JBMC: A Bounded Model Checking Tool for Verifying Java Bytecode

Lucas Cordeiro
Daniel Kroening
Peter Schrammel

The University of Manchester
University of Oxford
University of Sussex
diffblue
AI for Code

Toolympics / SV-COMP 2019

JBMC

Java and JVM languages:

- Most widely used
- Established software development culture

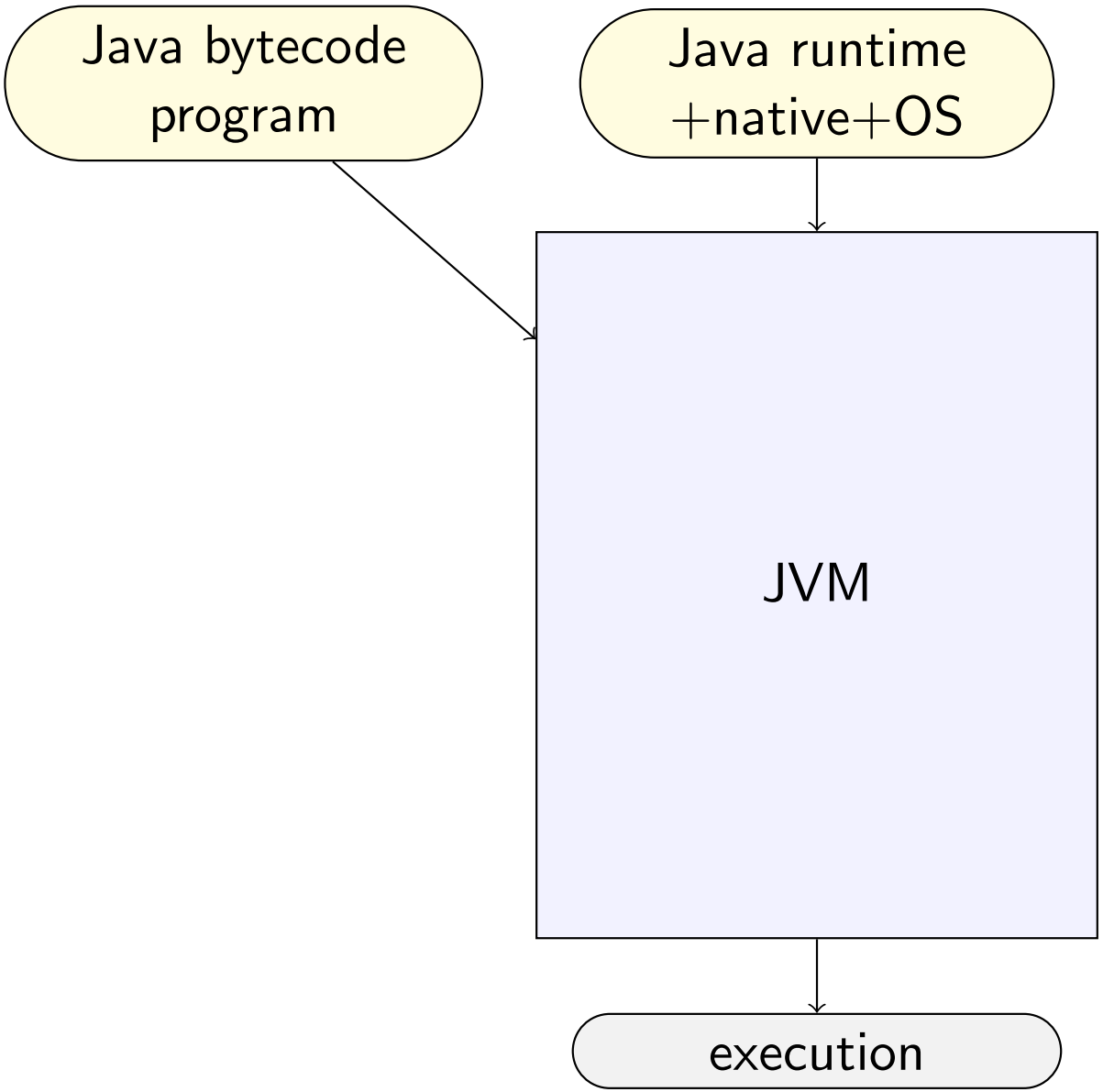
Only few model checking tools available:

- Symbolic JPF (Anand et al, TACAS'07)
- JayHorn (Kahsai et al, CAV'16)

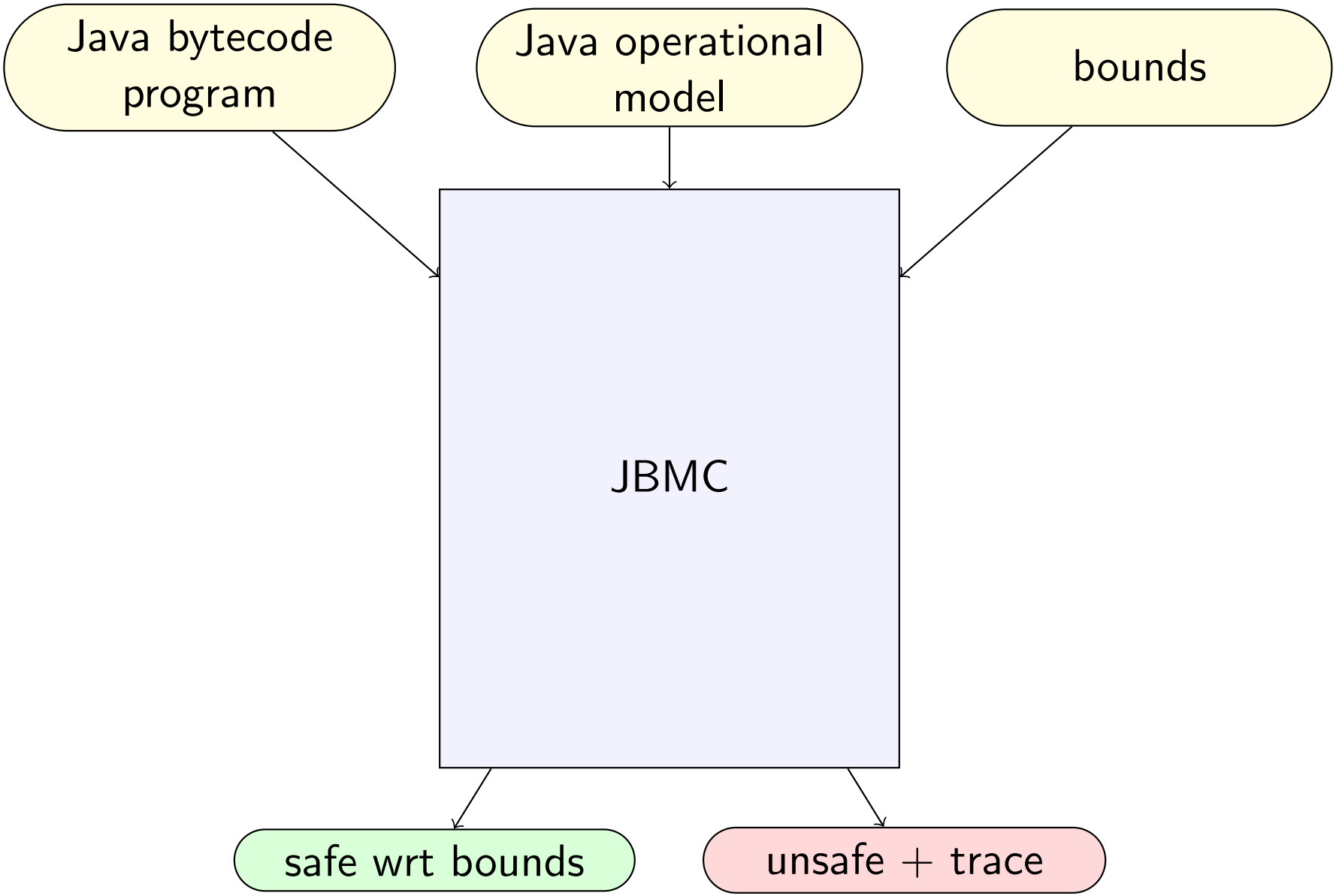
Many applications of BMC:

- Bug finding
- Program synthesis
- Test generation
- ...

