis in the fourth chapter. And the summary and the conclusion comes in the fifth chapter.

2 The coverage metrics

In this chapter we briefly describe the most frequently used coverage metrics.

2.1 Statement Coverage

To achieve statement coverage, every executable statement in the program is invoked at least once during software testing. The main advantage of this method is that it can be applied directly in object code and does not necessary to process source code. But this method is insensible to some control structure. Let us see the following example:

```
      T* t = NULL;
      if (condition)
          t = new T();
      t->method();
```

In this example only one test case (where the condition is true) is enough to achieve 100% statement coverage because every statement invoked once. It that case our program works fine, and we recognize it faultless. But in the real usage, the condition can be false, and it causes indeterministic behavior or segmentation fault.

2.2 Decision Coverage

This method requires that every decision must be evaluated to true and false. In this case the error can be seen in the previous example turns out in testing time. This metric has the advantage of simplicity without the problems of statement coverage. A disadvantage is that this metric ignores branches within boolean expressions which occur due to short-circuit operators. Let us see to following example:

```
      if A or B then
            true_statement;
      else
            false_statement;
      end if;
```

Two test cases where (A = true, B = false and A = false, B = false) can satisfy the requirements of Decision Coverage. But the effect of B is not tested, so these test cases cannot distinguish between the decision (A or B) and the decision A.

2.3 Condition Coverage

This method requires that every condition in decision take on all possible outcomes at least once. This solves the problem in previous example. But it does not require that the decision evaluated to both true and false. For example, the test cases where A = true, B = false and A = false, B = true satisfy the requirements of Condition Coverage in previous example, but the decision outcomes always true.

2.4 Condition / Decision Coverage

This is a mixture of Condition and Decision Coverage. So the test cases to satisfy the requirements of Condition / Decision Coverage when satisfy the requirements of Condition Coverage and Decision Coverage. The test cases A = true, B = true and A = false, B = false in example from chapter 2.2 meet the coverage criterion. However, these two test cases do not distinguish the correct expression (A or B) from the expression A or from the expression B or from the expression (A and B).

2.5 Multiple Condition Coverage

Multiple Condition Coverage requires test cases that ensure each possible combination of inputs to a decision is executed at least once; that is, multiple condition coverage requires exhaustive testing of the input combinations to a decision. In theory, multiple condition coverage is the most desirable structural coverage measure; but, it is impractical for many cases. For a decision with n inputs, multiple condition coverage requires 2^n tests.

2.6 Modified Condition / Decision Coverage

The MC/DC criterion enhances the Condition / Decision Coverage criterion by requiring that each condition be shown to independently affect the outcome of the decision. The independence requirement ensures that the effect of each condition is tested relative to the other conditions. In general, a minimum of n+1 test cases for a decision with n inputs. In example from the chapter 2.2 three test cases where A = false, B = false and A = true, B = false and A = false, B = true provide MC/DC.

3 The analysis method

In this chapter we describe our method to analyze the source codes written in Ada programming language. We used Antlr [3] parser generator with [4] grammar file to create the Abstract Syntax Tree (AST) of the source code. Our analysis is worked on this AST.

3.1 Counting Test Cases for Decision Coverage

The Decision Coverage requires that every decision must be evaluated to true and false at least once. So we need at least two test cases for every decision to satisfy these requirements. But one test case can tests several decision if they are not nested. Let us see the following example:

```
      if Condition_1 then
            true_statement_1;
      else
            false_statement_1;
      end if;

      ...

      if Condition_2 then
            true_statement_2;
      else
            false_statement_2;
      end if;
```

If the Condition_1 and the Condition_2 will be evaluated to true by the first test, and false by the second one, then these two test cases satisfy the requirements of the Decision Coverage. There are some extreme situations where the decisions cannot be fully covered. For example when both of Condition_1 and Condition_2 are identical to ***true***. These situations are rare and usually come from a coding error, what the static analyzers can alert, so we do not deal with.

Let us see how does it work with nested decisions:

```
      if Condition_1 then
            if Condition_2 then
                  true_statement_2;
            else
                  false_statement_2;
            end if;
      else
            false_statement_1
      end if;
```

We need two test cases for the Condition_2 to be evaluated both true and false. But in these test cases the Condition_1 must be evaluated to true, otherwise the false_statement_1 will be executed instead of the nested Condition_2. And at last we need a third test case where the Condition_1 is evaluated to false.

Let us see what is happens if there is a nested condition in both true part and false part of an ***if*** statement.

```
      if Condition_1 then
            if Condition_2 then
                  true_statement_2;
            else
                  false_statement_2;
            end if;
      else
            if Condition_3 then
                  true_statement_3;
            else
                  false_statement_3;
            end if;
      end if;
```

We need two test cases for both Condition_2 and Condition_3. Condition_1 must be evaluated to true in test cases belong to Condition_2, and it must be evaluated to false in test cases belong to Condition_3. But with these four test cases the requirements of Condition_1 are covered, so we do not need extra test case.

In summary we can say, $\boldsymbol{T} + \boldsymbol{F}$ test cases are needed to cover a decision. $\boldsymbol{T}$ means the number of test cases are needed for nested decision in true part or 1 if there is no nested decision there. $\boldsymbol{F}$ means the same in false part. A subprogram may contain more decisions in a same level.

We create classes of decisions and the identical decisions will be placed in the same classes. Then we consider the $\boldsymbol{max\ (T_j + F_j)}$ where $\boldsymbol{T_j, F_j}$ belong to the $\boldsymbol{j}$th class. We calculate $\boldsymbol{T_j}$ and $\boldsymbol{F_j}$ in the following way: $\boldsymbol{T_j}$ = max ($\boldsymbol{T_{j1}}$ .. $\boldsymbol{T_{jk}}$), $\boldsymbol{F_j}$ = max ($\boldsymbol{F_{j1}}$ .. $\boldsymbol{F_{jk}}$) where $\boldsymbol{k}$ is the number of the decisions in class $\boldsymbol{j}$. The $\boldsymbol{T_{jl}}$ and $\boldsymbol{F_{jl}}$ means the number of necessary test cases for true and false parts of the corresponding decisions. $\boldsymbol{(l = 1 .. k)}$

3.2 Counting Test Cases for Modified Condition / Decision Coverage

In this case we have two main steps. First we count how many test cases are needed to cover the decisions separately and then we check how do these decisions affect each others. If a decision contains more than 15 arguments, then we calculate with argument number plus one test cases, which is a lower bound estimation.

3.2.1 Analyzing decisions separately

We count how much test cases are needed to cover MC/DC for one decision in the following way:

- If the decision contains only one argument or the negation of that argument we need exactly two test cases. This case is same as Decision Coverage.
- If the decision contains two arguments with logical operator ***and***, ***and then***, ***or***, ***or else***, or ***xor***, we need exactly three test cases:

we need exactly three test cases:	TT, TF, FT	for ***and***
	TT, TF, one of FT, FF	for ***and then***
	FF, FT, TF	for ***or***
	FF, FT, one of TF, TT	for ***or else***
	three of TT, TF, FT, FF	for ***xor***

  where T means true and F means false.
- If the decision contains more arguments, then we use the following algorithm described in chapter 3.2.2.

3.2.2 The algorithm

This algorithm has five steps and based on algorithm described in [1].

1. Transform the AST belongs to the decision to contain information about the precedence of logical operators. (The AST, which generated by [3,4] is a bit different.)
2. Generate the all possible combination of values what the arguments can get. (2^n combinations, where n is a number of arguments.) These are the potential test cases.
3. Eliminate the masked test cases. For example let consider A and B, where B is false. In this occasion independently of A the whole logical expression is false. But A is not necessarily a logical variable, it can be another logical expression too and in this case the value of A does not affect the whole logical expression. It means this test case is masked for A and it can be

eliminated (for A). You can find more detailed description and examples in [1] about this step.

