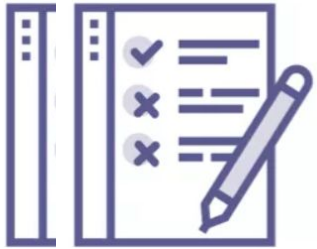
Automated Patching Using GP
Project Status

TEAM 8

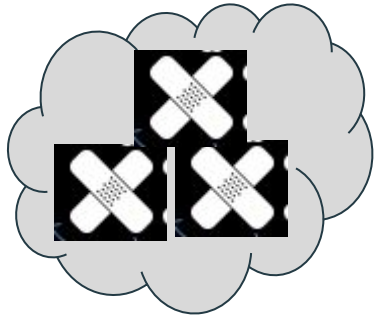
Peerapon Akkapusit, Zuzana Jelčicová, Zelalem Mihret, Peter Muschick

Buggy program

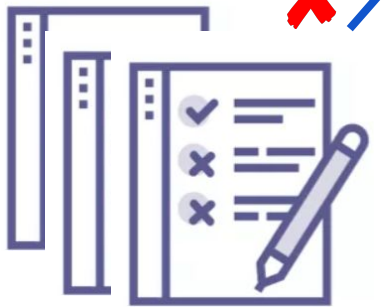


Fault localization

GA operations

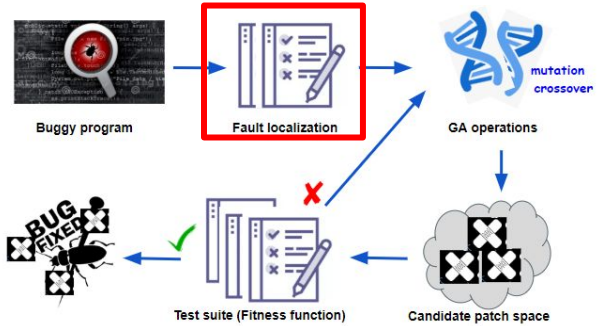


Candidate patch space



Test suite (Fitness function)

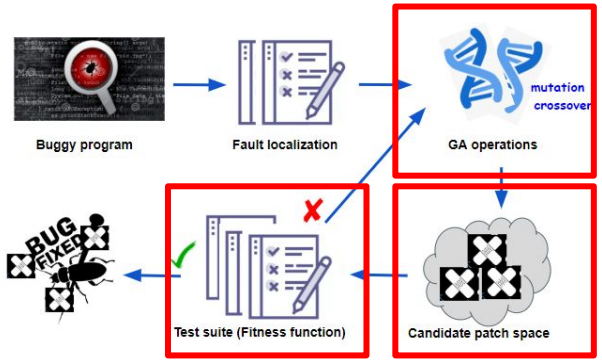
Fault Localization




```
15⊖    public String giffString(String a){
16         Fault likelihood: 0.7071067811865476")){
17             return a;
18         }else if (a.contains("b")) {
19             return a;
20         }else if (a.contains("c")) {
21             return a;
22         }else {
23             return "x";
24         }
25     }
```

