Static Analysis
SFWR ENG 3S03: Software Testing

A. Marinache

Department of Computing and Software, McMaster University
Canada L8S 4L7, Hamilton, Ontario

Acknowledgments: Slides adapted from Dr.R.Paige

SE 3SO3: Reviews and Evaluation

A. Marinache

Preliminaries

Type Checking

Code Quality Analysis

Contract Verification

Advanced Notions

Quiz

Static Analysis

Objectives

- Understand what static analysis means
- Explore different kinds of static analysis
- Understand how static analysis improves our SE lives

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

A quiz about the material

`https://bit.ly/3F3XqGk`

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

Definition (Dynamic Testing)

Testing code by executing it

Definition (Static Analysis (Testing))

Testing code without executing it

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

What are some advantages of dynamic testing?

- No need for special tools
  -
- No special skills needed to start
  -
    - Lots of people have some skills already
- No source code needed
  -

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

What are some disadvantages/common issues of dynamic testing?

- Complete coverage may be hard (or impossible) to achieve
- Some failures are triggered
- In a high-level test, it may be unclear what is the low-level cause

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

Solution?

- Type Checking
- Code Quality Analysis
- Contract Verification
- Advanced Notions

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

Theorem (Gödel's First Incompleteness Theorem (1931))

Any sufficiently powerful formal system that is sound and expressive enough is incomplete

Theorem (Rice's Theorem (1953))

Any nontrivial property about the language recognized by a Turing machine is undecidable

- Every static analysis is incomplete and/or unsound and/or undecidable

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

Suppose you want to prevent bad thing X in a system.
A static checker is

- Sound if 'accepts' no system that does X (no false negatives)
- Complete if 'rejects' no system that does not do X (no false postivies)
- Decidable if 'accepts' or 'rejects' any system (i.e., terminates)

Goal:

	Type system accepts program	Type system rejects program
Program doesn't do X	True negative	**False positive**
Program does do X	**False negative**	True positive

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

What happens if we run this JavaScript code?

```
1   function onlyWorksOnNumbers(x)
    {
3     return x * 10;
    }
5
    console.log(onlyWorksOnNumbers("Hello,
        world!"));
```

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Java typing e.g.1
Java typing e.g.2
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

What if we add type annotations and a type checker?

```
function onlyWorksOnNumbers(x: Number)
{
  return x * 10;
}

console.log(onlyWorksOnNumbers("Hello,
    world!"));
```

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Java typing e.g.1
Java typing e.g.2
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

What are some other type rules (in Java)?

- Array index must be an integer
- Can't put a Double into a Float or Int
- Can't put an Animal object into a Cat variable

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Java typing e.g.1
Java typing e.g.2
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

Can Java typing catch every issue?

```
    private static void
        printStringLength(String s) {
      System.out.println("String " + s + " is "
          + s.length() + " characters long");
    }

    public static void main(String[] args) {
      printStringLength("Hello");
    }
```

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Java typing e.g.1
Java typing e.g.2
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

Can Java typing catch every issue? (cont'd)

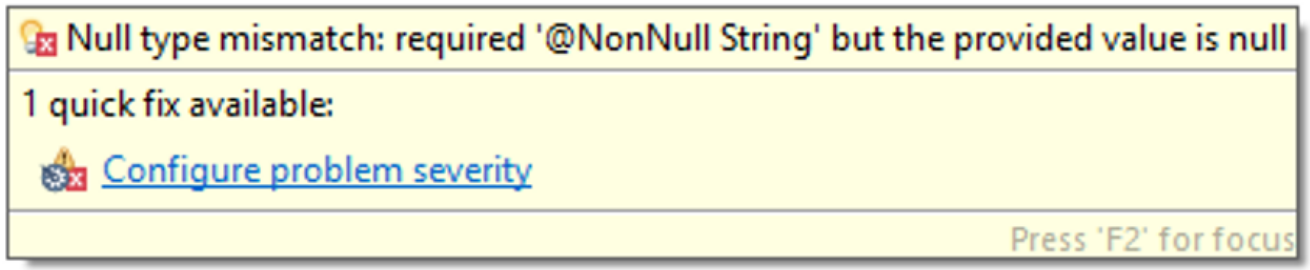
```
1  private static void printStringLength(String
       s) {
     System.out.println("String " + s + " is " +
         s.length() + " characters long");
3  }

5  public static void main(String[] args) {
     printStringLength(null);
7  }
```