4. For every logical operator in decision: we collect the not masked test cases which satisfy one of its requirements described in previous chapter. So we get a set of test cases for every requirement of every logical operator. If one of these sets is empty the decision cannot be covered 100% by MC/DC. If it is happened we try to achieve as big coverage as possible.
5. We get the minimal covering set of these sets. We do it in a following way: let us suppose we have n arguments in a decision. The maximum number of test cases is m = 2^n and we numbered them 0..m-1. Of course almost all will be masked. Let us suppose all of the logical operators having two arguments (none of them are ***not***), so we have s = 3 x (n-1) sets. We calculate the minimal covering set by Integer Programming, where for every s_i set we have a disparity which is:

   $$\sum_{(k=0..m-1)} \chi_{(k\in s_i)} x_k > 1$$

   And our target function is:

   $$min\sum_{(k=0..m-1)} x_k$$

   With constraint: the value of every x_k can be only 0 or 1.

   When the result is calculated we get the minimal covering set. Every test case indexed with k is a member of the minimal covering set if x_k is 1.

   To do that calculation we used Lemon graph library [5] with glpk linear programming kit [6].

3.2.3 Analyzing decisions together

The calculation of nested decisions is similar to the Decision Coverage case but has some differences. It will be explained by an example below:

```
if A and B then
      if C or D then
            true_statement_2;
      else
            false_statement_2;
      end if;
else
      false_statement_1;
end if;
```

We need three test cases for inner decision where C = false, D = false; C = false, D = true; C = true, D = false. And we need three test cases for outer decision where A = true, B = true; A = true, B = false; A = false, B = true. But when A = true and B = true, the whole expression is true, so in that case we can test the inner decision simultaneously. So we need only five test cases to satisfy the requirements of MC/DC, which are in the following table:

	A	**B**	**C**	**D**
1.	true	true	false	false
2.	true	true	false	true
3.	true	true	true	false
4.	true	false	any	any
5.	false	true	any	any

If the outer decision is ***A or B*** instead of ***A and B*** then only four test cases are needed, because in that case the outer decision is evaluated to true twice so two test cases of inner decision can be run simultaneously.

If the outer decision is P or R or S then it is evaluated to true three times, so there is no additional test cases needed because we can run all the three inner test cases simultaneously. But we need four test cases to cover the outer decision.

Let us see another example, where there is a nested decision both of true and false part of outer decision:

```
if A and B then
     if C or D then
          true_statement_1;
     else
          false_statement_1;
     end if;
else
     if E and F then
          true_statement_2;
     else
          false_statement_2;
     end if;
end if;
```

We need 6 test cases to achieve 100% MC/DC coverage. These test cases can be seen in the following table.

	A	**B**	**C**	**D**	**E**	**F**
1.	true	true	false	false	any	any
2.	true	true	true	false	any	any

	A	**B**	**C**	**D**	**E**	**F**
3.	true	true	false	true	any	any
4.	true	false	any	any	true	true
5.	false	true	any	any	true	false
6.	false	any	any	any	false	true

In general we count the test cases needed for inner decision. If the outer decision is evaluated to true (or false if the inner decision is in else branch) less than the number of test cases required for inner decision then we increase the number test cases for outer decision.

If the if statement contains ***elsif*** branch then we transform the code as it can be seen in the following example:

```
if Condition_1 then                      if Condition_1 then
      statement_1;                             statement_1;
elsif Condition_2 then                   else
      statement_2;                             if Condition_2 then
else                    ----->                       statement_2;
      statement_3;                             else
end if;                                              statement_3;
                                               end if;
                                         end if;
```

With the transformed code we can work as we described above.

When there are decisions in same level and the variables in these decisions are independent, we need as many test cases as the maximum of test cases are needed to these decisions separately. If more decisions contain the same variable we need additional test cases. Let us see the following example:

```
if A and B then
      statement_1;
end if;
...
if A or C then
      statement_2;
end if;
```

In the first decision the variable ***A*** must be evaluated to ***true*** twice and to ***false*** once, and in the second decision it must be evaluated to ***true*** once and to ***false*** twice. For the whole subprogram ***A*** must be evaluated to ***true*** twice and to ***false*** twice, which means we need four test cases to satisfy the requirements of MC/DC. When the value of a variable changes between the two decisions, we consider it as a different variable. The value of a variable can be changed if it stands on the left side of an assignment or it stands on the ***out***, or ***in out*** position of a procedure as argument. In the following table you can see the values of the variables in the four test cases:

	A	**B**	**C**
1.	true	true	any
2.	true	false	false
3.	false	true	true
4.	false	any	false

In general way:

Decision 1 has n variable: $a_1, ..., a_n$
Decision 2 has m variable: $b_1, ..., b_m$
The first s variables are the common variables where $s \leq \min(n,m)$
Our algorithm works with k variables where $k = n+m-s$; There are $c_1, ..., c_k$
c_i.true means the number of test cases where the variable c_i evaluated to true.
c_i.false means the similar then previous one.

Let consider:

$$c_{i.}\text{true} = \max(a_i.\text{true}, b_i.\text{true}) \text{ if } i = 1 .. s$$
$$c_{i.}\text{true} = a_i.\text{true if } i = s+1..n$$
$$c_{i.}\text{true} = b_{i-n}.\text{true if } i = n+1..n+m-s$$

Number of test cases:

$$\max_{i=1..k} (c_i.\text{true} + c_i.\text{false})$$

4 Measurements and results

We analyzed six projects. In this chapter you can find statistics about these projects separately and summary of them.

4.1 Project: A

4.1.1 Statistic of the whole project

A means: the all files of the project,

B means: those files of the project, which contain at least one subprogram definition not only subprogram declarations.

	A	**B**
Number of files	249	110
Effective lines of code (without empty and comment lines)	81542	68736
Average Eloc / File	327	625
Number of subprograms	1678	
Average subprograms / File	15.3	

4.1.2 Subprograms and the argument number of decisions

In this chapter you can see how are the subprograms distributed by the argument number of their decisions.

Nr. of subprograms which has no decision	1134
Nr. of subprograms where all decisions have exactly one argument	389
Nr. of subprograms where all decisions have exactly one or two arguments	492

Nr. of subprograms where all decisions have exactly 1, 2 or 3 arguments	523
Nr. of subprogs where all decisions have at least one and at most five args.	541
Nr. of subprograms where all decisions have at least one arguments	544

4.1.3 Argument numbers and decisions

In this chapter you can see how are the decisions distributed by their argument numbers.

The argument numbers	**Number of decisions**
1	3664
2	274
3	63
4	21
5	9
6	2
8	1

4.1.4 DC - MC/DC in several aspects

In this chapter we examined how several aspects (McCabe metric, number of necessary MC/DC test cases, nesting, maximum argument number in decisions per subprogram and the summation of argument numbers in decisions per subprogram) do affect the difference between the necessary test cases for DC and MC/DC.

The whole project				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
1678	4449	4682	233	1.05

4.1.4.1 Grouping by McCabe metrics

Subprograms where McCabe metrics are between 0 and 10				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
1533	2361	2517	156	1.07

Subprograms where McCabe metrics are between 11 and 20				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
85	791	825	34	1.04

Subprograms where McCabe metrics are between 21 and 30				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
34	579	587	8	1.01

Subprograms where McCabe metrics are between 31 and 40				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
13	295	300	5	1.02

Subprograms where McCabe metrics are more than 40				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
13	423	453	30	1.07

4.1.4.2 Grouping by necessary MC/DC test cases

Subprograms where number of MC/DC test cases are between 1 and 2				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
1305	1476	1476	0	1.00

Subprograms where number of MC/DC test cases are between 3 and 4				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
165	484	533	49	1.10

Subprograms where number of MC/DC test cases are between 5 and 7				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
77	349	436	87	1.33

Subprograms where number of MC/DC test cases are between 8 and 10				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
42	333	372	39	1.12

Subprograms where number of MC/DC test cases are more than 10				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
89	1807	1865	58	1.03

4.1.4.3 Grouping by nesting values

Subprograms where the maximum nesting is between 0 and 1				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
1360	2498	2564	66	1.03

Subprograms where the maximum nesting is between 2 and 3				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
234	1058	1125	67	1.06

Subprograms where the maximum nesting is between 4 and 6				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
76	804	895	91	1.11

Subprograms where the maximum nesting is between 7 and 9				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
8	89	98	7	1.10

4.1.4.4 Grouping by maximum arguments number

Subprograms where there are no decisions				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
1134	1134	1134	0	1.00

Subprograms where the argument numbers in decisions are exactly 1				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
389	2308	2308	0	1.00

Subprograms where the maximum of argument numbers in decisions is between 2 and 3				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
134	873	1031	158	1.18

