Formal Methods for Java

Lecture 23: Software Modelchecking

Jochen Hoenicke

Software Engineering
Albert-Ludwigs-University Freiburg

July 17, 2017

Bridging the Gap

JML Tools	JPF	Ultimate	ESC/Java2 KeY Jahob
Runtime Checking	Model Checking	Software Model Checking	Verification

Software Model checking

Model checking:

- Idea: exhaustively check the system
- Try all possible paths/all possible input values.
- Use search strategies to find errors fast.

Software Model Checking

- Idea: Symbolic Abstract states
- Use model checking on abstraction
- Analyse error paths to improve abstraction

Abstractions (1)

An abstraction is a symbolic representation of multiple states.
Examples:

- Variable abstraction: Variables x,y,z, abstract y:

  Abstract State Concrete States

$$\boxed{\begin{array}{c} x:1 \\ z:4 \end{array}} \quad = \quad \cdots \boxed{\begin{array}{c} x:1 \\ y:-1 \\ z:4 \end{array}} \quad \boxed{\begin{array}{c} x:1 \\ y:0 \\ z:4 \end{array}} \quad \boxed{\begin{array}{c} x:1 \\ y:1 \\ z:4 \end{array}} \cdots$$

- Sign abstraction: Variables x,y,z:

  Abstract State Concrete States

$$\boxed{\begin{array}{c} x:+ \\ y:0 \\ z:- \end{array}} \quad = \quad \boxed{\begin{array}{c} x:1 \\ y:0 \\ z:-1 \end{array}} \quad \boxed{\begin{array}{c} x:2 \\ y:0 \\ z:-1 \end{array}} \quad \boxed{\begin{array}{c} x:1 \\ y:0 \\ z:-2 \end{array}} \cdots$$

Abstractions (2)

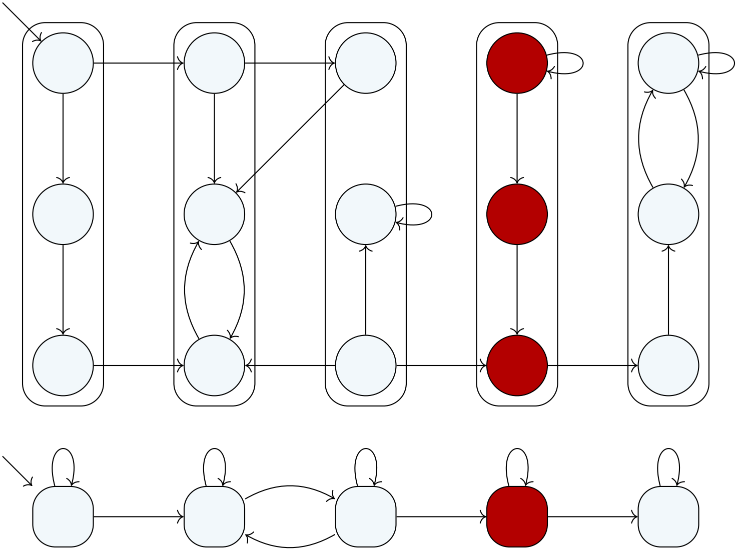
More Examples:

- Predicate abstraction: Variables x,y,z, Predicates $x < y$, $z < 0$:

  Abstract State   Concrete States

$$\boxed{\begin{array}{c} x < y \\ z \geq 0 \end{array}} \quad = \quad \cdots \boxed{\begin{array}{c} x : 0 \\ y : 1 \\ z : 0 \end{array}} \quad \boxed{\begin{array}{c} x : 1 \\ y : 3 \\ z : 0 \end{array}} \quad \boxed{\begin{array}{c} x : 0 \\ y : 1 \\ z : 2 \end{array}} \cdots$$

(Abstract) Model Checking

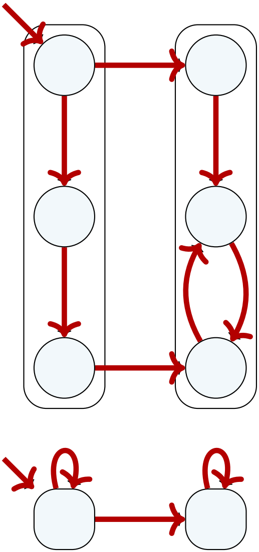


Overapproximation

There is an edge between abstract states

iff

There is an edge between two corresponding concrete states



Model Checking with Overapproximation

Check reachability of error states.