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Java typing e.g.1
Java typing e.g.2
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

```
1   private static void
        printStringLength(@NonNull String s) {
      System.out.println("String " + s + " is "
          + s.length() + " characters long");
3   }

5   public static void main(String[] args) {
      printStringLength(null);
7   }
```



SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Java typing e.g.1
Java typing e.g.2
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

Can Java typing catch every issue?

```
    private static String[] strings = new
        String[] {"Zero", "One", "Two"};
2
    private static void printString(int index) {
4     System.out.println(strings[index]);
    }
6
    public static void main(String[] args) {
8     printString(1);
    }
```

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Java typing e.g.1
Java typing e.g.2
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

```
    private static String[] strings =
      new String[] {"Zero", "One", "Two"};

    private static void printString(int index) {
      System.out.println(strings[index]);
    }

    public static void main(String[] args) {
      printString(3);
    }
```

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Java typing e.g.1
Java typing e.g.2
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

```
1   private static String[] strings =
    new String[] {"Zero", "One", "Two"};
3
    private static void
        printString(@IntRange(0, 2) int index) {
5     System.out.println(strings[index]);
    }
7
    public static void main(String[] args) {
9     printString(3);
    }
```

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Java typing e.g.1
Java typing e.g.2
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

```
1   private static final String[] strings =
    new String[] {"Zero", "One", "Two"};
3
    private static void
        printString(@IntRangeOfArray(strings)
        int index) {
5     System.out.println(strings[index]);
    }
7
    public static void main(String[] args) {
9     printString(3);
    }
```

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Java typing e.g.1
Java typing e.g.2
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Java typing e.g.1
Java typing e.g.2
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

Definition (Static Type Checking)

Checking source code prior to run-time to determine whether the type rules in force (from the language or from other tools) have been respected

Do all modern languages use strong compile-time (static) type checking?

Some do

- Java
- C++
- C#
- Rust
- Go
- SPARK
- Haskell

Some don't

- Javascript
- Python
- Lua
- Objective-C

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Java typing e.g.1
Java typing e.g.2
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

When static type checking is available, do good programmers always use the strongest form of it?

- E.g. Do good Java programmers always use extended type annotations, as supported by FindBugs?

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Java typing e.g.1
Java typing e.g.2
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

Class exercise: Take about 3 minutes to discuss and answer:

- Why don't all modern languages use strong static typing?
- Why do good programmers not always use the strongest typing available for their language?
- When can strong static typing help us most?

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Java typing e.g.1
Java typing e.g.2
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

Why don't all modern languages use strong static typing?

-
-
-
-

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Java typing e.g.1
Java typing e.g.2
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

Why do good programmers not always use the strongest typing available for their language?

- All reason from previous slide plus
-
-
-

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Java typing e.g.1
Java typing e.g.2
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

Trade-offs of Strong Typing

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Java typing e.g.1
Java typing e.g.2
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

When can strong static typing help us most?

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Java typing e.g.1
Java typing e.g.2
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Java typing e.g.1
Java typing e.g.2
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Compiler Warnings
False Alarms?
CQA Tools
Contract Verification
Advanced Notions
Quiz

Definition (Code Quality Analysis (CQA))

Automated checking of source code for patterns that are often (but not always) indicators of errors.

What is the most common form of CQA?

-

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Compiler Warnings
False Alarms?
CQA Tools
Contract Verification
Advanced Notions
Quiz

```
1   private boolean a;

3   public void doSomething(boolean b) {
      if(a = b) {
5       //do something
      }
7     ...
    }
```