JVM vs JBMC

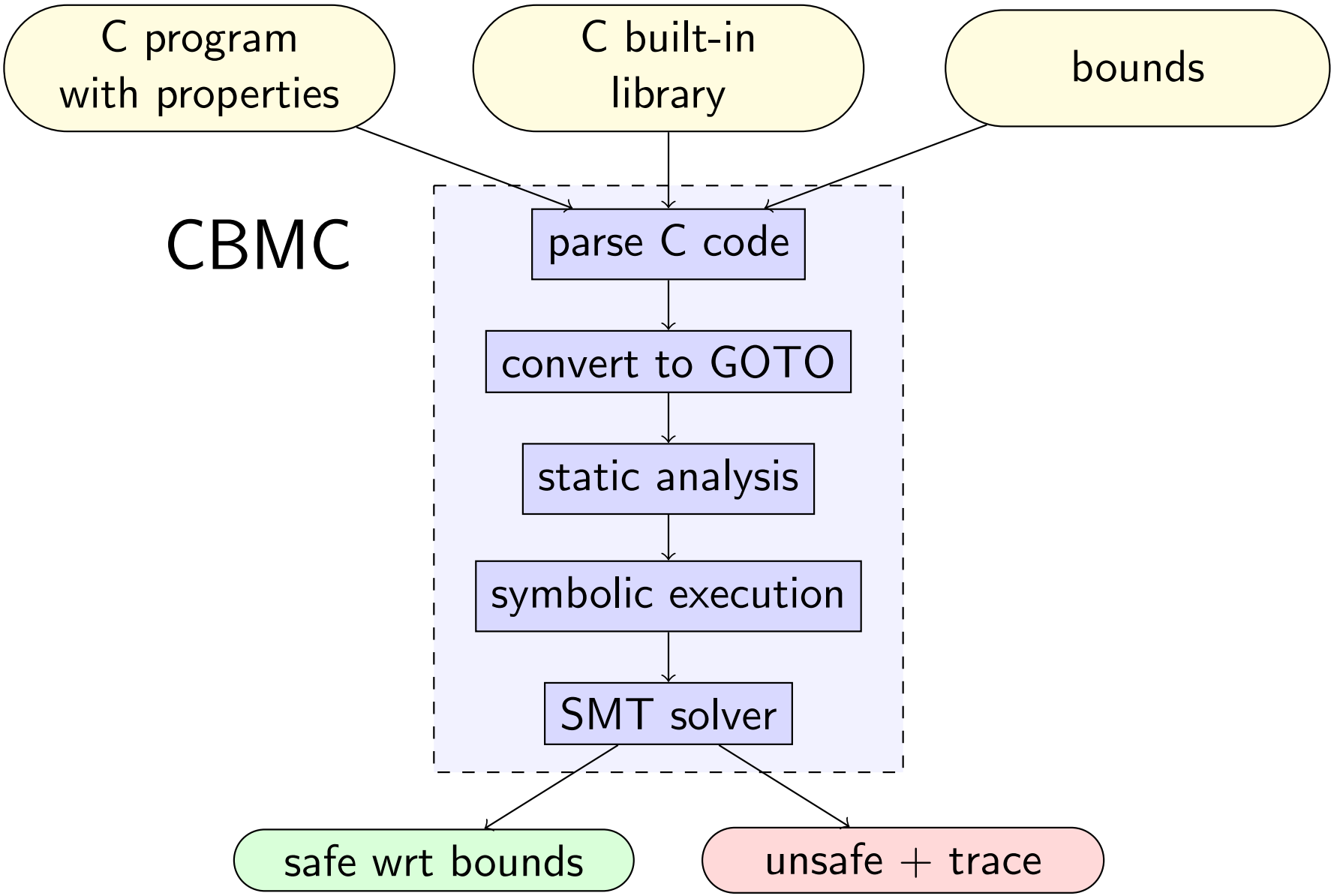

Java bytecode program
Java runtime +native+OS
JVM
execution

JVM vs JBMC


Java bytecode program
Java operational model
bounds
JBMC
safe wrt bounds
unsafe + trace