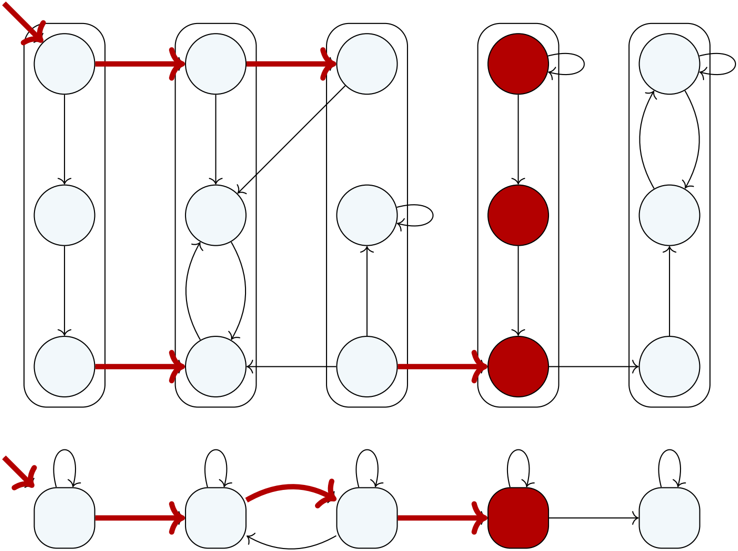
A system is correct if the error state is not reachable.

Theorem

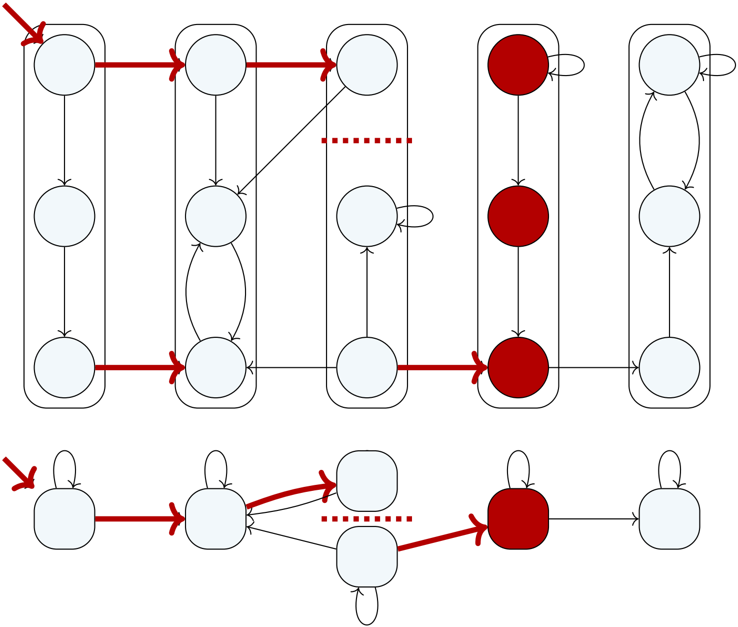
If the abstract system is correct, the concrete system is correct.

The reverse is not true, though.

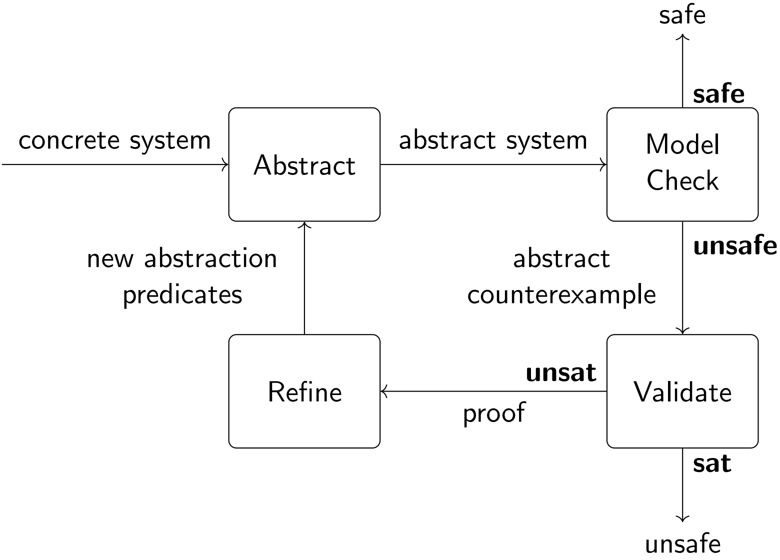
Spurious Counter Example



Refinement of Counter Example



Counterexample Guided Abstraction Refinement (CEGAR)



