Symbolic Execution

...

Yegor Bugayenko

Lecture #8 out of 10
90 minutes

In Theory

In Practice

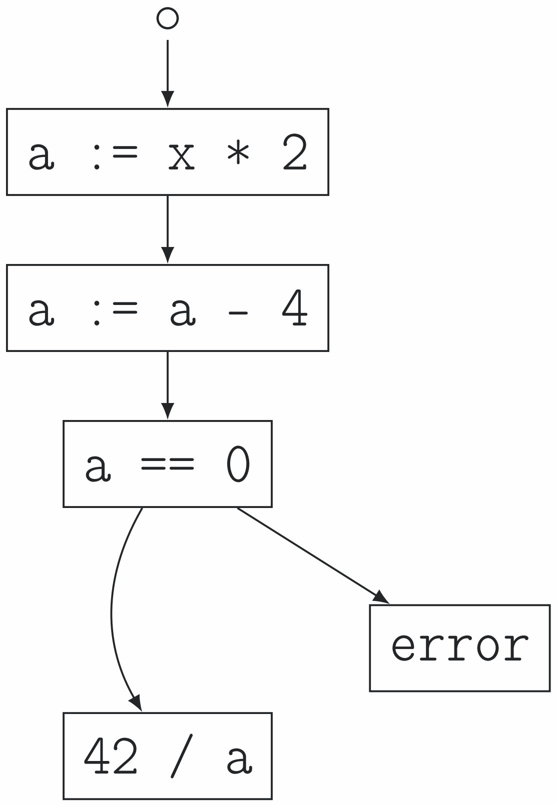
Concolic Execution

Chapter #1:

In Theory

Control Flow Graph

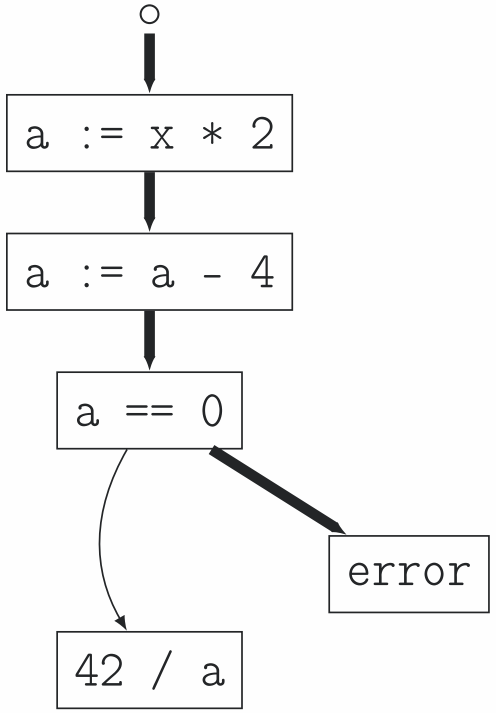
```
int f(int x) {
  int a = x * 2;
  a = a - 4;
  if (a == 0)
    error("Div by zero!");
  return 42 / a;
}
```



Path Feasibility

A path is feasible if there exists an input $\mathcal{I}$ to the program that covers the path; i.e., when program is executed with $\mathcal{I}$ as input, the path is taken.

