

# Program Repair That Learns From Mistakes

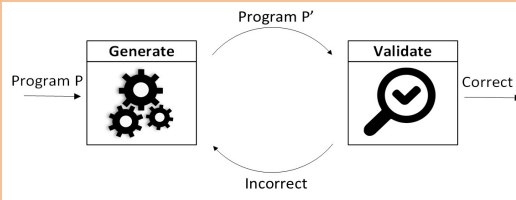
Mistakes

Bat-Chen Rothenberg and Orna Grumberg

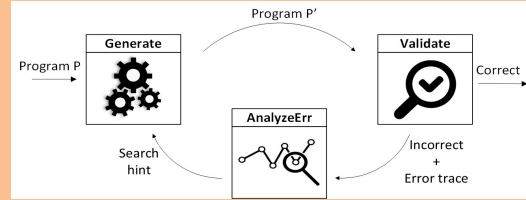


Computer Science Department, Technion – Israel Institute of Technology

## A novel working scheme for program repair



Generate and Validate



Generate – Validate – AnalyzeErr

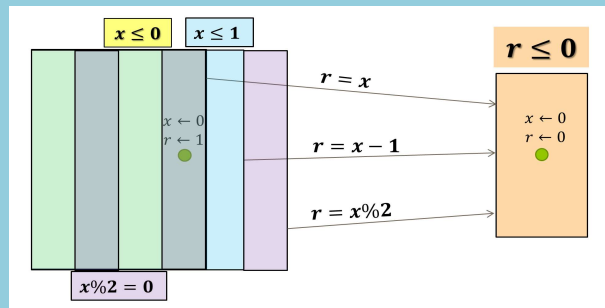
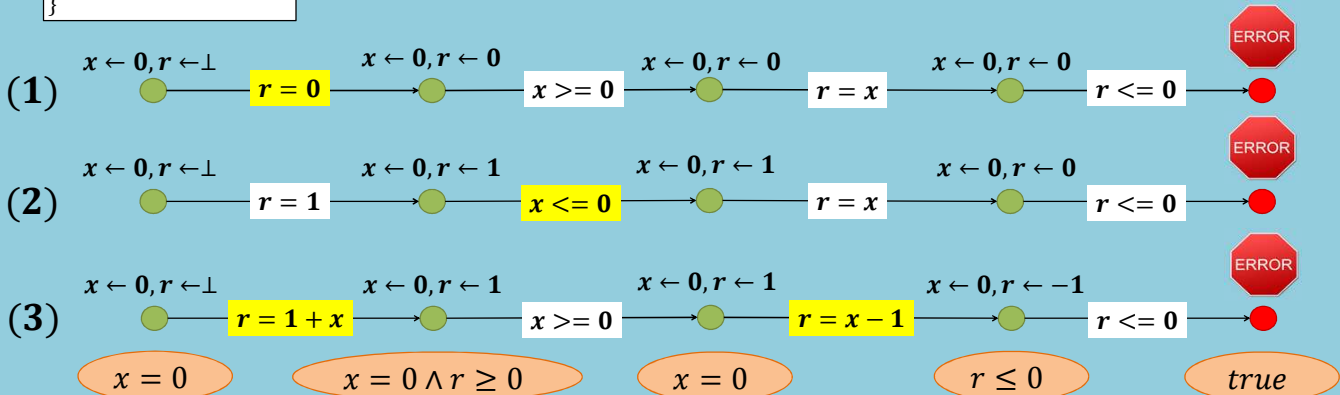
## Analyze error using error generalization

```
int max1 (int x){
  r = 1;
  if(x >= 0){
    r = x;
  }
  assert(r > 0);
}
```

$r = 0, \quad r = 2, \quad x = 1, \quad r = 1 + x$   
 $x > 0, \quad x \leq 0, \quad x \geq r, \quad x + r \geq 0$   
 $r = x + 1, \quad r = x - 1, \quad x = r, \quad r = x \% 2$

Generate using a SAT solver

Validate using an SMT solver



LEARN FROM YOUR MISTAKES



## Analyze error in other ways

### Prune the search space differently:

- Remove all programs that fail on same input
- Remove all programs that fail on same path
- Remove some (not all) programs
- Remove programs that only might fail

### Update search priority:

- Prioritize all locations along the path
- Use fault localization for the trace, and prioritize resulting locations
- Prioritize certain changes over others, based on observations in error traces

Your ideas ... ?