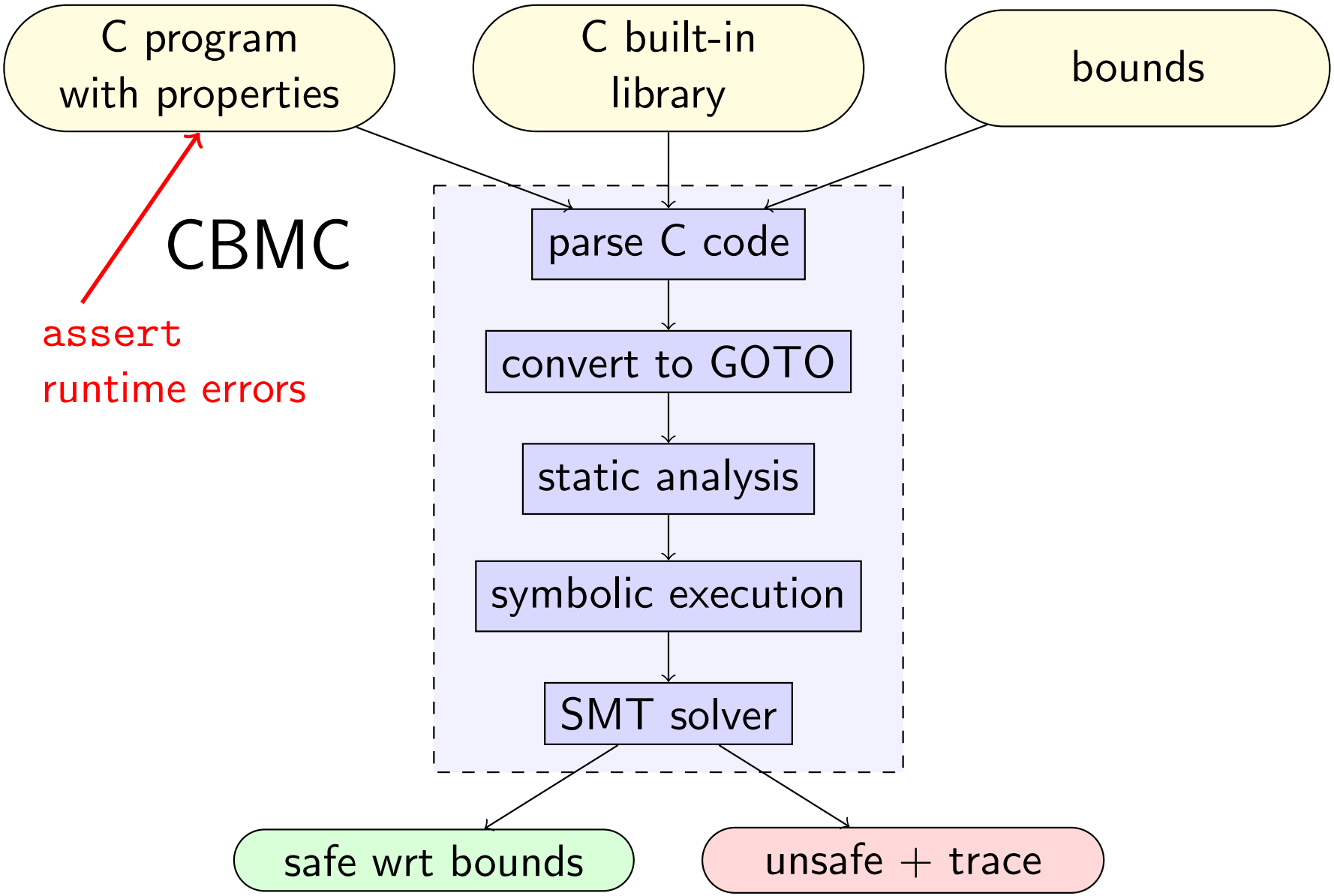
CBMC

Clarke, Kroening & Lerda, TACAS'04



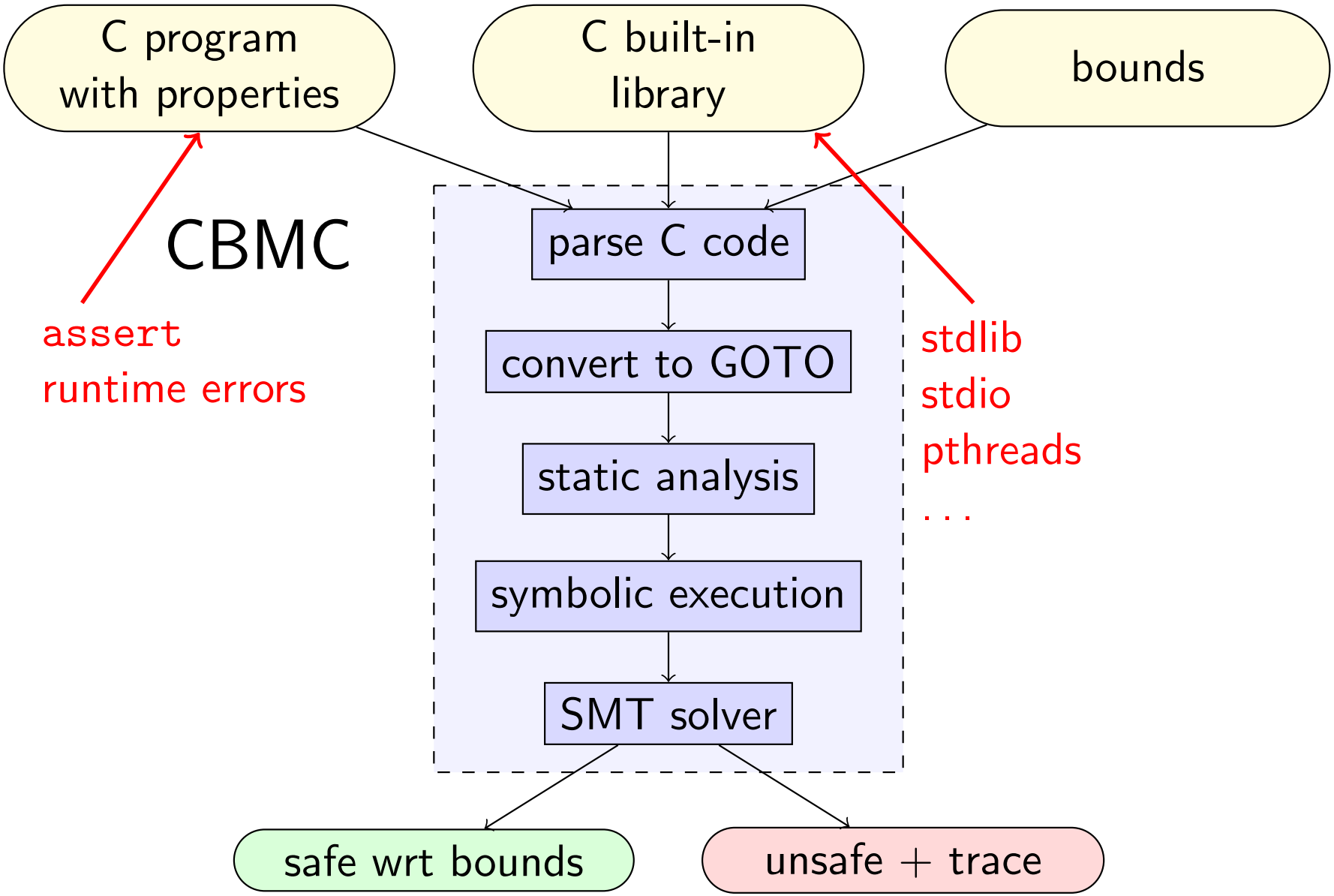
CBMC

Clarke, Kroening & Lerda, TACAS'04



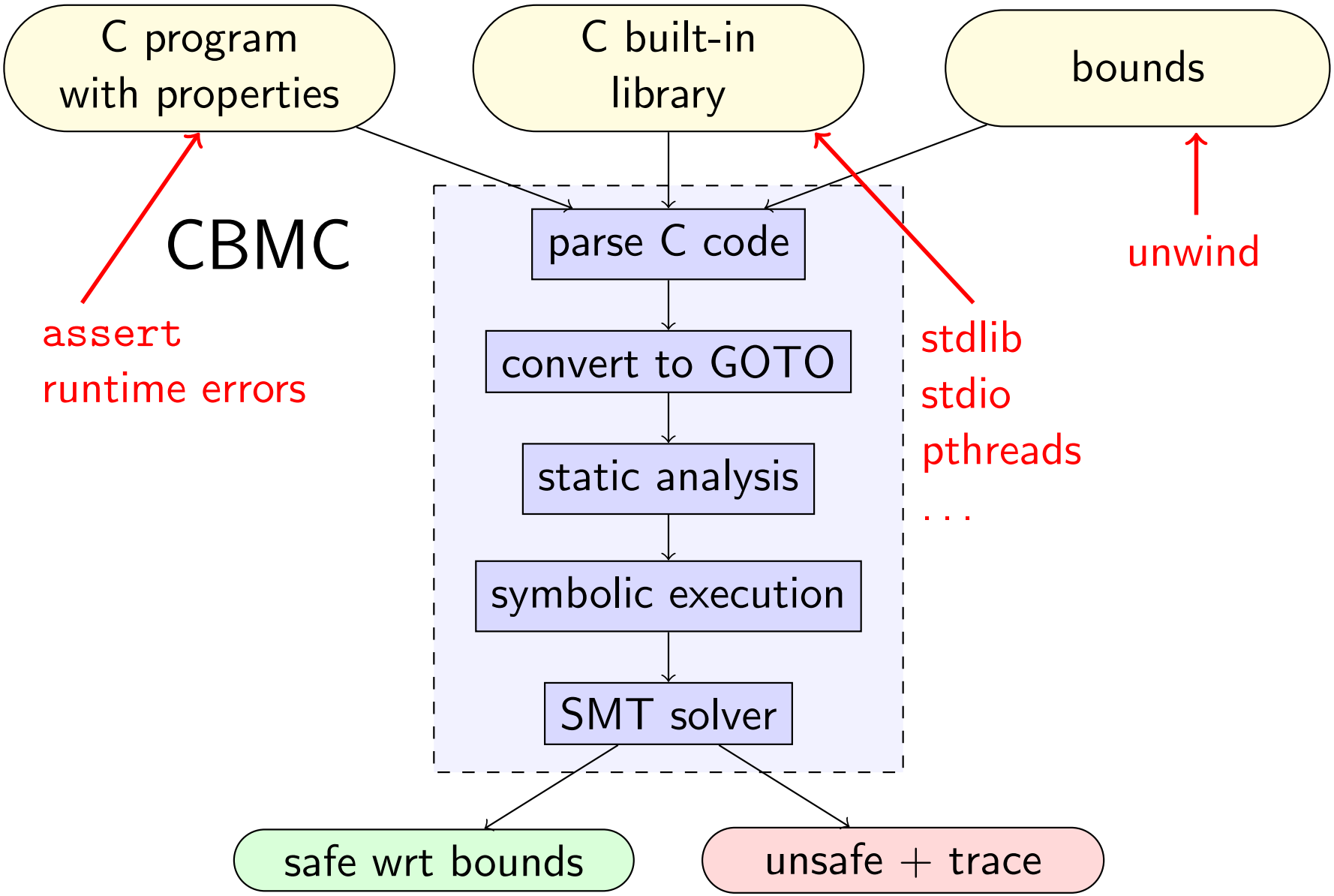
CBMC

Clarke, Kroening & Lerda, TACAS'04