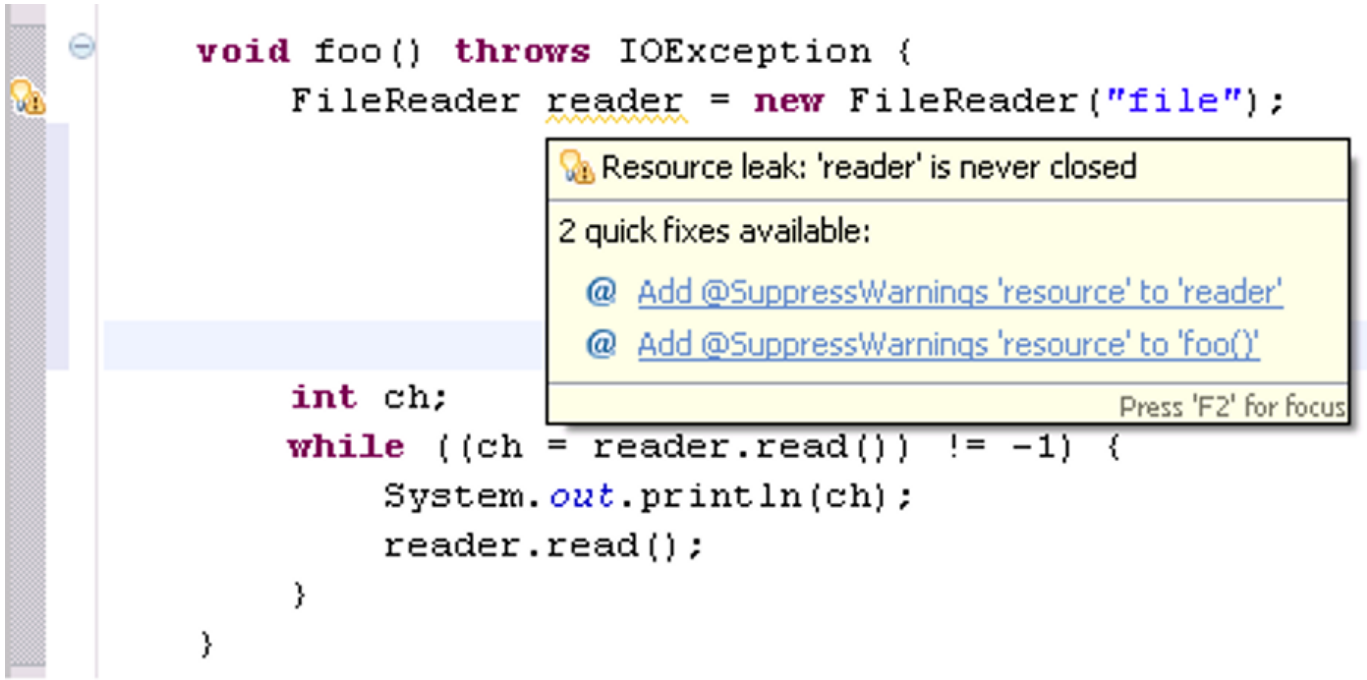
SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Compiler Warnings
False Alarms?
CQA Tools
Contract Verification
Advanced Notions
Quiz

```
1     if (launchApproved);
        launchRocket()
```

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Compiler Warnings
False Alarms?
CQA Tools
Contract Verification
Advanced Notions
Quiz

```
1    java.util.Date myDate = new
         java.util.Date();
     int currentDay = myDate.getDay();
```

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Compiler Warnings
False Alarms?
CQA Tools
Contract Verification
Advanced Notions
Quiz



Source: Eclipse online docs

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Compiler Warnings
False Alarms?
CQA Tools
Contract Verification
Advanced Notions
Quiz

```
    private boolean a;
2
    public void doSomething(boolean b) {
4     if(a = b) {
        //do something
6     }
      ...
8   }
```

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Compiler Warnings
False Alarms?
CQA Tools
Contract Verification
Advanced Notions
Quiz

```
1   private boolean a;

3   public void doSomething(boolean b) {
      a = b;
5
      if(a) {
7       //do something
      }
9     ...
    }
```

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Compiler Warnings
False Alarms?
CQA Tools
Contract Verification
Advanced Notions
Quiz

Compilers don't implement every possible warning

- They are complex enough already
- They need to have fast runtime and be predictable

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Compiler Warnings
False Alarms?
CQA Tools
Contract Verification
Advanced Notions
Quiz

- Lint (grandparent of all): `http://www.gimpel.com/html/index.htm`
- FindBugs (main open source for Java): `http://findbugs.sourceforge.net/`
- .Net Analyzers (for .Net): `https://learn.microsoft.com/en-us/dotnet/fundamentals/code-analysis/overview?utm_source=chatgpt.com&tabs=net-9`
- PyLint (for Python): `https://pylint.pycqa.org/`

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Compiler Warnings
False Alarms?
CQA Tools
Contract Verification
Advanced Notions
Quiz

FindBugs: some bad practice defects it detects `http://findbugs.sourceforge.net/bugDescriptions.html`

- CNT: Rough value of known constant found (CNT_ROUGH_CONSTANT_VALUE)
  - e.g., a literal 3.1415 was used for Pi
- EQ: Equals method fails for subtypes (EQ_GETCLASS_AND_CLASS_CONSTANT)
  - e.g., Code used: `Foo.class == o.getClass()`; It is better to check `this.getClass() == o.getClass()`

