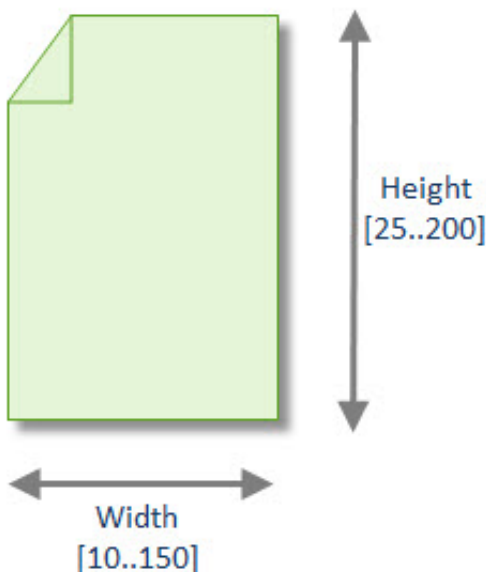


DOMAIN ANALYSIS TESTING



Written by **Sergei Terekhov** | Wednesday, 12 March 2014

Here is a simple example: you should test creation of a new page. User can input width and height of the page (2 parameters). Width can have the integer value from 10 to 150 inclusive. Height can have the integer value from 25 to 200 inclusive. How many tests should you execute in this case?



In order to define effective tests, I recommend using Domain analysis testing in this case. The technique supposes the following operations:

- 1) Split the values of each parameter into equivalence classes
- 2) Define strict bounds for each class
- 3) Find ON, OFF and IN points for each boundary
- 4) Make up the test cases, so thus ON and OFF values of the first parameter will be checked with IN values of the rest parameters, then ON and OFF values of the second parameter will be checked with IN values of the rest parameters, and so on.

The first and second steps for our example are:

- Width – one class of correct values [10..150]; bounds 10 and 150
- Height – one class of correct values [25..200]; bounds 25 and 200

Now let's define which points are ON, OFF and IN.

ON point is a point that lies on a bound. In the example, ON points for width are 10 and 150; ON points for height are 25 and 200

OFF point is a point that lies near the bound outside the domain. In the example, OFF points for width are 9 and 151; OFF points for height are 24 and 201.

IN point is a point inside the domain (typical value). In the example, IN points for width and height can be 100.

So, now we must make up the test cases. For this purpose, Binder's Domain Test Matrix is very helpful. I have designed the test cases very easy:

			Test cases							
			1	2	3	4	5	6	7	8
Width	>=10	ON	10							
		OFF		9						
	<=150	ON			150					
		OFF				151				
	Typical	IN					100	100	100	100
Height	>=25	ON					25			
		OFF						24		
	<=200	ON							200	
		OFF								201
	Typical	IN	100	100	100	100				
Expected result			OK		OK		OK		OK	

Look, in the test cases #1-4 ON and OFF values of the width are checked with IN value of the height, and in the test cases #5-8 ON and OFF values of the height are checked with IN value of the width.

According to Domain analysis testing, I've got 8 tests. 4 of them are positive, and 4 are negative. Of course, this test set should be supplemented with such tests as input float number, leave the field empty, input string value, etc. But these are negative test cases; all positive tests (only 4) have been already designed. It is great, isn't it?

Additional parameters will follow the same algorithm. Let's introduce "Height/Width" parameter into our example. Ratio of Height/Width mustn't be less than 1. Then, there's one class of correct values $[1..+\infty)$ for ratio; and the boundary 1. ON point is 1, OFF is 0,99, and IN is any value more than 1.

I added Domain Test Matrix with a new parameter and changed some typical values (IN points) of other parameters, because positive test results are limited to both ranges of width and height and their ratio now.

			Test cases									
			1	2	3	4	5	6	7	8	9	10
Width	>=10	ON	10									
		OFF	9									
	<=150	ON	150									
		OFF	151									
	Typical	IN	20 20 100 100 100 100									
Height	>=25	ON	25									
		OFF	24									
	<=200	ON	200									
		OFF	201									
	Typical	IN	100	100	190	190					100	99
H/W	>=1	ON	1									
		OFF	0,99									
	Typical	IN	10	11,11	1,27	1,26	1,25	1,2	2	2,01		
Expected result			OK		OK		OK		OK		OK	

Note that you have to select one ON point and one OFF point per each relational condition (\geq or \leq). For each strict equality condition ($=$), you have to select one ON point and two OFF points (slightly less and slightly more than the conditional value). Also it's meaningless to replicate the same tests for neighboring domains. If OFF point for one domain is IN point for another, do not repeat these test cases.

So, we have briefly considered Domain Analysis Testing. You can read about this technique more detailed in "A Practitioner's Guide to Software Test Design" by Lee Copeland. Are you interested in software testing?