

Visualizing Path Exploration to Assist Problem Diagnosis for Structural Test Generation

Jiayi Cao¹, Angello Astorga¹, Siwakorn Srisakaokul¹, Zhengkai Wu¹,
Xueqing (Susan) Liu¹, Xusheng Xiao², Tao Xie¹

Email: {jcao7,aastorg2,srisaka2,zw3,xliu93,taoxie}@illinois.edu, xusheng.xiao@case.edu

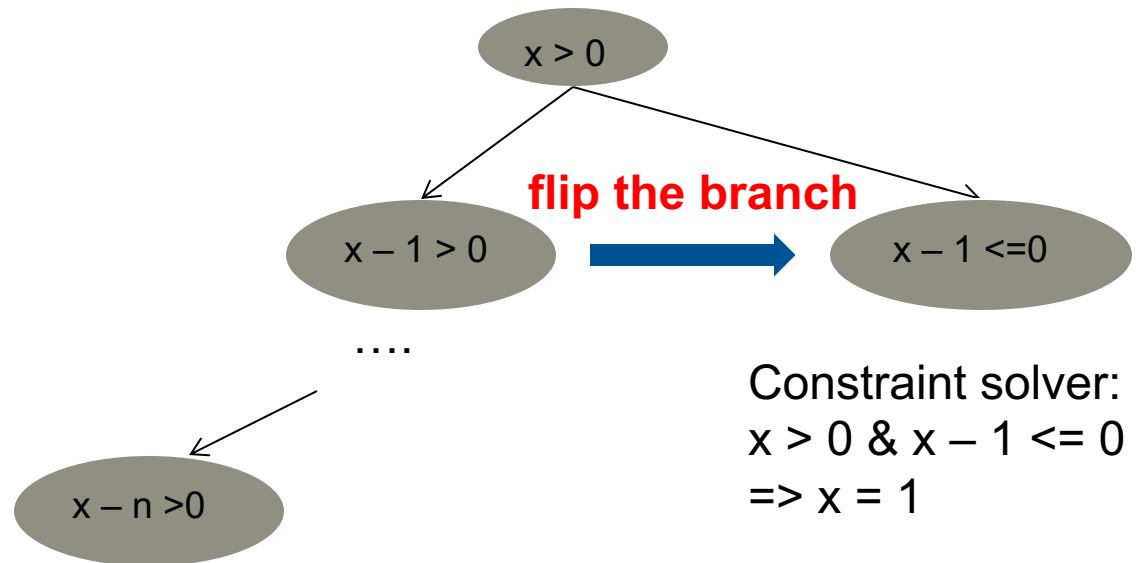
UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN



illinois.edu

Test Generation with Dynamic Symbolic Execution (DSE)

Input: x ;
while ($x > 0$)
 $x--$;



Symbol execution tree



Boundary Problem

```
public void BubbleSort(int[] array, int n){  
    for (int i = 0; i < array.length; i++)  
    {  
        for (j = 0; j < n; j++){  
            //...  
        }  
    }  
    if (array.Length > 200){  
        throw new Exception("bug");  
    }  
}
```

bug



Cannot reach the bug!

1. Incremental test generation
on number.Length

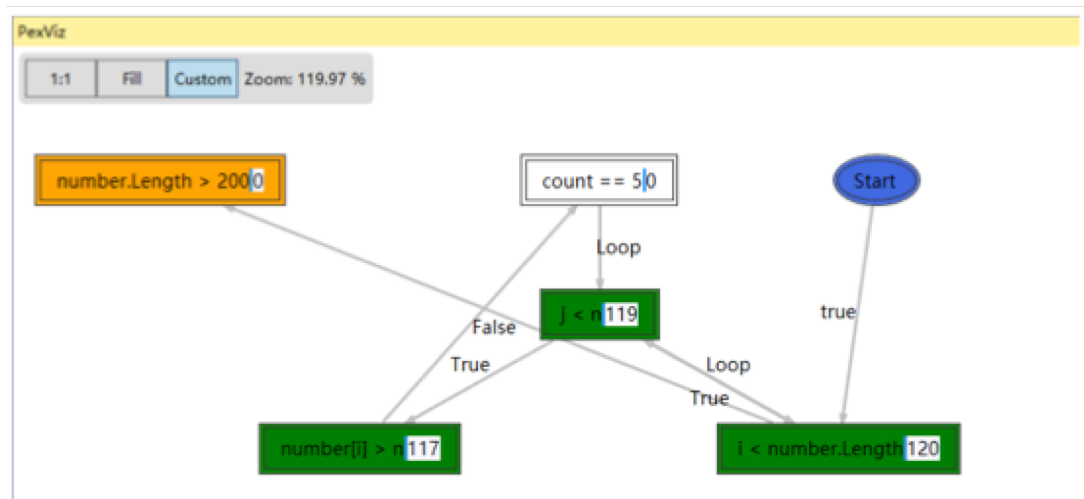
2. Inner loop: constraint solver
cost increases with n