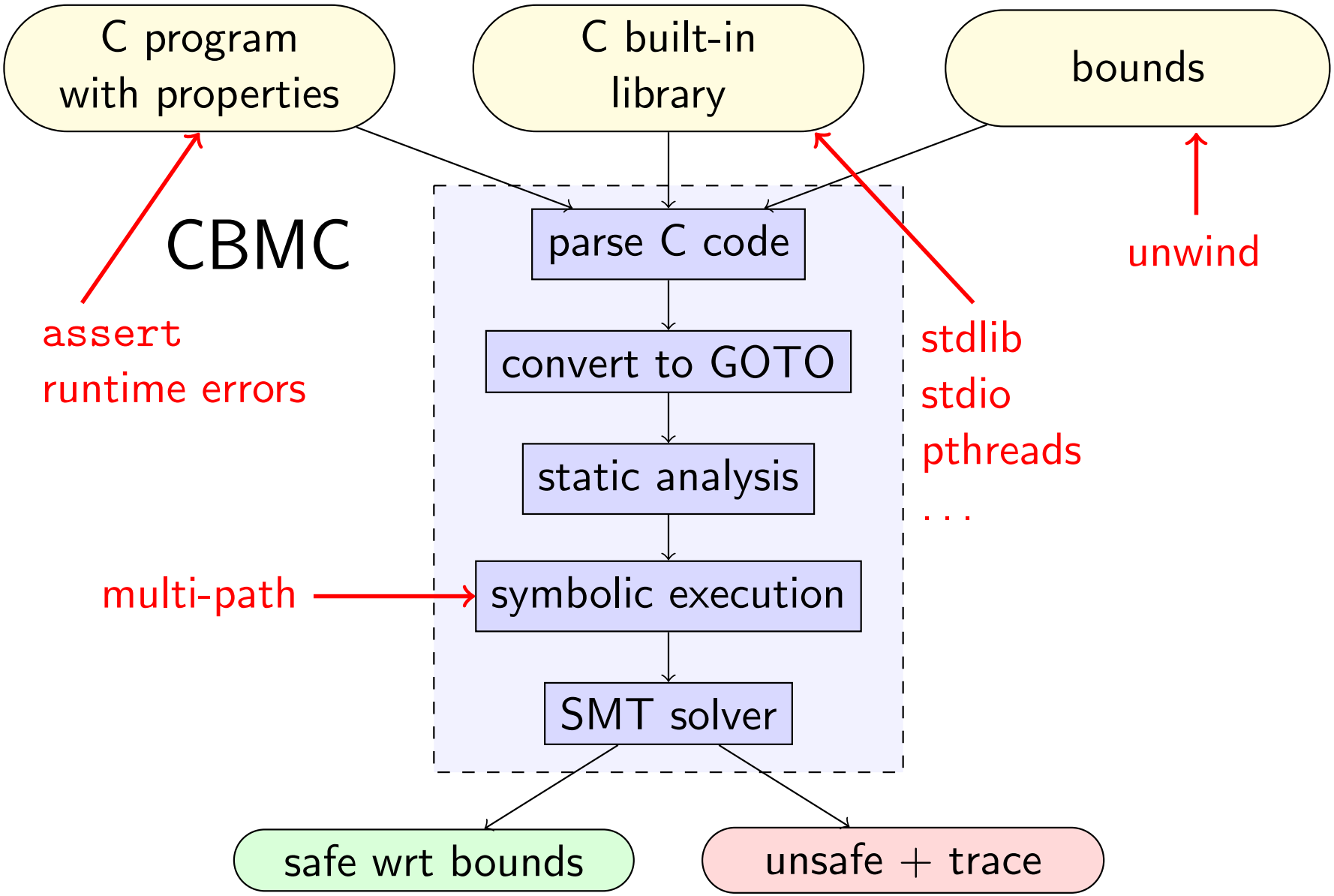
CBMC

Clarke, Kroening & Lerda, TACAS'04



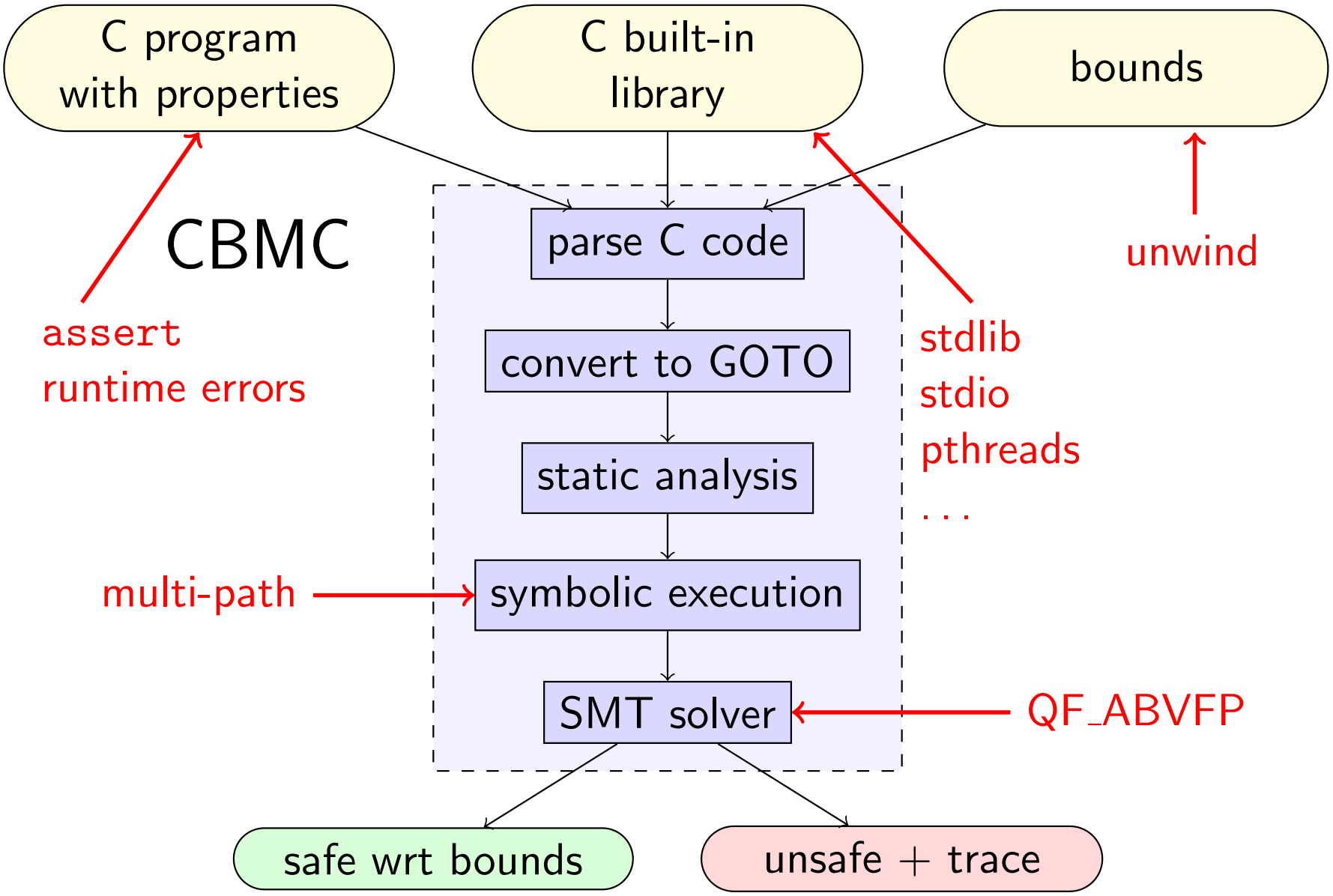
CBMC

Clarke, Kroening & Lerda, TACAS'04

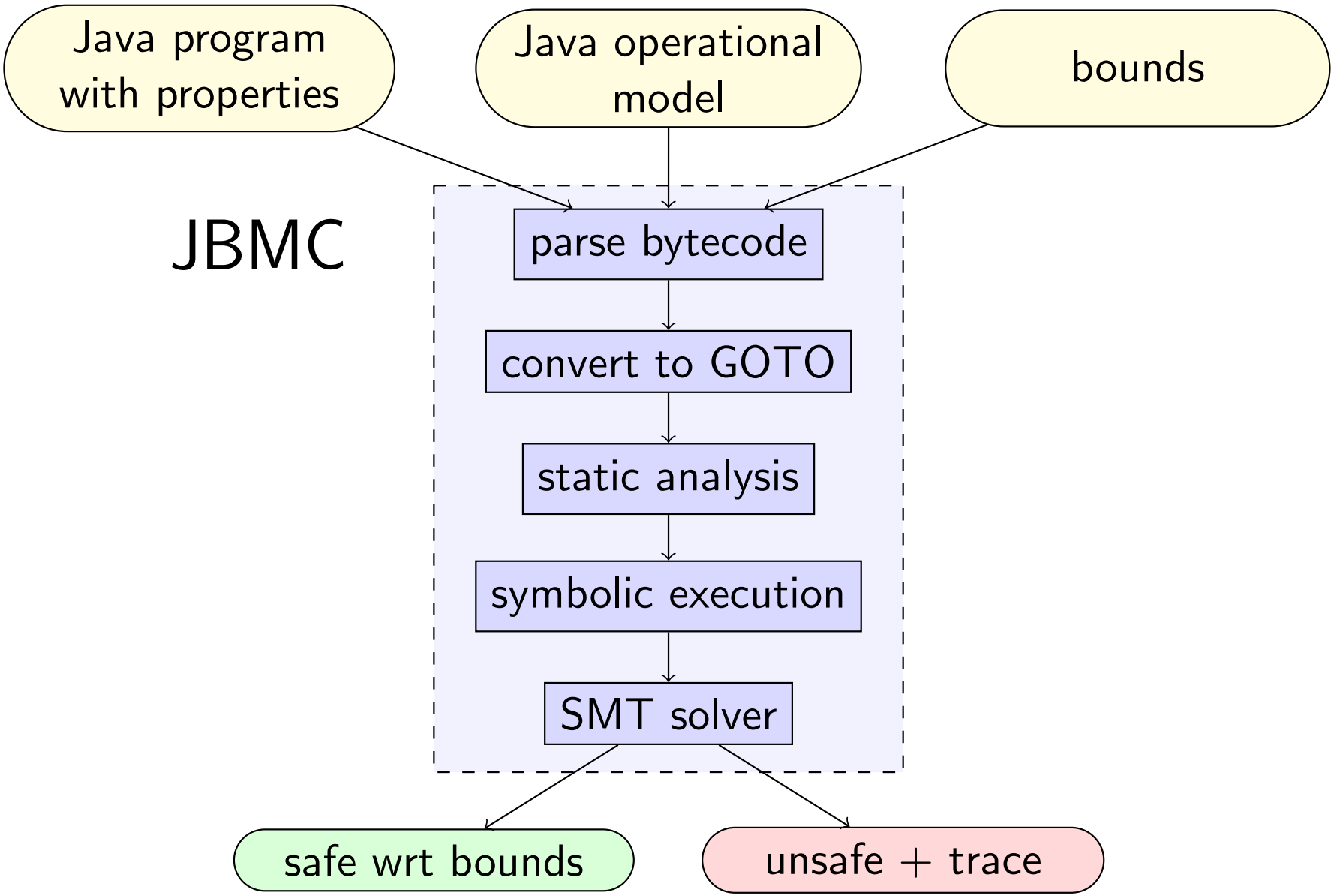


CBMC

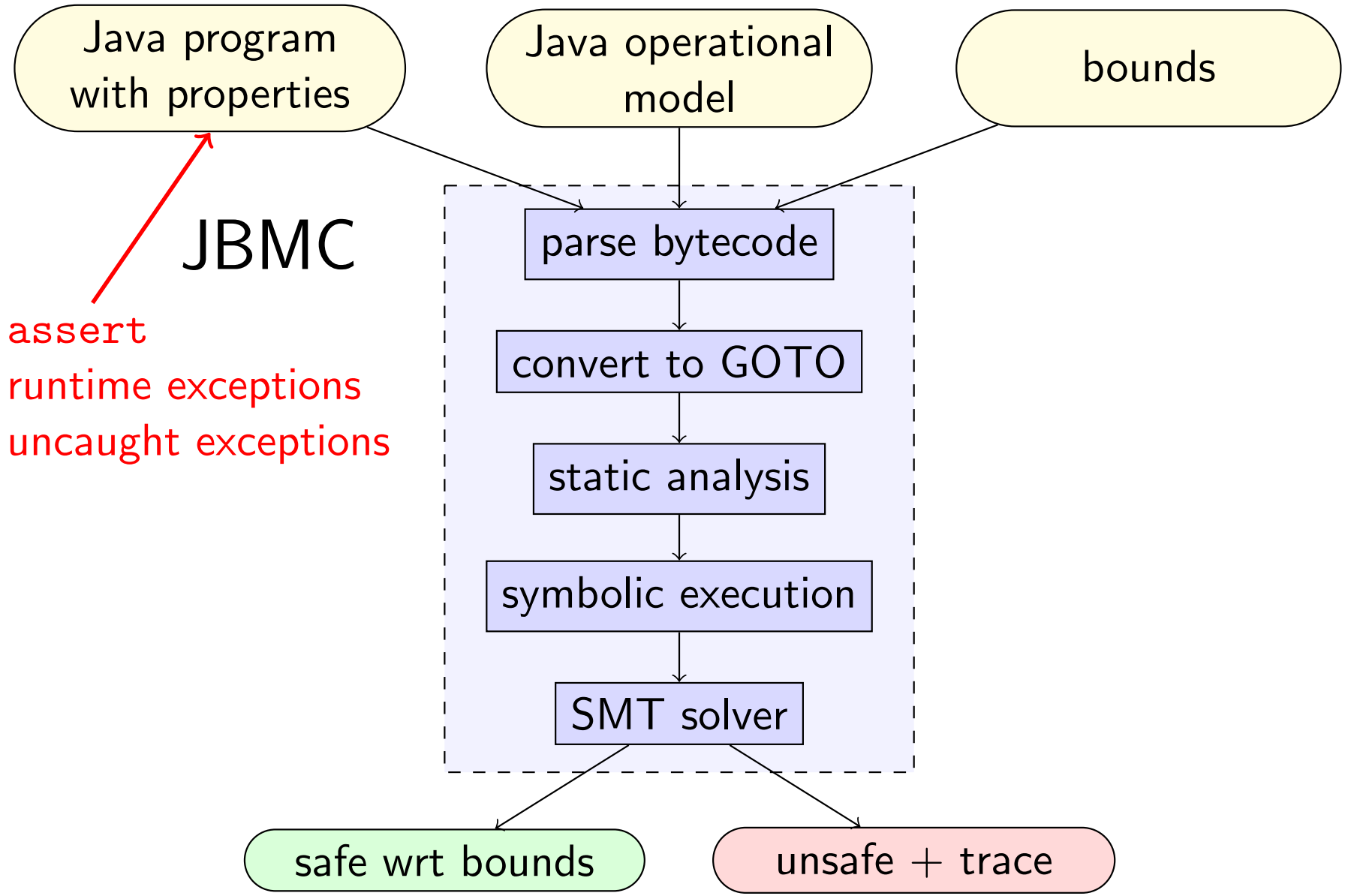
