In-Class Exercise: Reasoning About Mutants

- Today we'll learn how to use Z3, a formal theorem prover
- And we'll use it to help us create tests

Z3 • Online interface: https://rise4fun.com/23 • Tutorial: https://rise4fun.com/23/tutorial/guide • In-class assignment: https://people.cs.umass.edu/~brun/class/2019Fall/CS520/in-class4.pdf

2

4

1

Z3's language

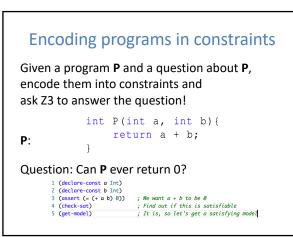
- Z3 uses a kind of programming language
- Can declare variables and functions, define constraints, print things to the screen, etc.

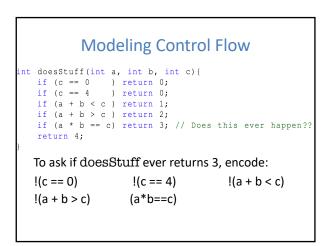
Z3's language

1 (echo "starting Z3...") 2 (declare-const a Int) 3 (declare-fun f (Int Bool) Int) 4 (assert (> a 100)) 5 (assert (< (f a true) 100)) 6 [check-sat]

This code prints "starting Z3..." to the screen, declares a constant a declares a function Int f (Int Bool) makes 2 assertions: a > 10 and f(a, true) < 100 asks "is this possible?"

3





5

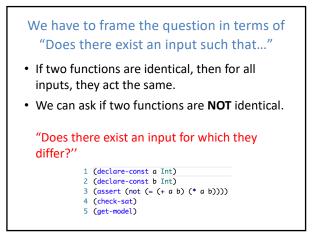
Modeling Control Flow int doesStuff(int a, int b, int c){ if (c == 0) return 0; if (c == 4) return 0; if (a + b < c) return 1; if (a + b < c) return 2; if (a * b == c) return 3; // Does this ever happen?? return 4; } / (declare-const a JInt) 3 (declare-const c JInt) 4 (declare-const c JInt) 5 (cassert (not (= c #x0000000)))) 7 (cassert (not (= c #x0000000))) 8 (cassert (not (bvslt (bvadd a b) c))) 9 (cassert (not (bvslt (bvadd a b) c))) 0 (cassert (- (bvwul a b) c)) il i2 (check-sat)

7

```
Z3 for Mutation Testing
int normal_sum(int a, int b){
   return a + b;
}
int mutant_sum(int a, int b){
   return a * b;
}

1 (declare-const a Int)
2 (declare-const b Int)
3 (assert (= (+ a b)(* a b)))
4 (check-sat)
5 (get-model)
```

8



9



- In-class assignment: https://people.cs.umass.edu/~brun/class/2018Fall/CS520/in-class4.pdf
- Online Z3 interface: https://rise4fun.com/Z3
- Tutorial: <u>https://rise4fun.com/Z3/tutorial/guide</u>

10