3. Time cost for reaching the
bug exceeds the limit !



PexViz

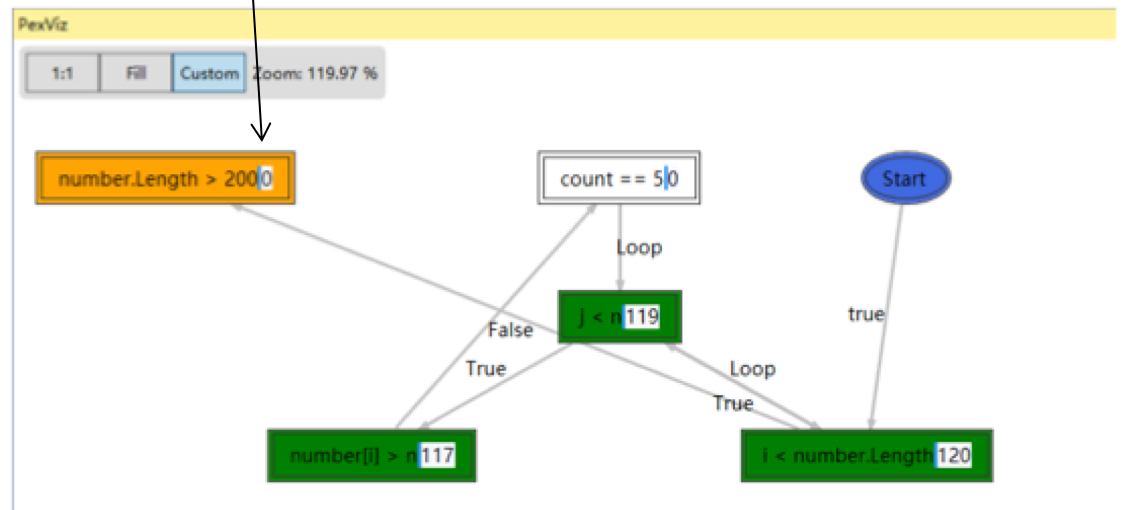
- **Objective:** Assist developers in diagnosing problems encountered during test generation using visualization



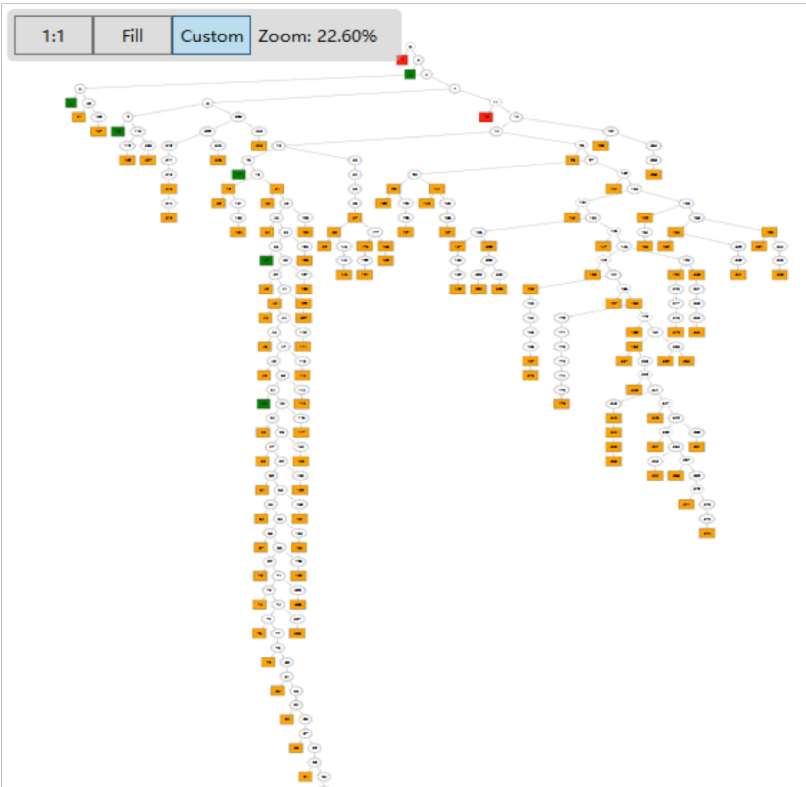
BubbleSort Example

Flip count = 0

Variant Control Flow Graph



Comparison with Baseline



- SEViz [1] outputs the symbolic execution tree → bad for visualization
- 6 nodes (PexViz) vs. 276 nodes (SEViz)

[1] SEViz: A Tool for Visualizing Symbolic Execution, Honfi et. al. ICST'15





Thank you!

Acknowledgment: NSF CNS-1513939, NSF CNS-1564274

UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN



illinois.edu