Clarke, Kroening & Lerda, TACAS'04



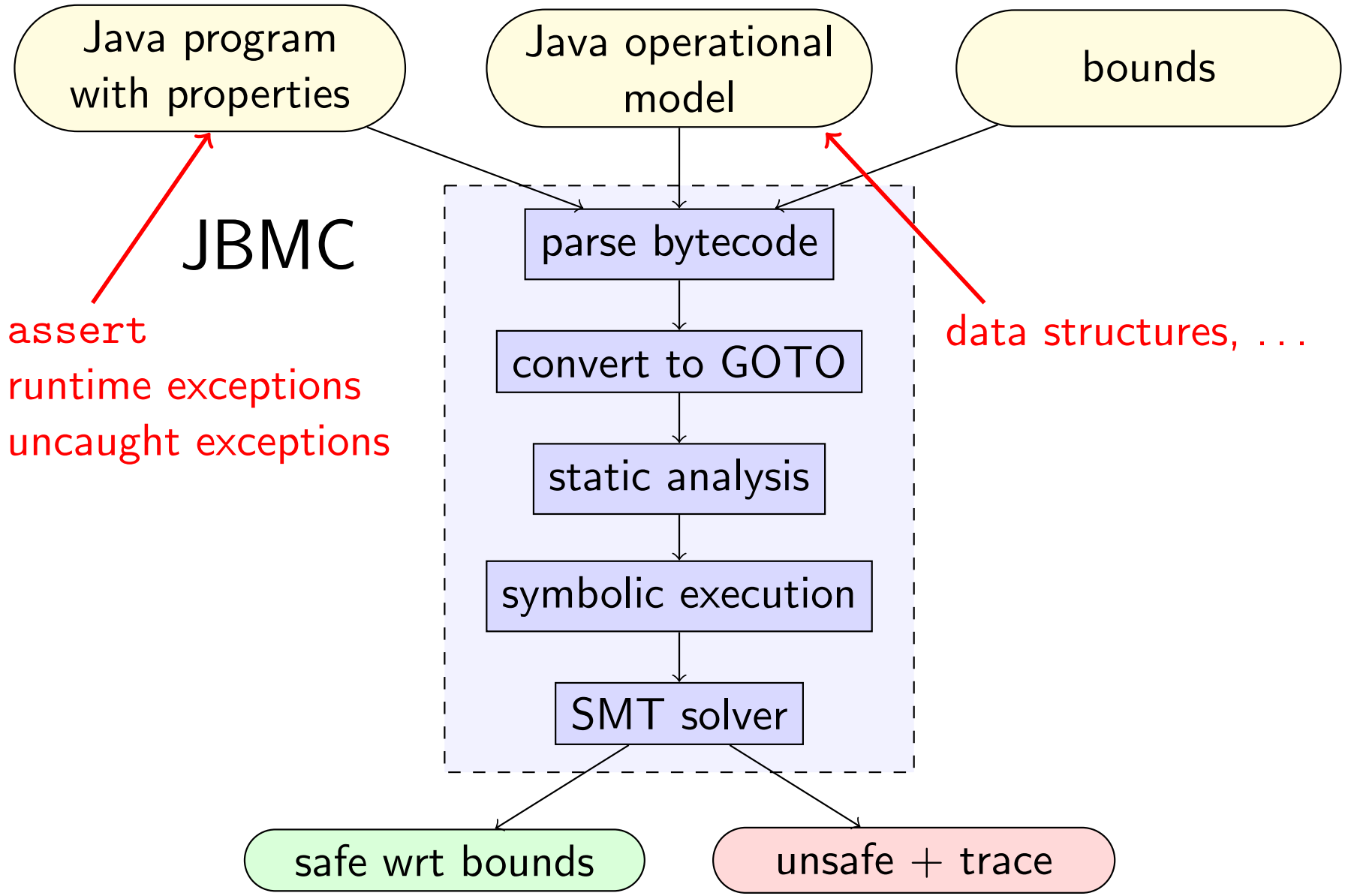
JBMC


Java program with properties
Java operational model
bounds
JBMC
parse bytecode
convert to GOTO
static analysis
symbolic execution
SMT solver
safe wrt bounds
unsafe + trace