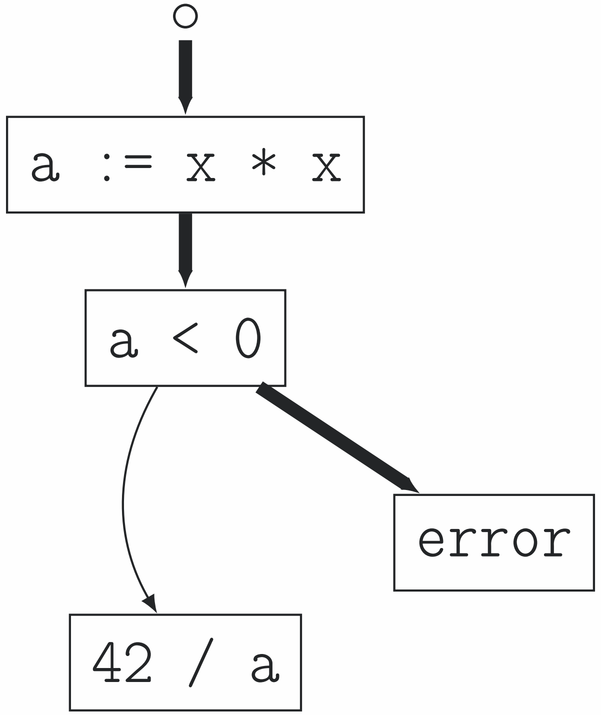
```
int f(int x) {
  int a = x * 2;
  a = a - 4;
  if (a == 0)
    error("Div by zero!");
  return 42 / a;
}
```



Infeasible Path

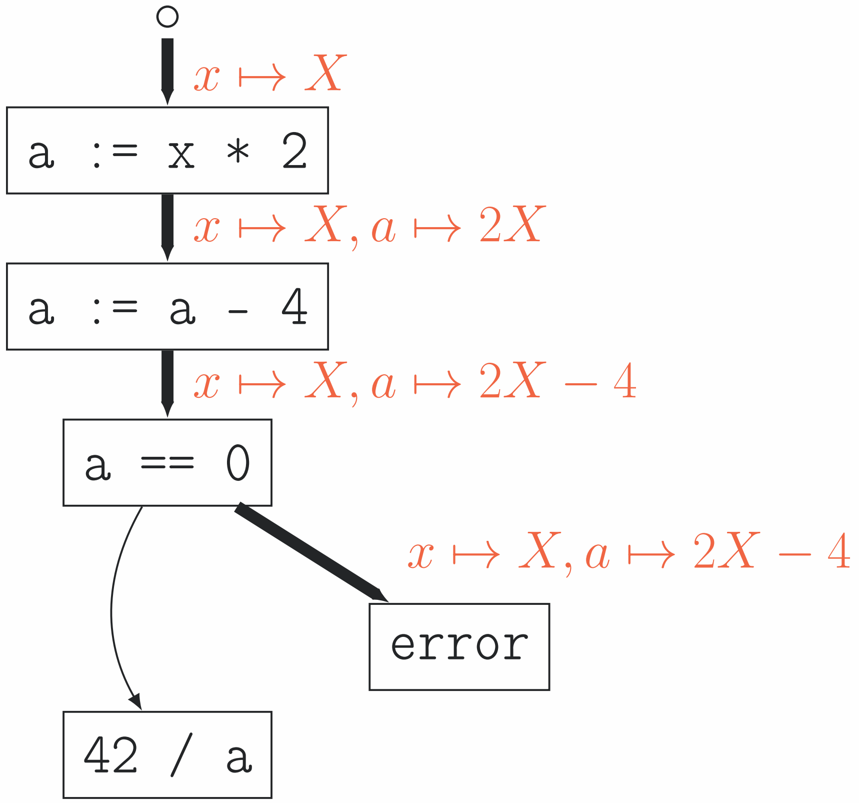
A path is infeasible if there exists no input $\mathcal{I}$ that covers the path.

```
int f(int x) {
  int a = x * x;
  if (a < 0)
    error("Too small!");
  return 42 / a;
}
```