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Compiler Warnings
False Alarms?
CQA Tools
Contract Verification
Advanced Notions
Quiz

FindBugs (cont'd)

- NP: toString method may return null (NP_TOSTRING_COULD_RETURN_NULL)
  - Officially allowed, bad practice because it is not expected
- NM: Confusing method names (NM_CONFUSING)
  - e.g., The referenced methods have names that differ only by capitalization

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Compiler Warnings
False Alarms?
CQA Tools
Contract Verification
Advanced Notions
Quiz

FindBugs (cont'd)

- IL: A collection is added to itself (IL_CONTAINER_ADDED_TO_ITSELF)
  - e.g., `list.add(list.toString());`
    `Warning: Adding container to itself can cause infinite loops`
- NM: Class defines equal(Object); should it be equals(Object)? (NM_BAD_EQUAL)
  - This class defines a method `equal(Object)`. This method does not override the `equals(Object)` method in `java.lang.Object`, which is probably what was intended.

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Compiler Warnings
False Alarms?
CQA Tools
Contract Verification
Advanced Notions
Quiz

CQA tools generate a lot of output

- Existing codebase may have a lot of areas to be warned about
- Not all of them are faults
- Newer tools are getting better at discovering real warnings
  - FindBugs developers aim for at least 50% of warnings to be something the user would consider a defect
- ML-based tools to prune outputs
  - Issues: performance, proportion pruned [KAL22]

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Compiler Warnings
False Alarms?
CQA Tools
Contract Verification
Advanced Notions
Quiz

Tunning CQAs tutorials

- http://www.riverblade.co.uk/downloads/slides/taming_the_lint_monster_slides.pdf
- https://www.riverblade.co/downloads/articles/taming_the_lint_monster_pt1.pdf
- https://www.riverblade.co/downloads/articles/taming_the_lint_monster_pt2.pdf

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Compiler Warnings
False Alarms?
CQA Tools
Contract Verification
Advanced Notions
Quiz

What if, after tunning, the CQA output is still too big?

- It may be that the problem for this codebase can't be fixed
- May need to give up, and only use CQA on future codebases that you build to be clean from the start

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Compiler Warnings
False Alarms?
CQA Tools
Contract Verification
Advanced Notions
Quiz

Are CQA Tools Worthwhile?

- Used well, they can be
- Google's experiences with Findbugs
  `http://dx.doi.org/10.1145/1831708.1831738`
  - "hundreds of engineers, thousands of warnings"
  - "77% of the reviews identified the warnings as real defects"
  - . . . but none of them were associated with serious misbehaviours in their current production code
  - . . . although did find serious faults in not-yet-deployed code

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Compiler Warnings
False Alarms?
CQA Tools
Contract Verification
Advanced Notions
Quiz

Class exercise: Take about 3 minutes to discuss and answer:

- Why do most programmers not use third-party CQA tools?
- Is there a time when a CQA tool might have saved you a lot of effort and stress?
- Could you usefully apply a CQA tools to something you're working on at the moment (or will be soon)

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Compiler Warnings
False Alarms?
CQA Tools
Contract Verification
Advanced Notions
Quiz

Why do most programmers not use third-party CQA tools?

-
-
-
-

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Compiler Warnings
False Alarms?
CQA Tools
Contract Verification
Advanced Notions
Quiz

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Compiler Warnings
False Alarms?
CQA Tools
Contract Verification
Advanced Notions
Quiz

- Is type checking enough?
  - Conservative tools (sound) – some errors will not be caught
- Is type checked code cleaner?
  - Not always – downcasting to avoid warnings

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

```
    private static Stack<String> strings = new
        Stack<>();
2
    private static void populateStack() {
4     //do nothing
    }
6
    private static String popString() {
8     return strings.pop();
    }
10
    public static void main(String[] args) {
12    populateStack();
      popString();
14  }
```

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

```
    /*@ ensures !strings.isEmpty() @*/
2   private static Stack<String> strings = new
        Stack<>();

4   private static void populateStack() {
      //do nothing
6   }

8   /*@ requires !strings.isEmpty() @*/
    private static String popString() {
10    return strings.pop();
    }
12
    public static void main(String[] args) {
14    populateStack();
      popString();
16  }
```

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

Definition (Contract Verification)

The process of formally checking that a software component adheres to its specified contract, which includes preconditions, postconditions, and invariants