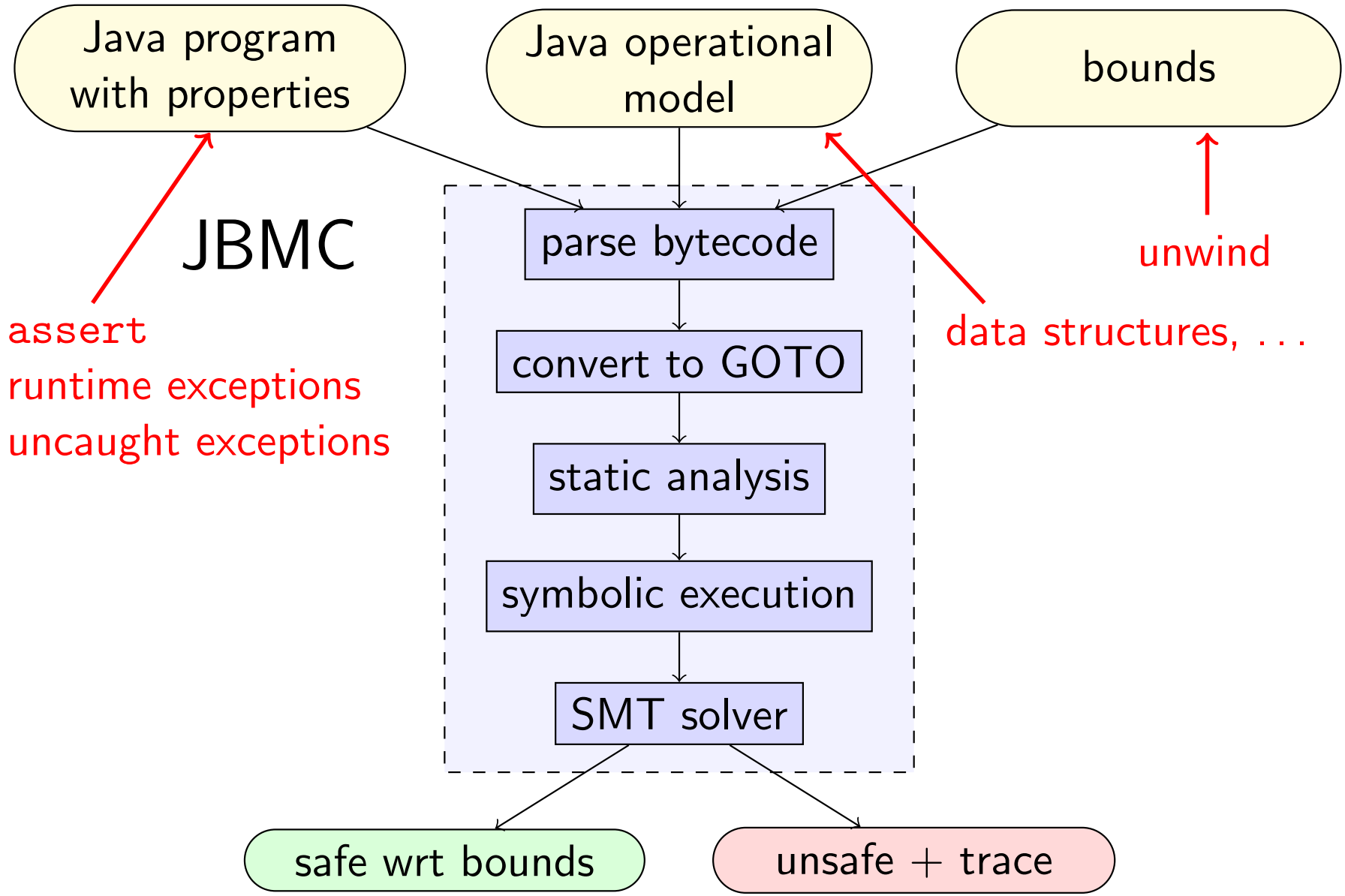
JBMC


Java program with properties
Java operational model
bounds
JBMC
assert
runtime exceptions
uncaught exceptions
parse bytecode
convert to GOTO
static analysis
symbolic execution
SMT solver
safe wrt bounds
unsafe + trace

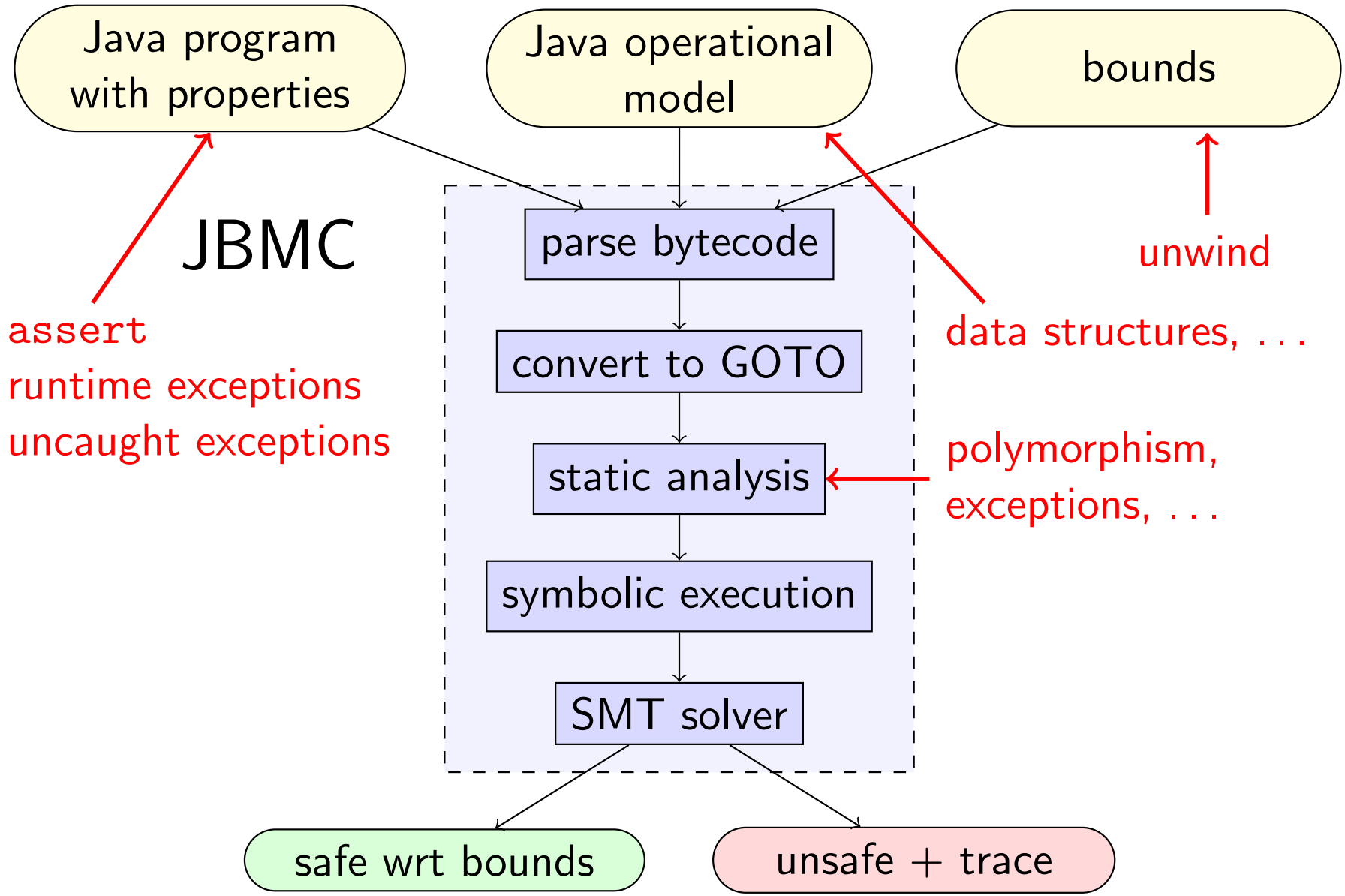
JBMC


Java program with properties
Java operational model
bounds
JBMC
parse bytecode
convert to GOTO
static analysis
symbolic execution
SMT solver
assert
runtime exceptions
uncaught exceptions
data structures, . . .
safe wrt bounds
unsafe + trace