Subprograms where the maximum of argument numbers in decisions is between 4 and 5				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
18	126	184	58	1.46

Subprograms where the maximum of argument numbers in decisions is more than 5				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
3	8	25	17	3.125

4.1.4.5 Grouping by the summation of arguments in decisions

Subprograms where the summation of argument numbers in decisions is between 1 and 5				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
346	883	983	100	1.11

Subprograms where the summation of argument numbers in decisions is between 6 and 10				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
56	316	346	30	1.09

Subprograms where the summation of argument numbers in decisions is between 11 and 50				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
132	1766	1839	73	1.04

Subprograms where the summation of argument numbers in decisions is between 51 and 100				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
9	343	371	28	1.08

Subprograms where the summation of argument numbers in decisions is more than 100				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
1	7	9	2	1.29

4.1.5 Difference between the necessary of DC and MC/DC test cases

In this chapter you can see the number of subprograms where the difference of necessary test cases are 0, 1, 2 ... The *Diff* means the difference between the necessary DC and MC/DC test cases. The *Subpr* means how many subprograms are in the project where the difference between the two types of test cases is in the previous column. The *Min*, *Max* mean the minimum, maximum of MC/DC test cases per subprogram, and *Avg, Dev* mean the average and the standard deviation both of MC/DC and DC.

				DC		**MC/DC**	
Diff	**Subpr**	**Min**	**Max**	**Avg**	**Dev**	**Avg**	**Dev**
0	1561	1	125	2.13	3.66	2.13	3.66
1	65	3	27	3.85	4.11	4.85	4.11
2	25	4	9	4.00	1.30	6.00	1.30
3	10	5	14	5.20	3.03	8.20	3.03
4	11	6	57	10.00	14.24	14.00	14.24
5	3	7	13	4.67	2.49	9.67	2.49
6	1	19	19	13	0	19	0
7	1	9	9	2	0	9	0
16	1	72	72	56	0	72	0

4.1.6 DC - MC/DC Summary

In this chapter you can find how many test cases are needed for the project to cover DC and MC/DC.

In the following table, the

A means: the whole project,
B means: the whole project without those subprograms which do not contain decision,
C means: the whole project without those subprograms which contain decision with at least two arguments.

	DC	MC/DC	difference	ratio
A	4449	4682	233	1.05
B	3315	3548	233	1.07
C	1007	1240	233	1.23

4.2 Project: B

4.2.1 Statistic of the whole project

A means: the all files of the project,

B means: those files of the project, which contain at least one subprogram definition not only subprogram declarations.

	A	B
Number of files	281	145
Effective lines of code (without empty and comment lines)	79012	63783
Average Eloc / File	281	439
Number of subprograms	1286	
Average Subprograms / File	8.87	

4.2.2 Subprograms and the argument number of decisions

In this chapter you can see how are the subprograms distributed by the argument number of their decisions.

Nr. of subprograms which has no decision	586
Nr. of subprograms where all decisions have exactly one argument	493

Nr. of subprograms where all decisions have exactly one or two arguments	623
Nr. of subprograms where all decisions have exactly 1, 2 or 3 arguments	659
Nr. of subprogs. where all decisions have at least one and at most five args.	684
Nr. of subprograms where all decisions have at least one arguments	700

4.2.3 Argument numbers and decisions

In this chapter you can see how are the decisions distributed by their argument numbers.

The argument numbers	**Number of decisions**
1	3411
2	253
3	63
4	21
5	10
6	7
7	7
9	1
12	1
15	3
16	1

4.2.4 DC - MC/DC in several aspects

In this chapter we examined how several aspects (McCabe metric, number of necessary MC/DC test cases, nesting, maximum argument number in decisions per subprogram and the summation of argument numbers in decisions per subprogram) do affect the difference between the necessary test cases for DC and MC/DC.

The whole project				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
1286	3694	4031	337	1.09

4.2.4.1 Grouping by McCabe metrics

Subprograms where McCabe metrics are between 0 and 10				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
1212	2854	3113	259	1.09

Subprograms where McCabe metrics are between 11 and 20				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
56	475	537	62	1.13

Subprograms where McCabe metrics are between 21 and 30				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
13	215	220	5	1.023

Subprograms where McCabe metrics are between 31 and 40				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
4	113	117	4	1.03

Subprograms where McCabe metrics are more than 40				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
1	37	44	7	1.19

4.2.4.2 Grouping by necessary MC/DC test cases

Subprograms where number of MC/DC test cases are between 1 and 2				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
807	1026	1026	0	1.00

Subprograms where number of MC/DC test cases are between 3 and 4				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
221	686	739	53	1.08

Subprograms where number of MC/DC test cases are between 5 and 7				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
145	750	830	80	1.11

Subprograms where number of MC/DC test cases are between 8 and 10				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
62	467	543	76	1.16

Subprograms where number of MC/DC test cases are more than 10				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
51	765	893	128	1.17

4.2.4.3 Grouping by nesting values

Subprograms where the maximum nesting is between 0 and 1				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
912	1696	1831	135	1.08

Subprograms where the maximum nesting is between 2 and 3				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
298	1436	1589	153	1.11

Subprograms where the maximum nesting is between 4 and 6				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
74	555	604	49	1.09

Subprograms where the maximum nesting is above than 7				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
2	7	7	0	1.00

4.2.4.4 Grouping by maximum arguments number

Subprograms where there are no decisions				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
586	586	586	0	1.00

Subprograms where the argument numbers in decisions are exactly 1				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
493	1957	1957	0	1.00

Subprograms where the maximum of argument numbers in decisions is between 2 and 3				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
166	900	1081	181	1.20

Subprograms where the maximum of argument numbers in decisions is between 4 and 5				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
25	156	236	80	1.51

Subprograms where the maximum of argument numbers in decisions is between 6 and 10				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
11	65	110	45	1.69

Subprograms where the maximum of argument numbers in decisions is more than 10				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
5	30	61	31	2.03

4.2.4.5 Grouping by the summation of arguments in decisions

Subprograms where the summation of argument numbers in decisions is between 1 and 5				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
406	1094	1167	73	1.07

Subprograms where the summation of argument numbers in decisions is between 6 and 10				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
194	933	1040	107	1.11

Subprograms where the summation of argument numbers in decisions is between 11 and 50				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
98	1022	1172	150	1.15

Subprograms where the summation of argument numbers in decisions is more than 50				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
2	59	66	7	1.12

4.2.5 Difference between the necessary of DC and MC/DC test cases

In this chapter you can see the number of subprograms where the difference of necessary test cases are 0, 1, 2 ... The *Diff* means the difference between the necessary DC and MC/DC test cases. The *Subpr* means how many subprograms are in the project where the difference between the two types of test cases is in the previous column. The *Min, Max* mean the minimum, maximum of MC/DC test cases per subprogram, and *Avg, Dev* mean the average and the standard deviation both of MC/DC and DC.

				DC		**MC/DC**	
Diff	**Subpr**	**Min**	**Max**	**Avg**	**Dev**	**Avg**	**Dev**
0	1135	1	39	2.55	3.09	2.55	3.09
1	81	3	29	4.74	4.38	5.74	4.38
2	39	4	19	5.41	3.37	7.41	3.37
3	9	5	27	5.44	6.73	8.44	6.73
4	7	8	20	6.29	4.03	10.29	4.03

				DC		MC/DC	
Diff	**Subpr**	**Min**	**Max**	**Avg**	**Dev**	**Avg**	**Dev**
5	1	12	12	7	0	12	0
6	5	8	13	4.80	2.32	10.80	2.32
7	2	10	44	20	17	27	17
8	2	12	15	5.5	1.5	13.5	1.5
9	1	21	21	12	0	21	0
11	1	13	13	2	0	13	0
12	2	20	20	8	0	20	0
14	1	16	16	2	0	16	0

4.2.6 DC - MC/DC

In this chapter you can find how many test cases are needed for the project to cover DC and MC/DC.

In the following table, the

A means: the whole project,
B means: the whole project without those subprograms which do not contain decision,
C means: the whole project without those subprograms which contain decision with at least two arguments.