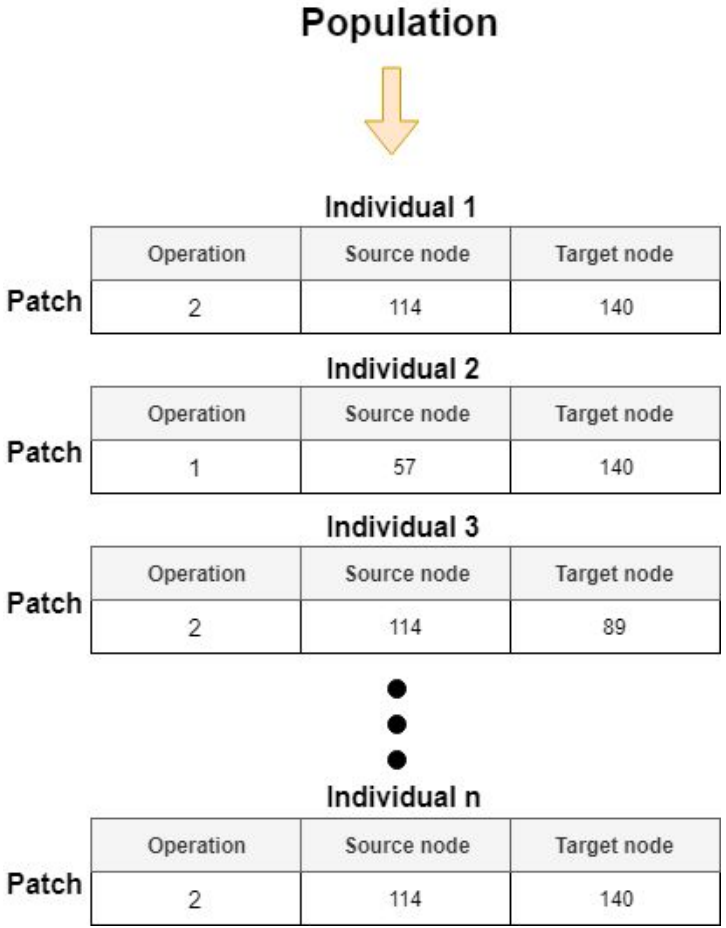
```
#16,0,70710678118654760000000000000000
#17,0,66666666666666660000000000000000
#18,0,33333333333333330000000000000000
#19,0,40824829046386310000000000000000
#20,0,00000000000000000000000000000000
#21,0,00000000000000000000000000000000
#23,0,00000000000000000000000000000000
```

Genetic Algorithm



Create initial population
Loop: population
Apply Patches of the Individual
Parse from AST to Java and Compile
Run JUnit tests and calculate Fitness
Create new population - perform mutations/crossovers

1. **Create initial population**



```
Patch(INSERT,  sNode: 106,  tNode: 96));
Patch(REPLACE,  sNode: 114,  tNode: 121));
Patch(DELETE,  sNode: -1,  tNode: 106));
Patch(DELETE,  sNode: -1,  tNode: 96));
```

2. **Apply Patches of Individual**
3. **Parse AST to java and compile**

```
    if(year % 4 == 0){
        if(year % 100 == 0){
            if (year % 400 == 0){
                leap = true;
            }else{
                leap = false;
            }
        }else{
            leap = true;
        }
    }else{
        leap = false;
    }

    if(leap){
        System.out.println(year + " is a leap year.");
    }else{
        System.out.println(year + " is not a leap year.");
    }
}
```