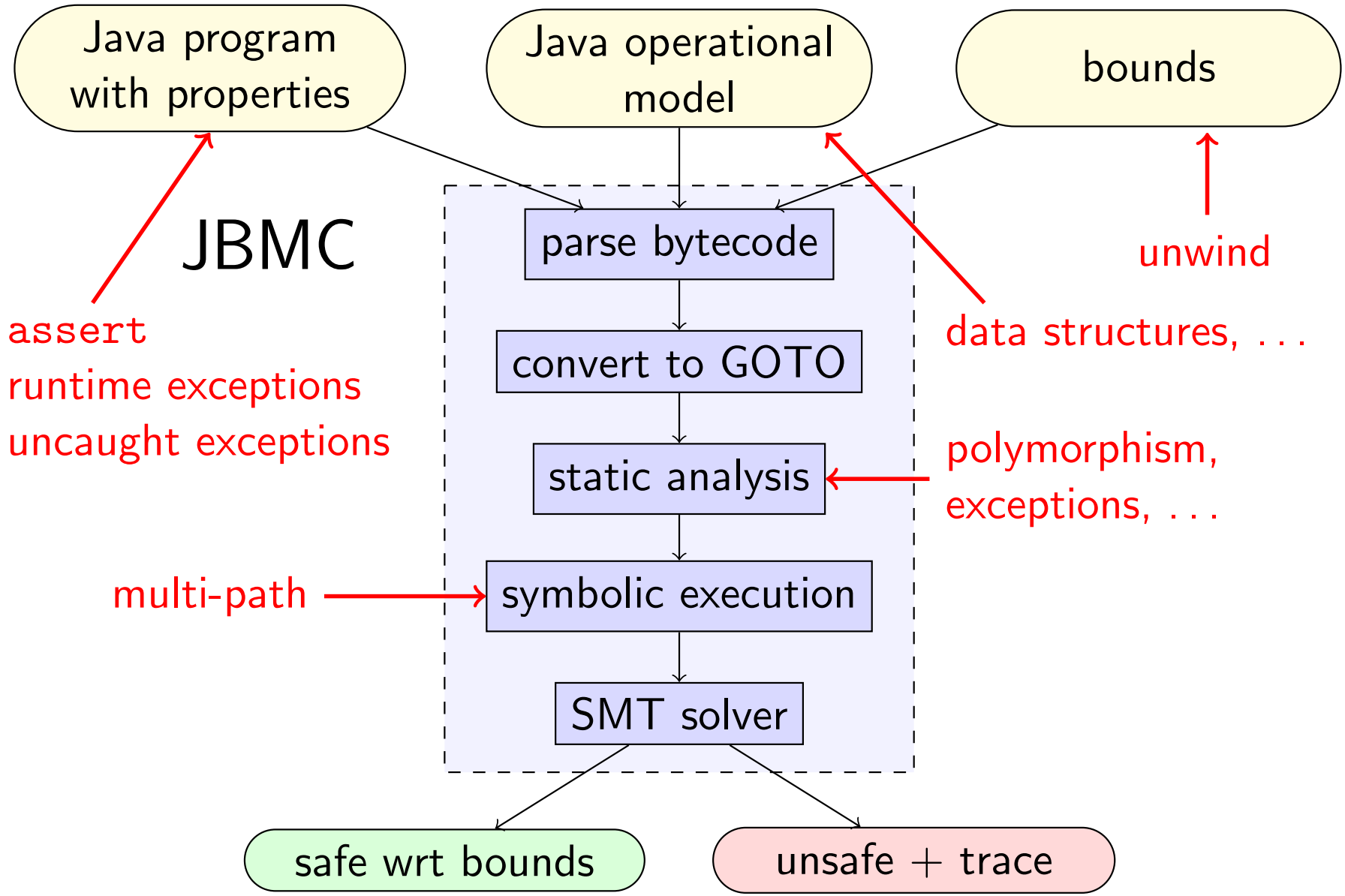
JBMC


Java program with properties
Java operational model
bounds
JBMC
parse bytecode
convert to GOTO
static analysis
symbolic execution
SMT solver
unwind
assert
runtime exceptions
uncaught exceptions
data structures, . . .
safe wrt bounds
unsafe + trace

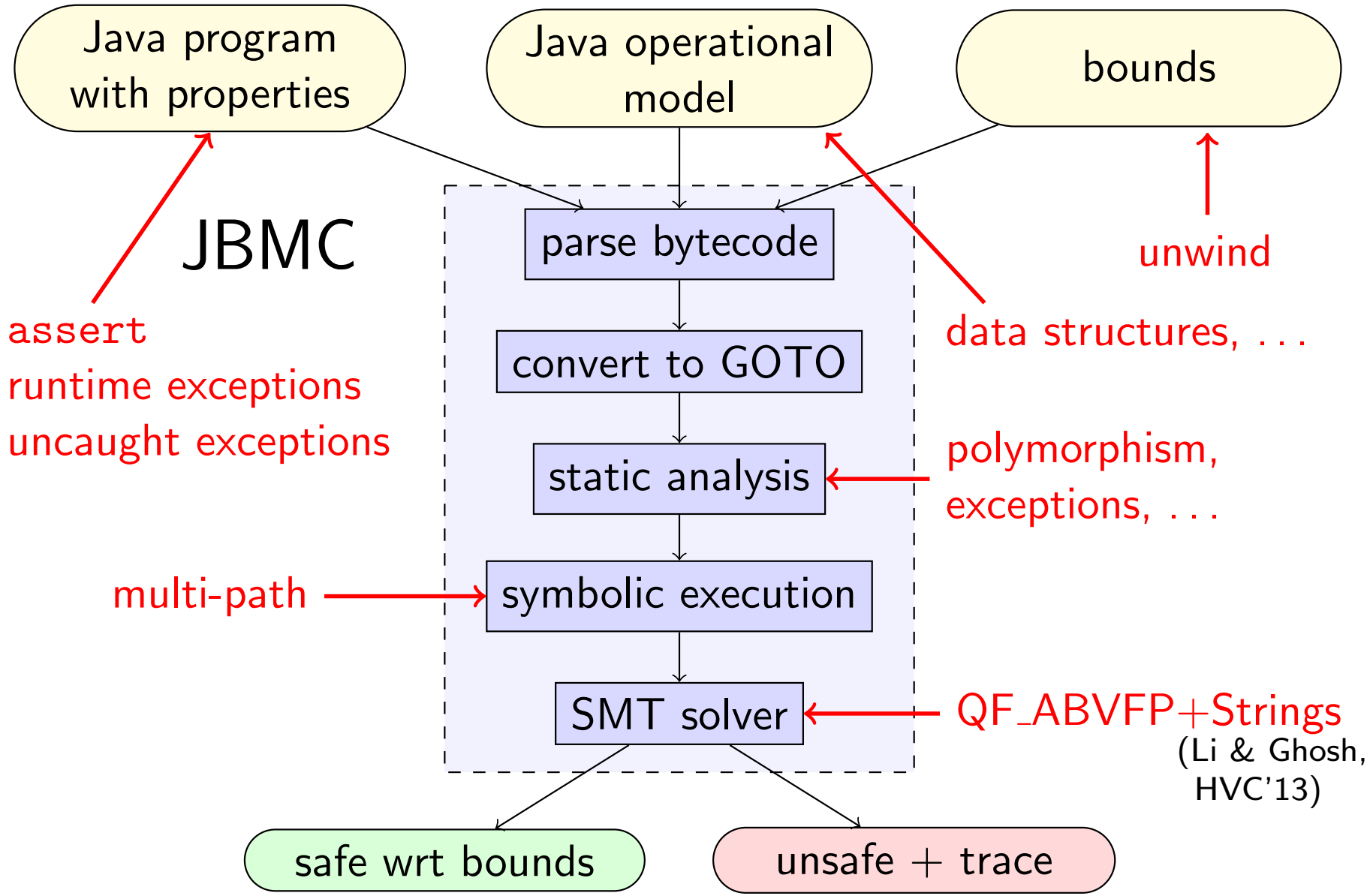
JBMC


Java program with properties
Java operational model
bounds
JBMC
parse bytecode
convert to GOTO
static analysis
symbolic execution
SMT solver
unwind
assert
runtime exceptions
uncaught exceptions
data structures, . . .
polymorphism, exceptions, . . .
safe wrt bounds
unsafe + trace