	DC	MC/DC	difference	ratio
A	3694	4031	337	1.09
B	3108	3445	337	1.11
C	1151	1488	337	1.29

4.3 Project: C

4.3.1 Statistic of the whole project

A means: the all files of the project,

B means: those files of the project, which contain at least one subprogram definition not only subprogram declarations.

	A	**B**
Number of files	2182	1039
Effective lines of code (without empty and comment lines)	283030	222924
Average Eloc / File	129	214
Number of subprograms	5577	
Average Subprograms / File	5.37	

4.3.2 Subprograms and the argument number of decisions

In this chapter you can see how are the subprograms distributed by the argument number of their decisions.

Nr. of subprograms which has no decision	3299
Nr. of subprograms where all decisions have exactly one argument	1817
Nr. of subprograms where all decisions have exactly one or two arguments	2177
Nr. of subprograms where all decisions have exactly 1, 2 or 3 arguments	2233
Nr. of subprogs. where all decisions have at least one and at most five args.	2266
Nr. of subprograms where all decisions have at least one arguments	2278

4.3.3 Argument numbers and decisions

In this chapter you can see how are the decisions distributed by their argument numbers.

The argument numbers	Number of decisions
1	11088
2	666
3	104
4	46
5	14
6	3
8	1
9	2
10	1
11	1
12	2
13	2
15	1
22	1

4.3.4 DC - MC/DC in several aspects

In this chapter we examined how several aspects (McCabe metric, number of necessary MC/DC test cases, nesting, maximum argument number in decisions per subprogram and the summation of argument numbers in decisions per subprogram) do affect the difference between the necessary test cases for DC and MC/DC.

The whole project				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
5577	12611	13292	681	1.05

4.3.4.1 Grouping by McCabe metrics

<table>
<tr><th colspan="5">Subprograms where McCabe metrics are between 0 and 10</th></tr>
<tr><td>Nr. of Subpr.</td><td>DC</td><td>MCDC</td><td>Difference</td><td>Ratio</td></tr>
<tr><td>5498</td><td>9220</td><td>9730</td><td>510</td><td>1.06</td></tr>
</table>

<table>
<tr><th colspan="5">Subprograms where McCabe metrics are between 11 and 20</th></tr>
<tr><td>Nr. of Subpr.</td><td>DC</td><td>MCDC</td><td>Difference</td><td>Ratio</td></tr>
<tr><td>177</td><td>1542</td><td>1634</td><td>92</td><td>1.06</td></tr>
</table>

<table>
<tr><th colspan="5">Subprograms where McCabe metrics are between 21 and 30</th></tr>
<tr><td>Nr. of Subpr.</td><td>DC</td><td>MCDC</td><td>Difference</td><td>Ratio</td></tr>
<tr><td>55</td><td>729</td><td>749</td><td>20</td><td>1.03</td></tr>
</table>

<table>
<tr><th colspan="5">Subprograms where McCabe metrics are between 31 and 40</th></tr>
<tr><td>Nr. of Subpr.</td><td>DC</td><td>MCDC</td><td>Difference</td><td>Ratio</td></tr>
<tr><td>27</td><td>394</td><td>409</td><td>15</td><td>1.04</td></tr>
</table>

<table>
<tr><th colspan="5">Subprograms where McCabe metrics are more than 40</th></tr>
<tr><td>Nr. of Subpr.</td><td>DC</td><td>MCDC</td><td>Difference</td><td>Ratio</td></tr>
<tr><td>20</td><td>726</td><td>770</td><td>44</td><td>1.06</td></tr>
</table>

4.3.4.2 Grouping by necessary MC/DC test cases

<table>
<tr><th colspan="5">Subprograms where number of MC/DC test cases are between 1 and 2</th></tr>
<tr><td>Nr. of Subpr.</td><td>DC</td><td>MCDC</td><td>Difference</td><td>Ratio</td></tr>
<tr><td>4509</td><td>5420</td><td>5420</td><td>0</td><td>1.00</td></tr>
</table>

Subprograms where number of MC/DC test cases are between 3 and 4				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
753	2308	2575	267	1.12

Subprograms where number of MC/DC test cases are between 5 and 7				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
298	1521	1671	150	1.10

Subprograms where number of MC/DC test cases are between 8 and 10				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
73	582	637	55	1.09

Subprograms where number of MC/DC test cases are more than 10				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
144	2780	2989	209	1.08

4.3.4.3 Grouping by nesting values

Subprograms where the maximum nesting is between 0 and 1				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
4331	6304	6552	248	1.04

Subprograms where the maximum nesting is between 2 and 3				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
948	3613	3840	227	1.06

Subprograms where the maximum nesting is between 4 and 6				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
262	2027	2189	162	1.08

Subprograms where the maximum nesting is above than 7				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
36	667	711	44	1.07

4.3.4.4 Grouping by maximum arguments number

Subprograms where there are no decisions				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
3299	3299	3299	0	1.00

Subprograms where the argument numbers in decisions are exactly 1				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
1817	6475	6475	0	1.00

Subprograms where the maximum of argument numbers in decisions is between 2 and 3				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
416	2396	2866	470	1.19

Subprograms where the maximum of argument numbers in decisions is between 4 and 5				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
33	362	484	122	1.34

Subprograms where the maximum of argument numbers in decisions is between 6 and 10				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
6	26	66	40	2.54

4.3.4.5 Grouping by the summation of arguments in decisions

Subprograms where the summation of argument numbers in decisions is between 1 and 5				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
1697	4361	4670	309	1.07

Subprograms where the summation of argument numbers in decisions is between 6 and 10				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
281	1357	1475	118	1.09

Subprograms where the summation of argument numbers in decisions is between 11 and 50				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
283	3004	3206	202	1.07

Subprograms where the summation of argument numbers in decisions is between 51 and 100				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
13	315	345	30	1.10

Subprograms where the summation of argument numbers in decisions is more than 100				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
4	275	297	22	1.08

4.3.5 Difference between the necessary of DC and MC/DC test cases

In this chapter you can see the number of subprograms where the difference of necessary test cases are 0, 1, 2 ... The *Diff* means the difference between the necessary DC and MC/DC test cases. The *Subpr* means how many subprograms are in the project where the difference between the two types of test cases is in the previous column. The *Min*, *Max* mean the minimum, maximum of MC/DC test cases per subprogram, and *Avg, Dev* mean the average and the standard deviation both of MC/DC and DC.

				DC		**MC/DC**	
Diff	**Subpr**	**Min**	**Max**	**Avg**	**Dev**	**Avg**	**Dev**
0	5162	1	101	2.08	3.36	2.08	3.36
1	314	3	88	3.63	5.59	4.63	5.59
2	60	4	37	5.33	6.14	7.33	6.14
3	17	5	35	6.35	7.05	9.35	7.05
4	5	7	28	13	8.60	17	8.60
6	4	9	13	4.75	1.79	10.75	1.79
7	4	10	24	7	5.83	14	5.83
8	4	11	20	8.25	3.90	16.25	3.90
9	1	16	16	7	0	16	0
10	1	25	25	15	0	25	0
11	1	13	13	2	0	13	0
12	1	14	14	2	0	14	0
13	1	15	15	2	0	15	0
16	1	51	51	35	0	51	0
21	1	95	95	74	0	95	0

4.3.6 DC - MC/DC

In this chapter you can find how many test cases are needed for the project to cover DC and MC/DC.

In the following table, the

A means: the whole project,
B means: the whole project without those subprograms which do not contain decision,
C means: the whole project without those subprograms which contain decision with at least two arguments.

	DC	MC/DC	difference	ratio
A	12611	13292	681	1.05
B	9312	9993	681	1.07
C	2837	3518	681	1.24

4.4 Project: D

4.4.1 Statistic of the whole project

A means: the all files of the project,
B means: those files of the project, which contain at least one subprogram definition not only subprogram declarations.

	A	B
Number of files	722	410
Effective lines of code (without empty and comment lines)	108395	89418
Average Eloc / File	150	218
Number of subprograms	1847	
Average Subprograms / File	4.5	

4.4.2 Subprograms and the argument number of decisions

In this chapter you can see how are the subprograms distributed by the argument number of their decisions.

