

CENG552 Software Testing Project

Graph-based Testing Algorithm 1

Senem Kaplan
Azize Karagöz

- The project covers all prime paths according to the triangle classification problem and next date problem. These all paths also covers whole nodes, edges and edge-pairs inside, as well.

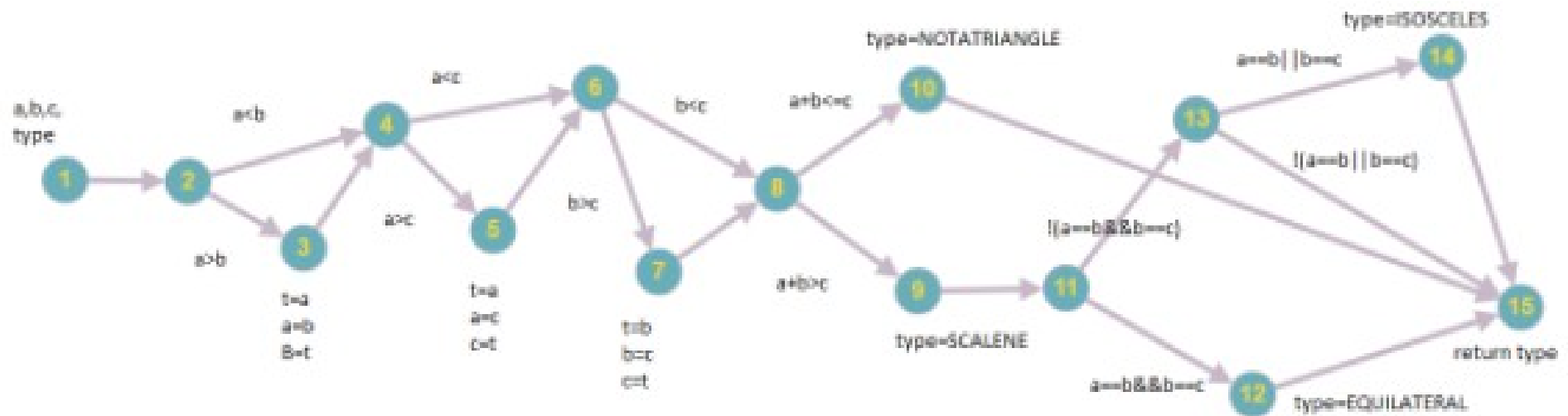
Node, Vertex and Graph are the classes related the graph. In Search class, the prime paths are found according the depth first search. UploadFile.java, Download.java and ReadAndWriteExcelFile.java files do uploading, downloading and reading and writing excel files, respectively. The software works well in Netbeans 8.2 environment.

For Triangle classification problem, this pseudocode is used for the graph

Triangle Classification Program

```
int tri_type(int a, int b, int c) {  
    int type;  
    if (a > b) { int t = a; a = b; b = t; }  
    if (a > c) { int t = a; a = c; c = t; }  
    if (b > c) { int t = b; b = c; c = t; }  
    if (a + b <= c) {  
        type = NOT_A_TRIANGLE;  
    } else {  
        type = SCALENE;  
        if (a == b && b == c) {  
            type = EQUILATERAL;  
        } else if (a == b || b == c) {  
            type = ISOSCELES;  
        }  
    }  
    return type;  
}
```

This is a graph representation based the previous(triangle classification) pseudo-code



In triangleInput.xls file, there are inputs according our graph representation. Nodes are numbered. Each cell refers to a node and in each row, the pair of cells are represented as directed edges.

There are 15 nodes and 20 edges in this example.

triangleInput.xls - LibreOffice Calc

Dosya Düzenle Görünüm Ekle Biçim Araçlar

Arial 10

	A	B	C
1	1	2	
2	2	3	
3	2	4	
4	3	4	
5	4	5	
6	4	6	
7	5	6	
8	6	7	
9	6	8	
10	7	8	
11	8	9	
12	8	10	
13	10	15	
14	9	11	
15	11	12	
16	11	13	
17	12	15	
18	13	15	
19	13	14	
20	14	15	

So, to get valid results, in our software system, inputs must be number and cell pair as in triangleInput file. So, after uploaded file, we will see under button the uploaded file name and whenever to click prime paths, the output file are downloaded named as result.xls. Always, the last uploaded file is downloaded.

When clicking Prime Paths button, it brings all prime paths between first node and last node. It could be changed inside our software. But to see all coverage, it is fixed according the first and last nodes inside input files.

In triangleInput.xls file, there are inputs according our graph representation. Nodes are numbered. Each cell refers to a node and in each row, the pair of cells are represented as directed edges.