Symbols

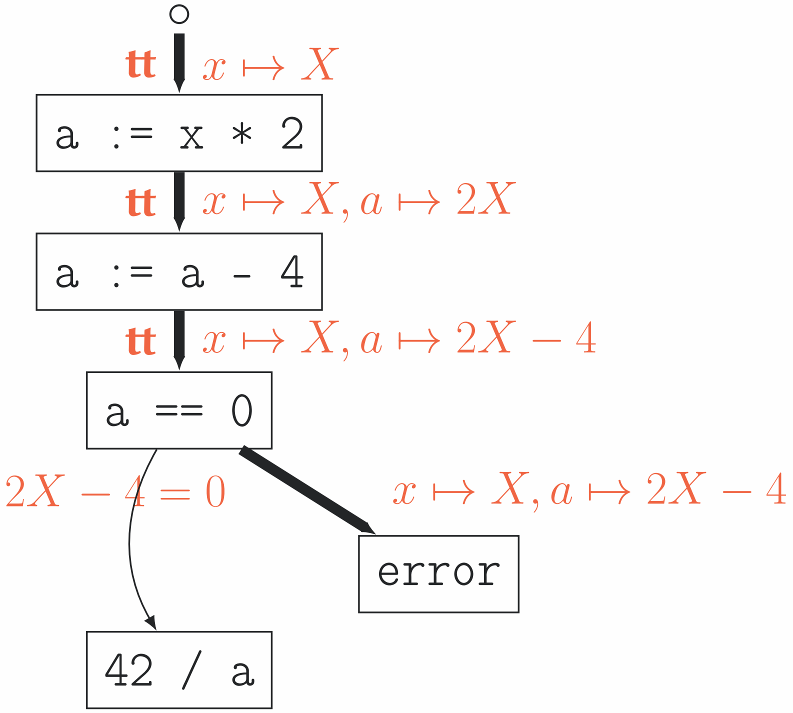
```
int f(int x) {
  int a = x * 2;
  a = a - 4;
  if (a == 0)
    error("Div by zero!");
  return 42 / a;
}
```


$x \mapsto X$
a := x * 2
$x \mapsto X, a \mapsto 2X$
a := a - 4
$x \mapsto X, a \mapsto 2X - 4$
a == 0
$x \mapsto X, a \mapsto 2X - 4$
error
42 / a

Path Conditions

Path condition is a condition on the input symbols such that if a path is feasible its path-condition is satisfiable.

```
int f(int x) {
  int a = x * 2;
  a = a - 4;
  if (a == 0)
    error("Div by zero!");
  return 42 / a;
}
```


tt
$x \mapsto X$
a := x * 2
tt
$x \mapsto X, a \mapsto 2X$
a := a - 4
tt
$x \mapsto X, a \mapsto 2X - 4$
a == 0
$2X - 4 = 0$
$x \mapsto X, a \mapsto 2X - 4$
error
42 / a

Constraint Solver

A constraint solver is a tool that finds satisfying assignments for a constraint, if it is satisfiable.

A solution of the constraint is a set of assignments, one for each free variable that makes the constraint satisfiable.

Constraint:

$$x \mapsto X, \; a \mapsto 2X - 4$$
$$2X - 4 = 0$$

Solution:

$$X = 2$$

Chapter #2:

In Practice

SAT Solvers

SAT solver is a computer program which aims to solve the Boolean satisfiability problem: whether the variables of a given Boolean formula can be consistently replaced by the values TRUE or FALSE in such a way that the formula evaluates to TRUE.

Examples:

$$\begin{aligned} a \wedge b &\rightarrow \ldots \\ a \wedge b \wedge \neg a &\rightarrow \ldots \\ a \vee b \vee \neg a &\rightarrow \ldots \\ a \wedge (\mathbf{ff} \vee \mathbf{tt}) &\rightarrow \ldots \end{aligned}$$

All expressions are in Boolean logic.

SMT Solvers

SMT solver is a computer program which aims to solve the satisfiability modulo theories: determine whether a mathematical formula is satisfiable.

Examples:

$$\begin{aligned} a < 5 \wedge a > 3 &\rightarrow \ldots \\ a < 5 \wedge f(a) > 42 &\rightarrow \ldots \\ a < 5 \vee a > 10 \vee \neg a &\rightarrow \ldots \\ a \wedge \mathbf{ff} \wedge x = 7 &\rightarrow \ldots \end{aligned}$$

SMT solvers: Z3, cvc5, Yices, and many more...

Chapter #3:

Concolic Execution

•••

...