Nr. of subprograms which has no decision	996
Nr. of subprograms where all decisions have exactly one argument	651
Nr. of subprograms where all decisions have exactly one or two arguments	809
Nr. of subprograms where all decisions have exactly 1, 2 or 3 arguments	835
Nr. of subprogs where all decisions have at least one and at most five args.	845
Nr. of subprograms where all decisions have at least one arguments	851

4.4.3 Argument numbers and decisions

In this chapter you can see how are the decisions distributed by their argument numbers.

The argument numbers	Number of decisions
1	3693
2	281
3	37
4	9
5	5
6	4
7	1
8	1

4.4.4 DC - MC/DC in several aspects

In this chapter we examined how several aspects (McCabe metric, number of necessary MC/DC test cases, nesting, maximum argument number in decisions per subprogram and the summation of argument numbers in decisions per subprogram) do affect the difference between the necessary test cases for DC and MC/DC.

The whole project				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
1847	4678	4908	230	1.05

4.4.4.1 Grouping by McCabe metrics

Subprograms where McCabe metrics are between 0 and 10				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
1746	3314	3505	191	1.06

Subprograms where McCabe metrics are between 11 and 20				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
72	705	729	21	1.03

Subprograms where McCabe metrics are between 21 and 30				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
12	166	178	12	1.07

Subprograms where McCabe metrics are between 31 and 40				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
11	207	210	3	1.01

Subprograms where McCabe metrics are more than 40

Nr. of Subpr.	DC	MCDC	Difference	Ratio
6	286	286	0	1.00

4.4.4.2 Grouping by necessary MC/DC test cases

Subprograms where number of MC/DC test cases are between 1 and 2

Nr. of Subpr.	DC	MCDC	Difference	Ratio
1359	1781	1781	0	1.00

Subprograms where number of MC/DC test cases are between 3 and 4

Nr. of Subpr.	DC	MCDC	Difference	Ratio
269	801	912	111	1.14

Subprograms where number of MC/DC test cases are between 5 and 7

Nr. of Subpr.	DC	MCDC	Difference	Ratio
122	639	695	56	1.09

Subprograms where number of MC/DC test cases are between 8 and 10

Nr. of Subpr.	DC	MCDC	Difference	Ratio
36	292	314	22	1.08

Subprograms where number of MC/DC test cases are more than 10

Nr. of Subpr.	DC	MCDC	Difference	Ratio
61	1165	1206	41	1.04

4.4.4.3 Grouping by nesting values

Subprograms where the maximum nesting is between 0 and 1				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
1341	2055	2133	78	1.04

Subprograms where the maximum nesting is between 2 and 3				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
366	1358	1449	91	1.06

Subprograms where the maximum nesting is between 4 and 6				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
121	910	971	61	1.07

Subprograms where the maximum nesting is above than 7				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
19	338	355	17	1.05

4.4.4.4 Grouping by maximum arguments number

Subprograms where there are no decisions				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
996	996	996	0	1.0

Subprograms where the argument numbers in decisions are exactly 1				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
656	2466	2466	0	1.00

<table>
<tr><th colspan="5">Subprograms where the maximum of argument numbers in decisions is between 2 and 3</th></tr>
<tr><td>Nr. of Subpr.</td><td>DC</td><td>MCDC</td><td>Difference</td><td>Ratio</td></tr>
<tr><td>179</td><td>1072</td><td>1249</td><td>177</td><td>1.17</td></tr>
</table>

<table>
<tr><th colspan="5">Subprograms where the maximum of argument numbers in decisions is between 4 and 5</th></tr>
<tr><td>Nr. of Subpr.</td><td>DC</td><td>MCDC</td><td>Difference</td><td>Ratio</td></tr>
<tr><td>10</td><td>117</td><td>135</td><td>18</td><td>1.15</td></tr>
</table>

<table>
<tr><th colspan="5">Subprograms where the maximum of argument numbers in decisions is between 6 and 10</th></tr>
<tr><td>Nr. of Subpr.</td><td>DC</td><td>MCDC</td><td>Difference</td><td>Ratio</td></tr>
<tr><td>5</td><td>18</td><td>46</td><td>28</td><td>2.56</td></tr>
</table>

<table>
<tr><th colspan="5">Subprograms where the maximum of argument numbers in decisions is more than 10</th></tr>
<tr><td>Nr. of Subpr.</td><td>DC</td><td>MCDC</td><td>Difference</td><td>Ratio</td></tr>
<tr><td>1</td><td>9</td><td>16</td><td>7</td><td>1.78</td></tr>
</table>

4.4.4.5 Grouping by the summation of arguments in decisions

<table>
<tr><th colspan="5">Subprograms where the summation of argument numbers in decisions is between 1 and 5</th></tr>
<tr><td>Nr. of Subpr.</td><td>DC</td><td>MCDC</td><td>Difference</td><td>Ratio</td></tr>
<tr><td>683</td><td>1726</td><td>1854</td><td>128</td><td>1.07</td></tr>
</table>

Subprograms where the summation of argument numbers in decisions is between 6 and 10				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
83	547	585	38	1.07

Subprograms where the summation of argument numbers in decisions is between 11 and 50				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
80	1179	1243	64	1.05

Subprograms where the summation of argument numbers in decisions is between 51 and 100				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
4	129	129	0	1.00

Subprograms where the summation of argument numbers in decisions is more than 100				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
1	101	101	0	1.00

4.4.5 Difference between the necessary of DC and MC/DC test cases

In this chapter you can see the number of subprograms where the difference of necessary test cases are 0, 1, 2 ... The *Diff* means the difference between the necessary DC and MC/DC test cases. The *Subpr* means how many subprograms are in the project where the difference between the two types of test cases is in the previous column. The *Min, Max* mean the minimum, maximum of MC/DC test cases per subprogram, and *Avg, Dev* mean the average and the standard deviation both of MC/DC and DC.

				DC		MC/DC	
Diff	**Subpr**	**Min**	**Max**	**Avg**	**Dev**	**Avg**	**Dev**
0	1679	1	101	2.33	4.22	2.33	4.22
1	134	3	26	3.71	3.25	4.71	3.25
2	24	4	29	6.29	6.94	8.29	6.94
3	3	5	15	5.33	4.71	8.33	4.71
4	2	7	12	5.5	2.5	9.5	2.5
5	1	7	7	2	0	7	0
6	2	9	13	5	1	11	1
7	2	10	16	6	3	13	3

4.4.6 DC - MC/DC

In this chapter you can find how many test cases are needed for the project to cover DC and MC/DC.

In the following table, the

A means: the whole project,
B means: the whole project without those subprograms which do not contain decision,
C means: the whole project without those subprograms which contain decision with at least two arguments.

	DC	MC/DC	difference	ratio
A	4678	4908	230	1.05
B	3682	3912	230	1.06
C	1216	1446	230	1.19

4.5 Project: E

4.5.1 Statistic of the whole project

A means: the all files of the project,

B means: those files of the project, which contain at least one subprogram definition not only subprogram declarations.

	A	**B**
Number of files	1105	704
Effective lines of code (without empty and comment lines)	249307	183258
Average Eloc / File	226	260
Number of subprograms	6243	
Average Subprograms / File	8.8	

4.5.2 Subprograms and the argument number of decisions

In this chapter you can see how are the subprograms distributed by the argument number of their decisions.

Nr. of subprograms which has no decision	3469
Nr. of subprograms where all decisions have exactly one argument	2324
Nr. of subprograms where all decisions have exactly one or two arguments	2557
Nr. of subprograms where all decisions have exactly 1, 2 or 3 arguments	2631
Nr. of subprogs. where all decisions have at least one and at most five args.	2700
Nr. of subprograms where all decisions have at least one arguments	2985

4.5.3 Argument numbers and decisions

In this chapter you can see how are the decisions distributed by their argument numbers.

The argument numbers	Number of decisions
1	15731
2	621
3	177
4	87
5	46
6	30
7	15
8	17
9	11
10	13
11	9
12	5
13	5
14	4
18	1
34	1

4.5.4 DC - MC/DC in several aspects

In this chapter we examined how several aspects (McCabe metric, number of necessary MC/DC test cases, nesting, maximum argument number in decisions per subprogram and the summation of argument numbers in decisions per subprogram) do affect the difference between the necessary test cases for DC and MC/DC.