There are 15 nodes and 20 edges in this example. The first node is set to one and last node set to 15. So, it brings 32 prime paths between node 1 and node 15. The output will be like this in a result.xls file.

triangleOutput.xls - LibreOffice Calc

Dosya Düzenle Görünüm Ekle Biçim Araçlar Veri Pencere Yardım

Arial 10

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	1	2	3	4	5	6	7	8	9	11	12	15								
2	1	2	3	4	5	6	7	8	9	11	13	15								
3	1	2	3	4	5	6	7	8	9	11	13	14	15							
4	1	2	3	4	5	6	7	8	10	15										
5	1	2	3	4	5	6	8	9	11	12	15									
6	1	2	3	4	5	6	8	9	11	13	15									
7	1	2	3	4	5	6	8	9	11	13	14	15								
8	1	2	3	4	5	6	8	10	15											
9	1	2	3	4	6	7	8	9	11	12	15									
10	1	2	3	4	6	7	8	9	11	13	15									
11	1	2	3	4	6	7	8	9	11	13	14	15								
12	1	2	3	4	6	7	8	10	15											
13	1	2	3	4	6	8	9	11	12	15										
14	1	2	3	4	6	8	9	11	13	15										
15	1	2	3	4	6	8	9	11	13	14	15									
16	1	2	3	4	6	8	10	15												
17	1	2	4	5	6	7	8	9	11	12	15									
18	1	2	4	5	6	7	8	9	11	13	15									
19	1	2	4	5	6	7	8	9	11	13	14	15								
20	1	2	4	5	6	7	8	10	15											
21	1	2	4	5	6	8	9	11	12	15										
22	1	2	4	5	6	8	9	11	13	15										
23	1	2	4	5	6	8	9	11	13	14	15									
24	1	2	4	5	6	8	10	15												
25	1	2	4	6	7	8	9	11	12	15										
26	1	2	4	6	7	8	9	11	13	15										
27	1	2	4	6	7	8	9	11	13	14	15									
28	1	2	4	6	7	8	10	15												
29	1	2	4	6	8	9	11	12	15											
30	1	2	4	6	8	9	11	13	15											
31	1	2	4	6	8	9	11	13	14	15										
32	1	2	4	6	8	10	15													