96 106 121 114 140

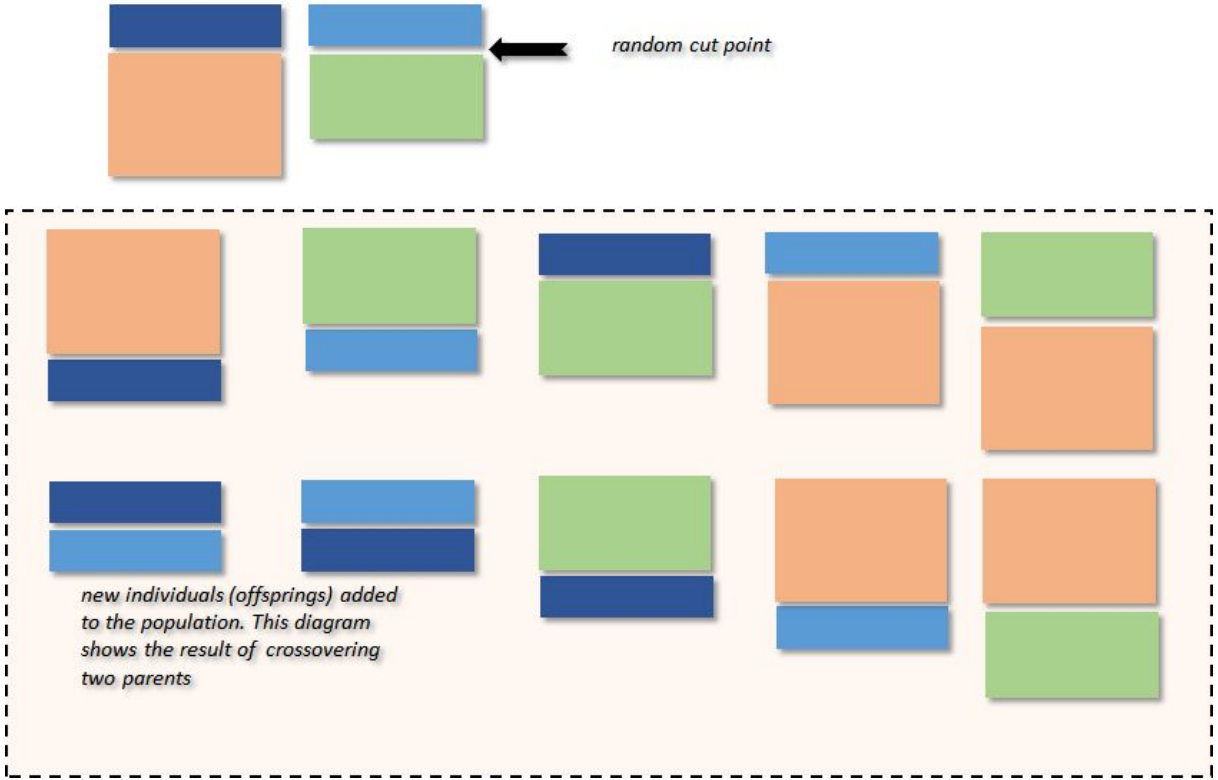
```
if (year % 4 == 0) {
    if (year % 100 == 0) {
        if (year % 400 == 0) {
            leap = true;
        } else {
            leap = false;
        }
    } else {
        leap = false;
    }
} else {

}

if (leap) {
    if (leap) {
        System.out.println(year + " is a leap year.");
    } else {
        System.out.println(year + " is not a leap year.");
    }
} else {
    System.out.println(year + " is not a leap year.");
```

96 106 114

5. Perform mutations and crossovers



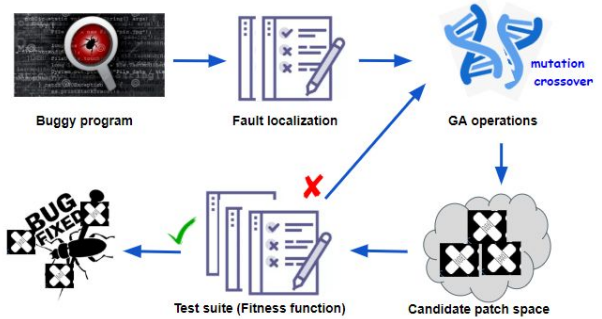
3,5,2	10,2,10	9,3,9
5,5,7	3,7,3	9,1,1
5,8,6	10,5,4	6,3,4
8,7,2	4,6,10	9,6,7
9,7,3	7,3,8	3,10,3

First population

10,2,10	9,3,9	4,6,10	9,6,7	9,3,9	6,3,4	10,2,10	9,3,9	10,2,10
3,7,3	9,1,1	7,3,8	3,10,3	9,1,1	9,6,7	3,7,3	9,1,1	3,7,3
10,5,4	6,3,4	9,3,9	10,2,10	6,3,4	3,10,3	6,3,4	10,5,4	6,3,4
9,6,7	4,6,10	9,1,1	3,7,3	9,6,7	10,2,10	9,6,7	4,6,10	9,6,7
3,10,3	7,3,8	6,3,4	10,5,4	3,10,3	3,7,3	3,10,3	7,3,8	3,10,3
				10,2,10		9,3,9	10,2,10	9,3,9
				3,7,3		9,1,1	3,7,3	9,1,1
						10,5,4	6,3,4	10,5,4

Crossovered population

Next milestone



- Finish Fitness function and JUnit tests
- Connect different modules
- Run tests
- Run experiments