The whole project				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
6243	14391	15685	1294	1.09

4.5.4.1 Grouping by McCabe metrics

Subprograms where McCabe metrics are between 0 and 10				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
5792	9801	10523	722	1.07

Subprograms where McCabe metrics are between 11 and 20				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
244	1499	1678	179	1.12

Subprograms where McCabe metrics are between 21 and 30				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
106	820	907	87	1.11

Subprograms where McCabe metrics are between 31 and 40				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
31	256	302	46	1.18

Subprograms where McCabe metrics are more than 40				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
70	2015	2275	260	1.13

4.5.4.2 Grouping by necessary MC/DC test cases

Subprograms where number of MC/DC test cases are between 1 and 2				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
4917	6363	6363	0	1.00

Subprograms where number of MC/DC test cases are between 3 and 4				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
779	2398	2557	159	1.07

Subprograms where number of MC/DC test cases are between 5 and 7				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
275	1305	1555	259	1.19

Subprograms where number of MC/DC test cases are between 8 and 10				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
103	732	913	181	1.25

Subprograms where number of MC/DC test cases are more than 10				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
169	3593	4297	704	1.20

4.5.4.3 Grouping by nesting values

Subprograms where the maximum nesting is between 0 and 1				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
4885	8231	8748	517	1.06

Subprograms where the maximum nesting is between 2 and 3				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
1058	3782	4173	391	1.10

Subprograms where the maximum nesting is between 4 and 6				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
262	1896	2165	269	1.14

Subprograms where the maximum nesting is above than 7				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
38	482	599	117	1.24

4.5.4.4 Grouping by maximum arguments number

Subprograms where there are no decisions				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
3469	3469	3469	0	1.00

Subprograms where the argument numbers in decisions are exactly 1				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
2324	7840	7840	0	1.00

Subprograms where the maximum of argument numbers in decisions is between 2 and 3				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
307	2106	2539	433	1.21

Subprograms where the maximum of argument numbers in decisions is between 4 and 5				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
69	401	642	241	1.60

Subprograms where the maximum of argument numbers in decisions is between 6 and 10				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
52	359	718	359	2.00

Subprograms where the maximum of argument numbers in decisions is more than 10				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
22	216	477	261	2.21

4.5.4.5 Grouping by the summation of arguments in decisions

Subprograms where the summation of argument numbers in decisions is between 1 and 5				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
1984	4871	5161	290	1.06

Subprograms where the summation of argument numbers in decisions is between 6 and 10				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
381	1517	1705	188	1.12

Subprograms where the summation of argument numbers in decisions is between 11 and 50				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
365	2989	3496	507	1.17

Subprograms where the summation of argument numbers in decisions is between 51 and 100				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
30	743	950	207	1.28

Subprograms where the summation of argument numbers in decisions is more than 100				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
14	802	904	102	1.13

4.5.5 Difference between the necessary of DC and MC/DC test cases

In this chapter you can see the number of subprograms where the difference of necessary test cases are 0, 1, 2 ... The *Diff* means the difference between the necessary DC and MC/DC test cases. The *Subpr* means how many subprograms are in the project where the difference between the two types of test cases is in the previous column. The *Min*, *Max* mean the minimum, maximum of MC/DC test cases per subprogram, and *Avg, Dev* mean the average and the standard deviation both of MC/DC and DC.

				DC		**MC/DC**	
Diff	**Subpr**	**Min**	**Max**	**Avg**	**Dev**	**Avg**	**Dev**
0	5856	1	175	2.03	4.58	2.03	4.58
1	169	3	47	4.07	4.53	5.07	4.53
2	74	4	41	4.72	4.98	6.72	4.98
3	43	5	26	5.70	4.42	8.70	4.42

				DC		MC/DC	
Diff	**Subpr**	**Min**	**Max**	**Avg**	**Dev**	**Avg**	**Dev**
4	30	6	22	5.77	4.46	9.77	4.46
5	14	7	17	4.64	3.22	9.64	3.22
6	8	8	17	5.88	3.14	11.88	3.14
7	8	9	42	8.13	10.36	15.13	10.36
8	9	10	22	4.33	3.62	12.33	3.62
9	11	11	49	7	11.53	18	11.53
10	2	12	14	3	1	13	1
11	4	13	28	8	6.36	19	6.36
12	1	14	14	2	0	14	0
13	2	15	16	2.50	0.50	15.50	0.50
15	1	105	105	90	0	105	0
16	1	44	44	28	0	44	0
18	1	20	20	2	0	20	0
19	1	33	33	14	0	33	0
21	2	61	247	133	93	154	93
22	1	145	145	123	0	145	0
25	1	61	61	36	0	61	0
27	1	68	68	41	0	68	0
29	1	39	39	10	0	39	0
31	1	74	74	43	0	74	0
36	1	77	77	41	0	77	0

4.5.6 DC - MC/DC

In this chapter you can find how many test cases are needed for the project to cover DC and MC/DC.

In the following table, the

A means: the whole project,
B means: the whole project without those subprograms which do not contain decision,
C means: the whole project without those subprograms which contain decision with at least two arguments.

	DC	**MC/DC**	**difference**	**ratio**
A	14391	15685	1294	1.08
B	10922	12216	1294	1.12
C	3082	4376	1294	1.42

4.6 Project: F

4.6.1 Statistic of the whole project

A means: the all files of the project,

B means: those files of the project, which contain at least one subprogram definition not only subprogram declarations.

	A	**B**
Number of files	1938	1040
Effective lines of code (without empty and comment lines)	335926	235512
Average Eloc / File	173	226
Number of subprograms	6176	
Average Subprograms / File	5.9	

4.6.2 Subprograms and the argument number of decisions

In this chapter you can see how are the subprograms distributed by the argument number of their decisions.

Nr. of subprograms which has no decision	3343
Nr. of subprograms where all decisions have exactly one argument	2130
Nr. of subprograms where all decisions have exactly one or two arguments	2599
Nr. of subprograms where all decisions have exactly 1, 2 or 3 arguments	2701
Nr. of subprogs. where all decisions have at least one and at most five args.	2775
Nr. of subprograms where all decisions have at least one arguments	2833

4.6.3 Argument numbers and decisions

In this chapter you can see how are the decisions distributed by their argument numbers.

The argument numbers	**Number of decisions**
1	12715
2	1251
3	171
4	110
5	25
6	46
7	9

The argument numbers	Number of decisions
8	17
9	6
10	4
11	4
12	4
13	2
23	4

4.6.4 DC - MC/DC and McCabe metric

In this chapter we examined how several aspects (McCabe metric, number of necessary MC/DC test cases, nesting, maximum argument number in decisions per subprogram and the summation of argument numbers in decisions per subprogram) do affect the difference between the necessary test cases for DC and MC/DC.

The whole project				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
6176	15172	16426	1254	1.08

4.6.4.1 Grouping by McCabe metrics

Subprograms where McCabe metrics are between 0 and 10				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
5690	9972	10636	664	1.07

Subprograms where McCabe metrics are between 11 and 20				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
289	1919	2138	219	1.11

Subprograms where McCabe metrics are between 21 and 30				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
74	701	791	90	1.13

Subprograms where McCabe metrics are between 31 and 40				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
55	638	778	138	1.22

Subprograms where McCabe metrics are more than 40				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
68	1942	2083	141	1.07

4.6.4.2 Grouping by necessary MC/DC test cases

Subprograms where number of MC/DC test cases are between 1 and 2				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
4637	5927	5927	0	1.00

Subprograms where number of MC/DC test cases are between 3 and 4				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
776	2431	2673	241	1.10

Subprograms where number of MC/DC test cases are between 5 and 7				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
439	2192	2534	342	1.16

<table>
<tr><th colspan="5">Subprograms where number of MC/DC test cases are between 8 and 10</th></tr>
<tr><td>Nr. of Subpr.</td><td>DC</td><td>MCDC</td><td>Difference</td><td>Ratio</td></tr>
<tr><td>148</td><td>1132</td><td>1297</td><td>165</td><td>1.15</td></tr>
</table>

<table>
<tr><th colspan="5">Subprograms where number of MC/DC test cases are more than 10</th></tr>
<tr><td>Nr. of Subpr.</td><td>DC</td><td>MCDC</td><td>Difference</td><td>Ratio</td></tr>
<tr><td>176</td><td>3490</td><td>3995</td><td>505</td><td>1.15</td></tr>
</table>