Sheet1

13:59 8.1.2018

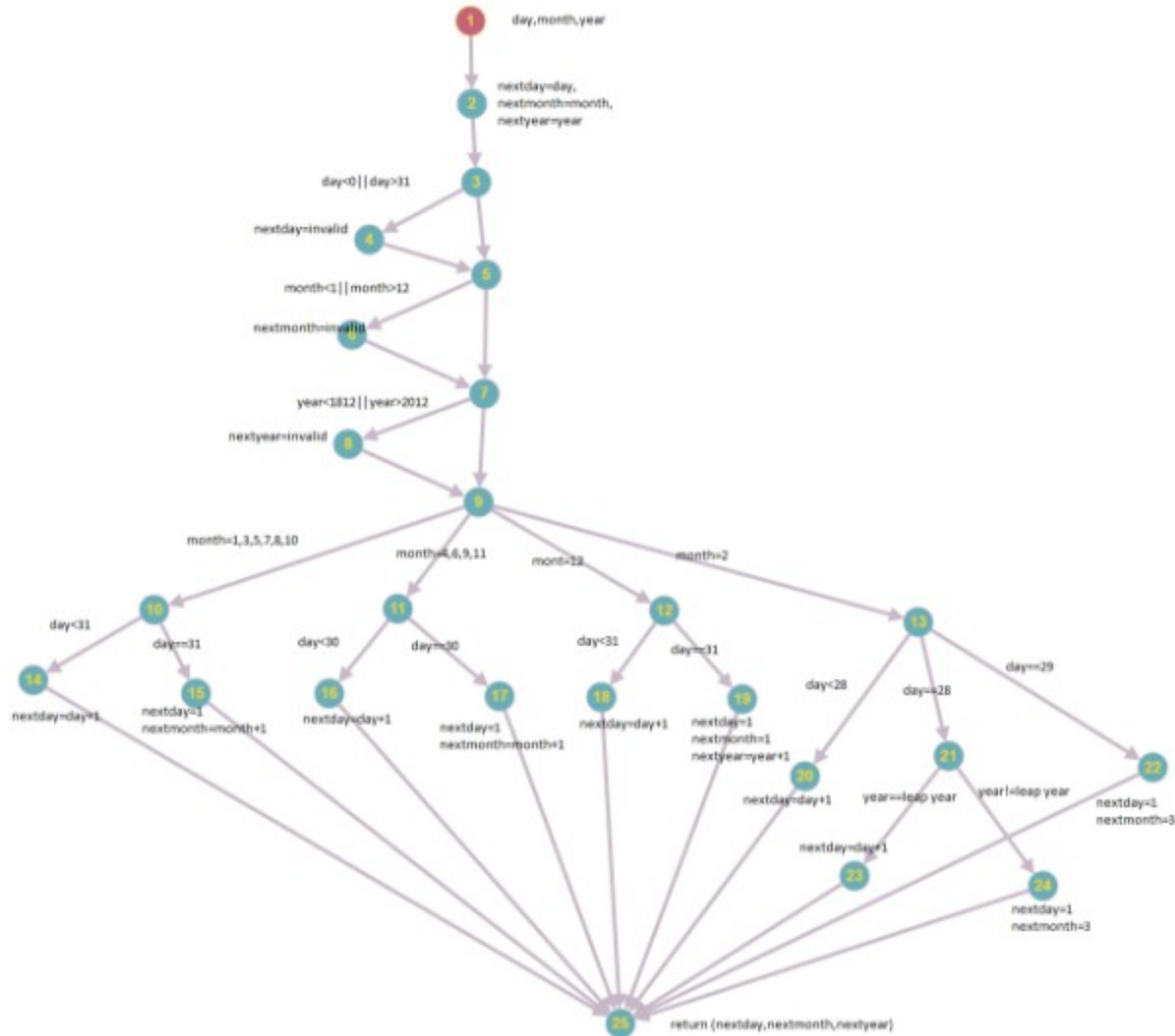
This is the pseudocode for next date problem

```
List(int) NextDate(int day,int month, int year){  
  
    int nextday int= day;  
    int nextmonth = month;  
    int nextyear = year;  
  
    if(day<=0 || day>31)  
        nextday=invalid;  
  
    if(month<=0 || month>12)  
        nextmonth=invalid;  
  
    if(year<1812 || day>2012)  
        nextyear=invalid;  
  
    switch(month){  
  
        case 1:{1,3,5,7,8,10}  
  
            if(day<31){  
                day=day+1;  
                break;  
            }  
            elseif(day==31){  
                nextday=1;  
                nextmonth=month+1;  
                break;  
            }  
  
        case 2:{4,6,9,11}  
  
            if(day<30)  
                nextday=day+1;  
                break;  
            elseif(day==30){  
                nextday=1;  
                nextmonth=month+1;  
                break;  
            }  
  
    }
```

```
case 3:{12}  
  
    if(day<31){  
        nextday=day+1;  
        break;  
    }  
    elseif(day==31){  
        nextday=1;  
        nextmonth=1;  
        nextyear=year+1;  
        break;  
    }  
}
```

```
case 4:{2}  
  
    if(day>29){  
        nextday=invalid;  
        break;  
    }  
    elseif(day<28){  
        nextday=day+1;  
        break;  
    }  
    elseif(day==28){  
        if(year is a leap year){  
            nextday=day+1;  
            break;  
        }  
        else{  
            nextday=1;  
            nextmonth=3;  
            break;  
        }  
    }  
    elseif(day==29){  
        nextday=1;  
        nextmonth=3;  
        break;  
    }  
}  
return (nextday,nextmonth,nextyear);
```


This is a graph representation based the previous(next date) pseudo-code



In dateInput.xls file, there are inputs according our graph representation. Nodes are numbered. Each cell refers to a node and in each row, the pair of cells are represented as directed edges.

There are 27 nodes and 38 edges in this example. The first node is set to one and last node set to 15. So, it brings 80 prime paths between node 1 and node 27.

1	1	2
2	2	3
3	3	4
4	3	5
5	4	5
6	5	6
7	5	7
8	6	7
9	7	8
10	7	9
11	8	9
12	9	10
13	9	11
14	9	12
15	9	13
16	10	14
17	10	15
18	11	16
19	11	17
20	12	18
21	12	19
22	13	20
23	14	27
24	15	27
25	16	27
26	17	27
27	18	27
28	19	27
29	20	21
30	20	22
31	20	23
32	21	27
33	22	24
34	23	27
35	24	25
36	24	26
37	25	27
38	26	27

In dateInput.xls file, there are inputs according our graph representation. Nodes are numbered. Each cell refers to a node and in each row, the pair of cells are represented as directed edges.