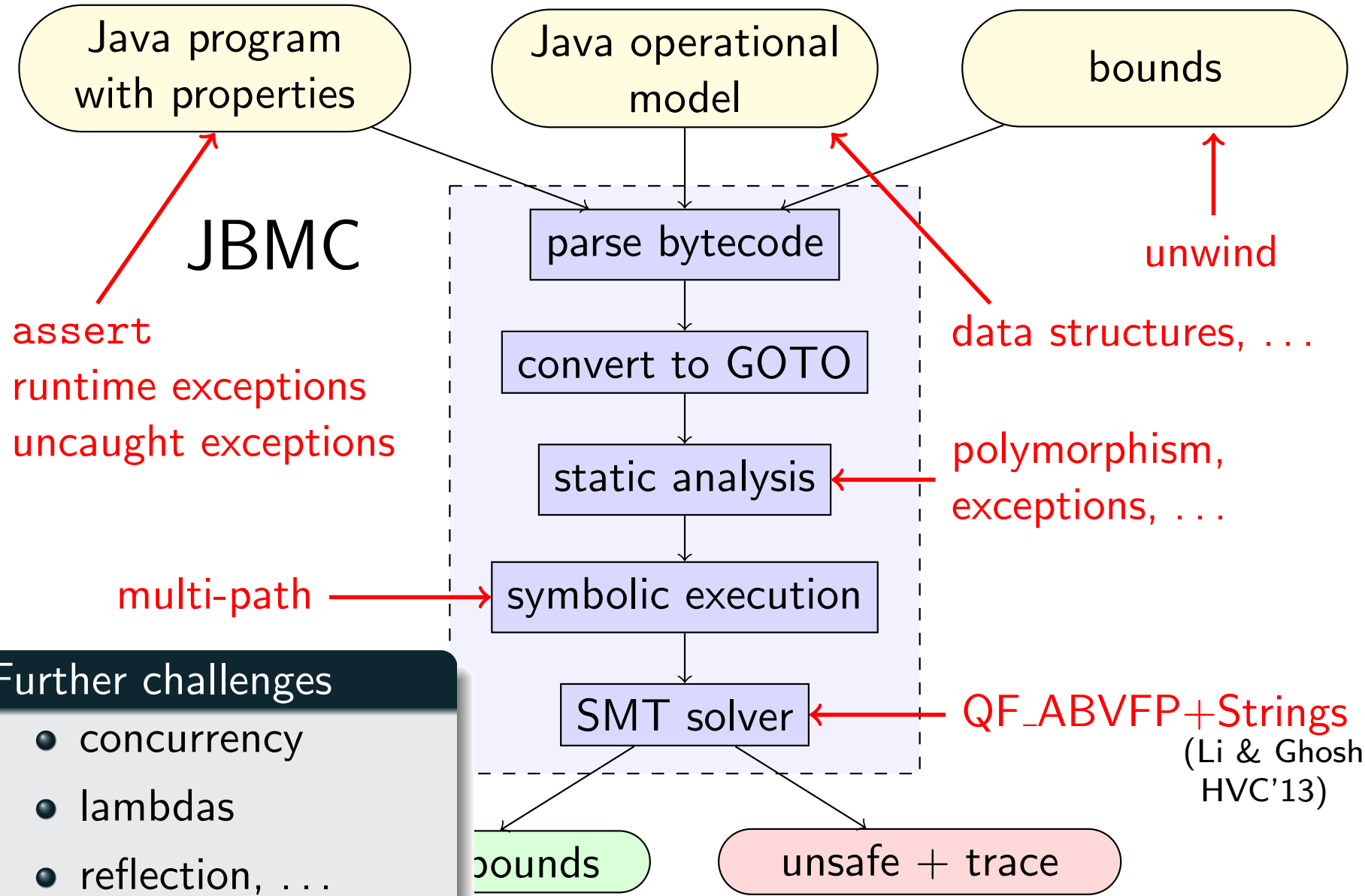
JBMC


Java program with properties
Java operational model
bounds
JBMC
parse bytecode
convert to GOTO
static analysis
symbolic execution
SMT solver
unwind
assert
runtime exceptions
uncaught exceptions
data structures, . . .
polymorphism, exceptions, . . .
multi-path
safe wrt bounds
unsafe + trace

JBMC


Java program with properties
Java operational model
bounds
JBMC
parse bytecode
convert to GOTO
static analysis
symbolic execution
SMT solver
unwind
assert
runtime exceptions
uncaught exceptions
data structures, . . .
polymorphism, exceptions, . . .
multi-path
QF_ABVFP+Strings
(Li & Ghosh, HVC'13)
safe wrt bounds
unsafe + trace

JBMC



Further challenges

- concurrency
- lambdas
- reflection, . . .

JBMC in Action

```
public class MyTranslator {
  static abstract class Translator {
    abstract String translate(String text);
    static Translator build(String language) {
      if("Chinese".equals(language))
        return new ChineseTranslator();
      return null;
    }
  }
  static class ChineseTranslator extends Translator {
    String translate(String text) {
      if(text.toLowerCase().contains("welcome to oxford"))
        return "欢迎来到牛津";
      return "I don't understand";
    }
  }
  public static void main(String[] args) {
    if(args.length < 2)
      return;
    Translator translator = Translator.build(args[0]);
    if(translator == null)
      return;
    String translatedText = translator.translate(args[1].trim());
    assert(!"欢迎来到牛津".equals(translatedText));
  }
}
```

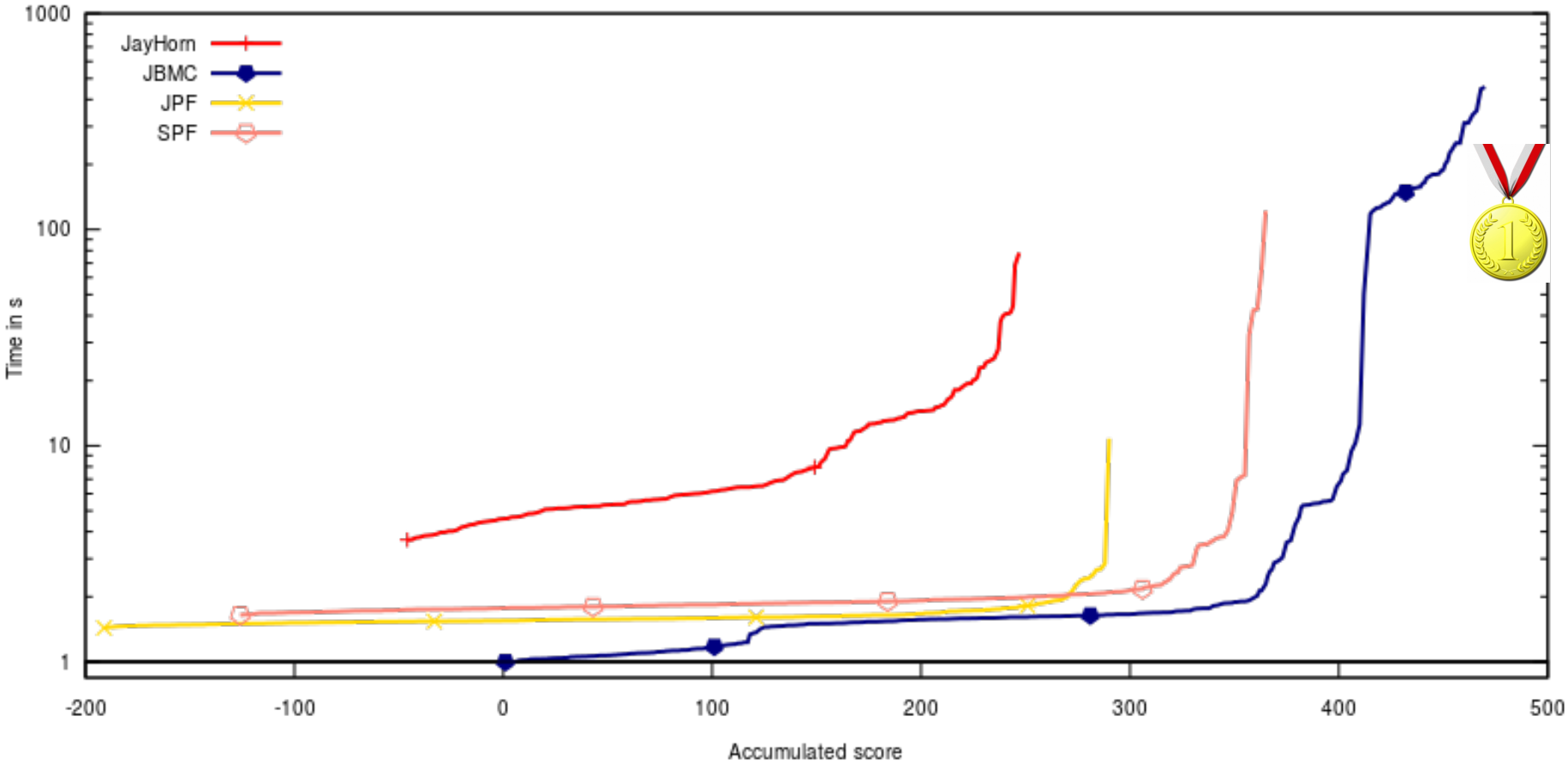
JBMC in Action

```
$ jbmc MyTranslator.class --classpath core-models.jar:. --trace
    --max-nondet-string-length 50

...
Runtime decision procedure: 0.786s
...
dynamic_object4={ 'C', 'h', 'i', 'n', 'e', 's', 'e' } // args[0].data
...
dynamic_object6={ '\u0010', '\u0017', 'w', 'e',
                  'w', 'E', 'L', 'C', 'O', 'M', 'E', ' ',
                  'T', 'o', ' ',
                  'o', 'x', 'f', 'o', 'r', 'D',
                  'x', 'x', 'x', 'x', 'x', 'x', 'x', 'x', 'x', 'x', 'x',
                  'x', ' ', ' ' }                    // args[1].data
...

Violated property:
  assertion at file MyTranslator.java line 31 function MyTranslator.main
```

SV-COMP 2019 Results



Strengths and Weaknesses

Good at

- Arithmetic, floating point arithmetic
- Strings

Limited/no support for

- Lambdas
- Concurrency
- Reflection
- JNI

What's next

JBMC:

- Concurrency support
- Witness output

SV-COMP 2020:

- Witness checker
- More and better benchmarks

Download JBMC and contribute!

`www.cprover.org/jbmc`