4.6.4.3 Grouping by nesting values

<table>
<tr><th colspan="5">Subprograms where the maximum nesting is between 0 and 1</th></tr>
<tr><td>Nr. of Subpr.</td><td>DC</td><td>MCDC</td><td>Difference</td><td>Ratio</td></tr>
<tr><td>4430</td><td>7507</td><td>7883</td><td>376</td><td>1.05</td></tr>
</table>

<table>
<tr><th colspan="5">Subprograms where the maximum nesting is between 2 and 3</th></tr>
<tr><td>Nr. of Subpr.</td><td>DC</td><td>MCDC</td><td>Difference</td><td>Ratio</td></tr>
<tr><td>1145</td><td>3903</td><td>4223</td><td>320</td><td>1.08</td></tr>
</table>

<table>
<tr><th colspan="5">Subprograms where the maximum nesting is between 4 and 6</th></tr>
<tr><td>Nr. of Subpr.</td><td>DC</td><td>MCDC</td><td>Difference</td><td>Ratio</td></tr>
<tr><td>493</td><td>2960</td><td>3387</td><td>427</td><td>1.14</td></tr>
</table>

<table>
<tr><th colspan="5">Subprograms where the maximum nesting is above than 6</th></tr>
<tr><td>Nr. of Subpr.</td><td>DC</td><td>MCDC</td><td>Difference</td><td>Ratio</td></tr>
<tr><td>108</td><td>802</td><td>933</td><td>131</td><td>1.16</td></tr>
</table>

4.6.4.4 Grouping by maximum arguments number

Subprograms where there are no decisions				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
3343	3343	3343	0	1.00

Subprograms where the argument numbers in decisions are exactly 1				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
2130	7304	7304	29	1.00

Subprograms where the maximum of argument numbers in decisions is between 2 and 3				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
571	3504	4223	719	1.21

Subprograms where the maximum of argument numbers in decisions is between 4 and 5				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
74	471	673	202	1.43

Subprograms where the maximum of argument numbers in decisions is between 6 and 10				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
44	399	630	231	1.58

Subprograms where the maximum of argument numbers in decisions is more than 10				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
14	151	253	102	1.68

4.6.4.5 Grouping by the summation of arguments in decisions

Subprograms where the summation of argument numbers in decisions is between 1 and 5				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
2020	4962	5268	306	1.06

Subprograms where the summation of argument numbers in decisions is between 6 and 10				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
433	2204	2469	265	1.12

Subprograms where the summation of argument numbers in decisions is between 11 and 50				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
354	3473	4077	604	1.17

Subprograms where the summation of argument numbers in decisions is between 51 and 100				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
18	628	684	56	1.09

Subprograms where the summation of argument numbers in decisions is more than 100				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
8	562	585	23	1.04

4.6.5 Difference between the necessary of DC and MC/DC test cases

In this chapter you can see the number of subprograms where the difference of necessary test cases are 0, 1, 2 ... The *Diff* means the difference between the necessary DC and MC/DC test cases. The *Subpr* means how many subprograms are in the project where the difference between the two types of test cases is in the previous column. The *Min, Max* mean the minimum, maximum of MC/DC test cases per subprogram, and *Avg, Dev* mean the average and the standard deviation both of MC/DC and DC.

				DC		**MC/DC**	
Diff	**Subpr**	**Min**	**Max**	**Avg**	**Dev**	**Avg**	**Dev**
0	5616	1	125	2.13	3.66	2.13	3.66
1	320	2	54	4.60	6.72	5.60	6.72
2	120	4	40	4.99	4.30	6.99	4.30
3	43	5	21	5.74	4.67	8.74	4.67
4	21	6	95	13.52	25.31	17.52	25.31
5	15	7	19	6.73	3.82	11.73	3.82
6	15	8	27	11.40	7.20	17.40	7.20
7	7	11	40	17.29	11.26	24.29	11.26
8	4	13	14	5.50	0.50	13.50	0.50
11	2	19	19	7	0	19	0
12	3	14	34	15.33	9.43	27.33	9.43
14	4	22	31	12.50	4.50	26.50	4.50
15	1	60	60	35	0	60	0
18	1	40	40	22	0	40	0
19	2	37	37	18	0	37	0
23	2	25	25	2	0	25	0

4.6.6 DC - MC/DC

In this chapter you can find how many test cases are needed for the project to cover DC and MC/DC.

In the following table, the

A means: the whole project,
B means: the whole project without those subprograms which do not contain decision,
C means: the whole project without those subprograms which contain decision with at least two arguments.

	DC	MC/DC	difference	ratio
A	15172	16426	1254	1.08
B	11829	13083	1254	1.11
C	4554	5779	1254	1.27

4.7 The six projects together

4.7.1 Statistic of the six projects

A means: the all files of the projects,

B means: those files of the projects, which contains at least one subprogram definition not only subprogram declarations.

	A	B
Number of files	6477	3448
Effective lines of code (without empty and comment lines)	1137212	863631
Average Eloc / File	176	250
Number of subprograms	22842	
Average Subprograms / File	6.6	

4.7.2 Subprograms and the argument number of decisions

In this chapter you can see how are the subprograms distributed by the argument number of their decisions.

Nr. of subprograms which has no decision	12827
Nr. of subprograms where all decisions have exactly one argument	7839
Nr. of subprograms where all decisions have exactly one or two arguments	9302
Nr. of subprograms where all decisions have exactly 1, 2 or 3 arguments	9617
Nr. of subprogs. where all decisions have at least one and at most five args.	9846
Nr. of subprograms where all decisions have at least one arguments	10015

4.7.3 Argument numbers and decisions

In this chapter you can see how are the decisions distributed by their argument numbers.

The argument numbers	Number of decisions
1	50302
2	3346
3	615
4	284
5	109
6	92
7	32

The argument numbers	Number of decisions
8	37
9	20
10	18
11	14
12	13
13	9
14	4
15	4
16	1
18	1
22	1
23	4
34	1

4.7.4 DC - MC/DC in several aspects

In this chapter we examined how several aspects (McCabe metric, number of necessary MC/DC test cases, nesting, maximum argument number in decisions per subprogram and the summation of argument numbers in decisions per subprogram) do affect the difference between the necessary test cases for DC and MC/DC.

The whole projects				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
22842	54995	59024	4029	1.07

4.7.4.1 Grouping by McCabe metrics

Subprograms where McCabe metrics are between 0 and 10				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
21306	37522	40027	2505	1.07

<table>
<tr><th colspan="5">Subprograms where McCabe metrics are between 11 and 20</th></tr>
<tr><td>Nr. of Subpr.</td><td>DC</td><td>MCDC</td><td>Difference</td><td>Ratio</td></tr>
<tr><td>923</td><td>6931</td><td>7541</td><td>610</td><td>1.09</td></tr>
</table>

<table>
<tr><th colspan="5">Subprograms where McCabe metrics are between 21 and 30</th></tr>
<tr><td>Nr. of Subpr.</td><td>DC</td><td>MCDC</td><td>Difference</td><td>Ratio</td></tr>
<tr><td>294</td><td>3210</td><td>3431</td><td>222</td><td>1.07</td></tr>
</table>

<table>
<tr><th colspan="5">Subprograms where McCabe metrics are between 31 and 40</th></tr>
<tr><td>Nr. of Subpr.</td><td>DC</td><td>MCDC</td><td>Difference</td><td>Ratio</td></tr>
<tr><td>141</td><td>1903</td><td>2114</td><td>211</td><td>1.11</td></tr>
</table>

<table>
<tr><th colspan="5">Subprograms where McCabe metrics are more than 40</th></tr>
<tr><td>Nr. of Subpr.</td><td>DC</td><td>MCDC</td><td>Difference</td><td>Ratio</td></tr>
<tr><td>178</td><td>5429</td><td>5911</td><td>482</td><td>1.09</td></tr>
</table>

4.7.4.2 Grouping by necessary MC/DC test cases

<table>
<tr><th colspan="5">Subprograms where number of MC/DC test cases are between 1 and 2</th></tr>
<tr><td>Nr. of Subpr.</td><td>DC</td><td>MCDC</td><td>Difference</td><td>Ratio</td></tr>
<tr><td>17369</td><td>21990</td><td>21990</td><td>0</td><td>1.00</td></tr>
</table>