There are 27 nodes and 38 edges in this example. The first node is set to one and last node set to 27. So, it brings 80 prime paths between node 1 and node 27. The output will be like the following.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	1	2	3	4	5	6	7	8	9	10	14	27			
2	1	2	3	4	5	6	7	8	9	10	15	27			
3	1	2	3	4	5	6	7	8	9	11	16	27			
4	1	2	3	4	5	6	7	8	9	11	17	27			
5	1	2	3	4	5	6	7	8	9	12	18	27			
6	1	2	3	4	5	6	7	8	9	12	19	27			
7	1	2	3	4	5	6	7	8	9	13	20	21	27		
8	1	2	3	4	5	6	7	8	9	13	20	22	24	25	27
9	1	2	3	4	5	6	7	8	9	13	20	22	24	26	27
10	1	2	3	4	5	6	7	8	9	13	20	23	27		
11	1	2	3	4	5	6	7	9	10	14	27				
12	1	2	3	4	5	6	7	9	10	15	27				
13	1	2	3	4	5	6	7	9	11	16	27				
14	1	2	3	4	5	6	7	9	11	17	27				
15	1	2	3	4	5	6	7	9	12	18	27				
16	1	2	3	4	5	6	7	9	12	19	27				
17	1	2	3	4	5	6	7	9	13	20	21	27			
18	1	2	3	4	5	6	7	9	13	20	22	24	25	27	
19	1	2	3	4	5	6	7	9	13	20	22	24	26	27	
20	1	2	3	4	5	6	7	9	13	20	23	27			
21	1	2	3	4	5	7	8	9	10	14	27				
22	1	2	3	4	5	7	8	9	10	15	27				
23	1	2	3	4	5	7	8	9	11	16	27				
24	1	2	3	4	5	7	8	9	11	17	27				
25	1	2	3	4	5	7	8	9	12	18	27				
26	1	2	3	4	5	7	8	9	12	19	27				
27	1	2	3	4	5	7	8	9	13	20	21	27			
28	1	2	3	4	5	7	8	9	13	20	22	24	25	27	
29	1	2	3	4	5	7	8	9	13	20	22	24	26	27	
30	1	2	3	4	5	7	8	9	13	20	23	27			

31	1	2	3	4	5	7	9	10	14	27				
32	1	2	3	4	5	7	9	10	15	27				
33	1	2	3	4	5	7	9	11	16	27				
34	1	2	3	4	5	7	9	11	17	27				
35	1	2	3	4	5	7	9	12	18	27				
36	1	2	3	4	5	7	9	12	19	27				
37	1	2	3	4	5	7	9	13	20	21	27			
38	1	2	3	4	5	7	9	13	20	22	24	25	27	
39	1	2	3	4	5	7	9	13	20	22	24	26	27	
40	1	2	3	4	5	7	9	13	20	23	27			
41	1	2	3	5	6	7	8	9	10	14	27			
42	1	2	3	5	6	7	8	9	10	15	27			
43	1	2	3	5	6	7	8	9	11	16	27			
44	1	2	3	5	6	7	8	9	11	17	27			
45	1	2	3	5	6	7	8	9	12	18	27			
46	1	2	3	5	6	7	8	9	12	19	27			
47	1	2	3	5	6	7	8	9	13	20	21	27		
48	1	2	3	5	6	7	8	9	13	20	22	24	25	27
49	1	2	3	5	6	7	8	9	13	20	22	24	26	27
50	1	2	3	5	6	7	8	9	13	20	23	27		
51	1	2	3	5	6	7	9	10	14	27				
52	1	2	3	5	6	7	9	10	15	27				
53	1	2	3	5	6	7	9	11	16	27				
54	1	2	3	5	6	7	9	11	17	27				
55	1	2	3	5	6	7	9	12	18	27				
56	1	2	3	5	6	7	9	12	19	27				
57	1	2	3	5	6	7	9	13	20	21	27			
58	1	2	3	5	6	7	9	13	20	22	24	25	27	
59	1	2	3	5	6	7	9	13	20	22	24	26	27	
60	1	2	3	5	6	7	9	13	20	23	27			
61	1	2	3	5	7	8	9	10	14	27				
62	1	2	3	5	7	8	9	10	15	27				
63	1	2	3	5	7	8	9	11	16	27				
64	1	2	3	5	7	8	9	11	17	27				
65	1	2	3	5	7	8	9	12	18	27				
66	1	2	3	5	7	8	9	12	19	27				
67	1	2	3	5	7	8	9	13	20	21	27			
68	1	2	3	5	7	8	9	13	20	22	24	25	27	
69	1	2	3	5	7	8	9	13	20	22	24	26	27	
70	1	2	3	5	7	8	9	13	20	23	27			
71	1	2	3	5	7	9	10	14	27					
72	1	2	3	5	7	9	10	15	27					
73	1	2	3	5	7	9	11	16	27					
74	1	2	3	5	7	9	11	17	27					
75	1	2	3	5	7	9	12	18	27					
76	1	2	3	5	7	9	12	19	27					
77	1	2	3	5	7	9	13	20	21	27				
78	1	2	3	5	7	9	13	20	22	24	25	27		
79	1	2	3	5	7	9	13	20	22	24	26	27		
80	1	2	3	5	7	9	13	20	23	27				