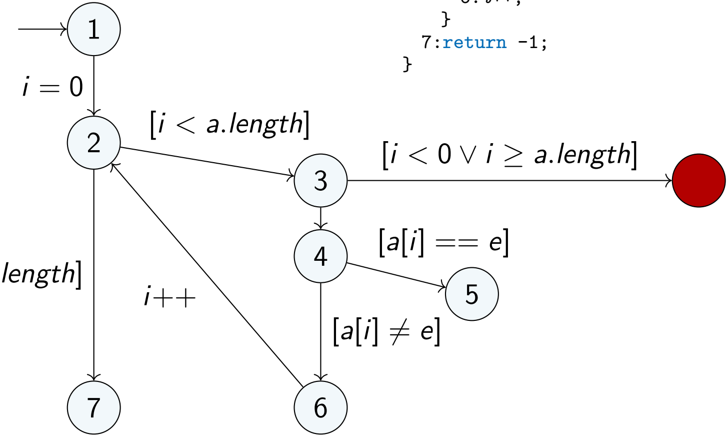
Model-checking Java Programs

```
public int find(int[] a, int e) {
    for (int i = 0; i < a.length; i++) {
        if (a[i] == e) {
            return i;
        }
    }
    return -1;
}
```

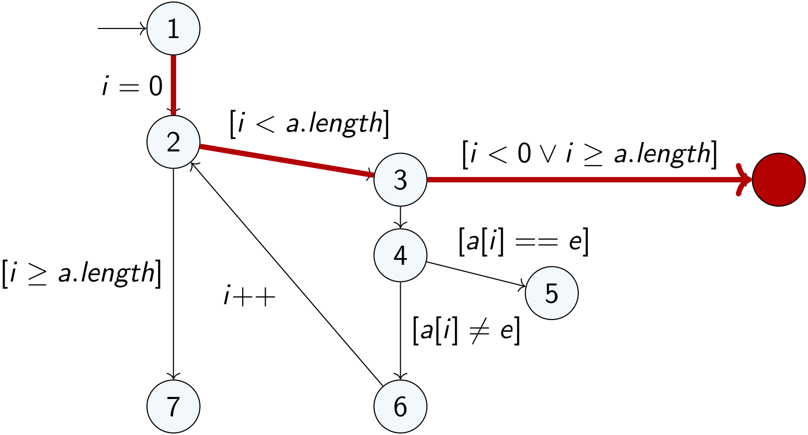
Is there an *ArrayOutOfBoundsException*?

Program Counter

```
public int find(int[] a, int e) {
  1:int i = 0;
  2:while (i < a.length) {
      3://@assert(i >= 0 && i < a.length);
      4:if (a[i] == e) {
          5:return i;
        }
      6:i++;
    }
  7:return -1;
}
```

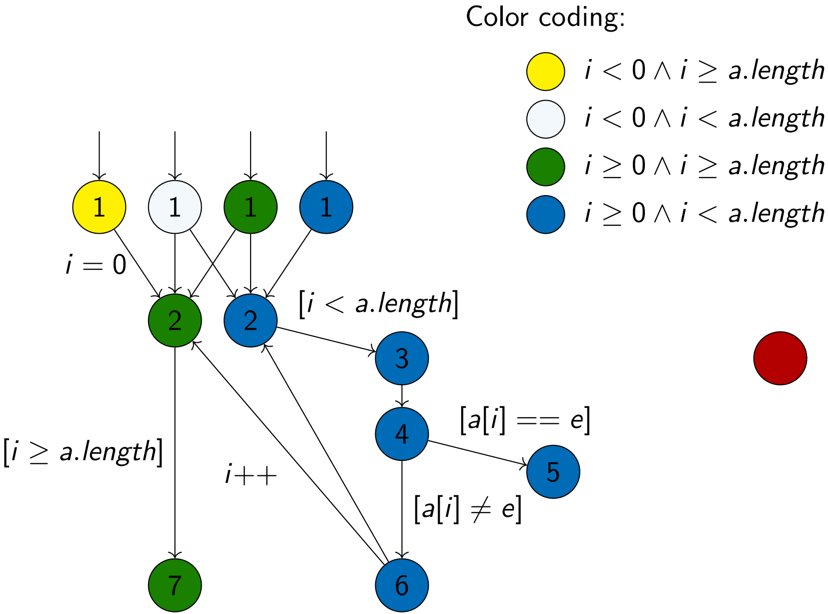


Error Path



$\Rightarrow$ we need to reason about $i < 0$ and $i < a.length$.

Predicate Abstraction



Color coding:

- 🟡 $i < 0 \land i \geq a.length$
- ⚪ $i < 0 \land i < a.length$
- 🟢 $i \geq 0 \land i \geq a.length$
- 🔵 $i \geq 0 \land i < a.length$