<table>
<tr><th colspan="5">Subprograms where number of MC/DC test cases are between 3 and 4</th></tr>
<tr><td>Nr. of Subpr.</td><td>DC</td><td>MCDC</td><td>Difference</td><td>Ratio</td></tr>
<tr><td>2963</td><td>9111</td><td>9992</td><td>881</td><td>1.09</td></tr>
</table>

<table>
<tr><th colspan="5">Subprograms where number of MC/DC test cases are between 5 and 7</th></tr>
<tr><td>Nr. of Subpr.</td><td>DC</td><td>MCDC</td><td>Difference</td><td>Ratio</td></tr>
<tr><td>1356</td><td>6756</td><td>7721</td><td>965</td><td>1.14</td></tr>
</table>

Subprograms where number of MC/DC test cases are between 8 and 10				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
464	3538	4076	538	1.15

Subprograms where number of MC/DC test cases are more than 10				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
690	13600	15245	1645	1.12

4.7.4.3 Grouping by nesting values

Subprograms where the maximum nesting is between 0 and 1				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
17294	28291	29713	1422	1.05

Subprograms where the maximum nesting is between 2 and 3				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
4049	15167	16399	1232	1.08

Subprograms where the maximum nesting is between 4 and 6				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
1288	9152	10211	1059	1.12

Subprograms where the maximum nesting is above than 7				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
211	2385	2701	316	1.13

4.7.5 Grouping by maximum arguments number

<table>
<tr><th colspan="5">Subprograms where there are no decisions</th></tr>
<tr><td>Nr. of Subpr.</td><td>DC</td><td>MCDC</td><td>Difference</td><td>Ratio</td></tr>
<tr><td>12827</td><td>12827</td><td>12827</td><td>0</td><td>1.00</td></tr>
</table>

<table>
<tr><th colspan="5">Subprograms where the argument numbers in decisions are exactly 1</th></tr>
<tr><td>Nr. of Subpr.</td><td>DC</td><td>MCDC</td><td>Difference</td><td>Ratio</td></tr>
<tr><td>7839</td><td>28350</td><td>28350</td><td>0</td><td>1.00</td></tr>
</table>

<table>
<tr><th colspan="5">Subprograms where the maximum of argument numbers in decisions is between 2 and 3</th></tr>
<tr><td>Nr. of Subpr.</td><td>DC</td><td>MCDC</td><td>Difference</td><td>Ratio</td></tr>
<tr><td>1778</td><td>10851</td><td>12989</td><td>2138</td><td>1.19</td></tr>
</table>

<table>
<tr><th colspan="5">Subprograms where the maximum of argument numbers in decisions is between 4 and 5</th></tr>
<tr><td>Nr. of Subpr.</td><td>DC</td><td>MCDC</td><td>Difference</td><td>Ratio</td></tr>
<tr><td>229</td><td>1633</td><td>2354</td><td>721</td><td>1.44</td></tr>
</table>

<table>
<tr><th colspan="5">Subprograms where the maximum of argument numbers in decisions is between 6 and 10</th></tr>
<tr><td>Nr. of Subpr.</td><td>DC</td><td>MCDC</td><td>Difference</td><td>Ratio</td></tr>
<tr><td>121</td><td>875</td><td>1595</td><td>720</td><td>1.82</td></tr>
</table>

<table>
<tr><th colspan="5">Subprograms where the maximum of argument numbers in decisions is above than 10</th></tr>
<tr><td>Nr. of Subpr.</td><td>DC</td><td>MCDC</td><td>Difference</td><td>Ratio</td></tr>
<tr><td>48</td><td>459</td><td>909</td><td>450</td><td>1.98</td></tr>
</table>

4.7.6 Grouping by the summation of arguments in decisions

Subprograms where the summation of argument numbers in decisions is between 1 and 5				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
7136	17897	19103	1206	1.07

Subprograms where the summation of argument numbers in decisions is between 6 and 10				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
1448	6874	7620	746	1.11

Subprograms where the summation of argument numbers in decisions is between 11 and 50				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
1327	13433	15033	1600	1.12

Subprograms where the summation of argument numbers in decisions is between 51 and 100				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
76	2217	2545	328	1.15

Subprograms where the summation of argument numbers in decisions is more than 100				
Nr. of Subpr.	DC	MCDC	Difference	Ratio
28	1747	1896	149	1.09

4.7.7 Difference between the necessary of DC and MC/DC test cases

In this chapter you can see the number of subprograms where the difference of necessary test cases are 0, 1, 2 ... The *Diff* means the difference between the necessary DC and MC/DC test cases. The *Subpr* means how many subprograms are in the project where the difference between the two types of test cases is in the previous column. The *Min*, *Max* mean the minimum, maximum of MC/DC test cases per subprogram, and *Avg*, *Dev* mean the average and the standard deviation both of MC/DC and DC.

				DC		MCDC	
Diff	**Subpr**	**Min**	**Max**	**Avg**	**Dev**	**Avg**	**Dev**
0	21059	1	175	2.15	4.00	2.15	4.00
1	1070	3	88	4.10	5.46	5.10	5.46
2	342	4	41	5.06	4.85	7.06	4.85
3	125	5	35	5.74	5.00	8.74	5.00
4	75	6	95	9.09	15.35	13.09	15.35
5	34	7	19	5.56	3.56	10.56	3.56
6	35	8	27	8.11	5.98	14.11	5.98
7	24	9	44	11.17	11.51	18.17	11.51
8	19	10	22	5.53	3.45	13.53	3.45
9	13	11	49	9.08	10.65	18.08	10.65
10	3	12	25	7	5.72	17	5.72
11	8	13	28	6.50	5.20	17.50	5.20
12	7	14	34	9.43	8.33	21.43	8.33
13	3	15	16	2.33	0.47	15.33	0.47
14	5	16	31	10.40	5.82	24.40	5.82
15	2	60	105	67.50	22.50	82.50	22.50
16	3	44	72	39.67	11.90	55.67	11.90
18	2	20	40	12	10	30	10
19	3	33	37	16.67	1.88	35.67	1.88
21	3	61	247	113.33	80.87	134.33	80.87

				DC		MCDC	
Diff	**Subpr**	**Min**	**Max**	**Avg**	**Dev**	**Avg**	**Dev**
22	1	145	145	123	0	145	0
23	2	25	25	2	0	25	0
25	1	61	61	36	0	61	0
27	1	68	68	41	0	68	0
29	1	39	39	10	0	39	0
31	1	74	74	43	0	74	0
36	1	77	77	41	0	77	0

4.7.8 DC - MC/DC

In this chapter you can find how many test cases are needed for the project to cover DC and MC/DC.

In the following table, the

A means:	the whole project,
B means:	the whole project without those subprograms which do not contain decision,
C means:	the whole project without those subprograms which contain decision with at least two arguments.

	DC	MC/DC	difference	ratio
A	54995	59024	4029	1.07
B	42168	46197	4029	1.09
C	13818	17847	4029	1.29

5 Summary and Conclusion

In this study we analyzed six projects written in Ada programming language. Our task was to estimate the difference of test cases needed to satisfy the requirements of Decision Coverage and Modified Condition / Decision Coverage.

The difference is about five to ten per cent depending the characteristics of the project. The main reason we could not achieve greater difference is the decisions in most subprograms have only one argument and there are several subprograms which do not contain decisions at all. If we exclude these subprograms we get four times bigger difference. Most of all, the maximum number of arguments in decisions affects the difference. For those subprograms where there are decisions with more than six arguments, almost twice MC/DC test cases are needed than DC. But unfortunately these subprograms are only less than one per cent of the whole projects.

In general we can say almost ten per cent more test cases are needed to satisfy the requirements of Modified Condition / Decision Coverage than Decision Coverage.

6 References

[1] Kelly J. Hayhurst, Dan S. Veerhusen, John J. Chilenski, Leanna K. Rierson: A Practical Tutorial on Modified Condition/Decision Coverage

[2] http://www.bullseye.com/coverage.html 2008.03.01

[3] http://www.antlr.org/ 2008.03.02

[4] Oliver Kellogg: http://www.antlr.org/grammar/ada

[5] https://lemon.cs.elte.hu/site/ 2008.03.09

[6] http://www.gnu.org/software/glpk/ 2008.03.09