- Contract:
- Precondition:
- Postcondition:
- Invariant:

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

Bank Account example: invariant, precondition, postcondition

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

How do tools perform contract verification? – Generate proof obligation(s)

- If a precondition is present
- If a postcondition is present
- Generally in a loop

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

Contract verification tools usage

- Check with the tool, compile with a standard compiler
- Generate proof obligations
- Verify proof obligations using a variety of theorem provers
  - in SPARK: Z3, Alt-Ergo, CVC4 – supported via an intermediate language Why3

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

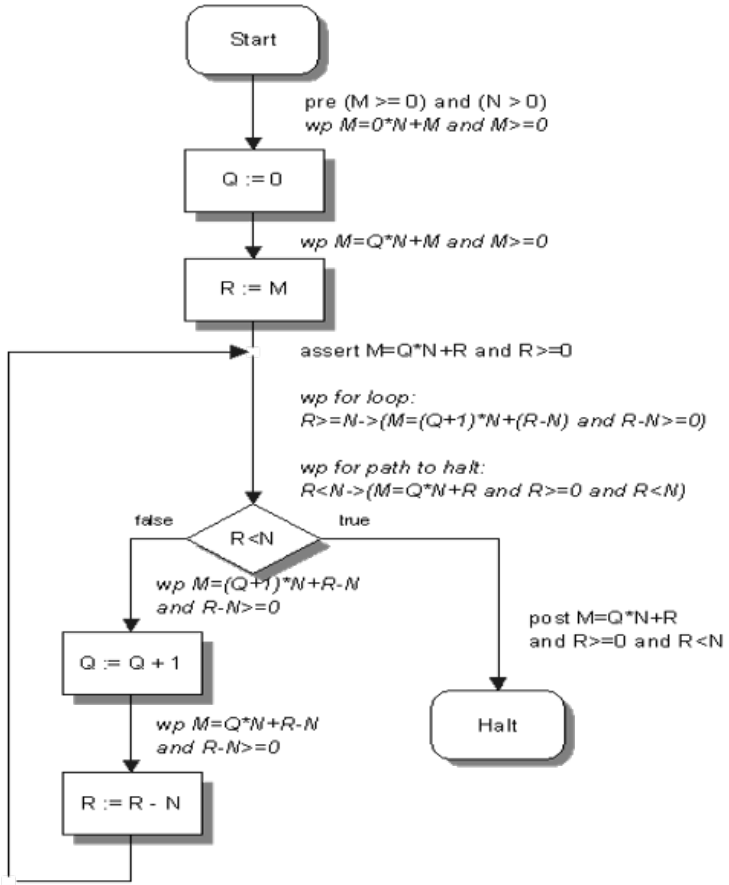
Proof Obligations – Integer Division

```
   --# pre     (M >= 0) and (N > 0);
2  --# post    (M = Q * N + R) and (R >= 0) and
       (R < N);

4  Q := 0;
   R := M;
6  loop
      --# assert (M = Q * N + R) and (R >= 0);
8     exit when R < N;
      Q := Q + 1;
10    R := R - N;
   end loop;
```

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

Proof Obligations



SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

SPARK restrictions

- Dynamic memory allocation
- Pointers
- Any other source of aliasing
- Recursion
- Expressions with side effects
  e.g. there's no equivalent of this C code

```
1       while (i++ < 5) {
          printf("Count is %d\n", i);
3       }
```

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

Is contract verification too. . .

- complicated
- restrictive
- complex

. . . to be useful and practical in enterprise-level projects?

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

Sometimes contract verification is needed

- Avionics (flight control)
- High security
  `http://www.adacore.com/sparkpro/tokeneer/`
- Air traffic control (Z + CSP + SPARK)
- Nuclear plant control?

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

Class exercise: Take about 3 minutes to discuss and answer:

- What do these three Static Analysis (SA) approaches have in common?
  - Type Checking
  - Code Quality Analysis
  - Contract Verification

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

SA Approaches Commonalities

- - Type Checking:
  - CQA:
  - Contract Verifications:
-
-
- Need

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

Aspect	Type Checking	CQA	Contract Verification
What			
Example			
Tools	Built into compilers (Java, Rust, TypeScript)	SonarQube, ESLint, Pylint, Checkstyle	SPARK/Ada, Frama-C, Dafny, OpenJML

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

- Dataflow analysis (and null pointer analysis)
- Model checking
- Formal reasoning
  - Hoare logic
  - Automated theorem proving

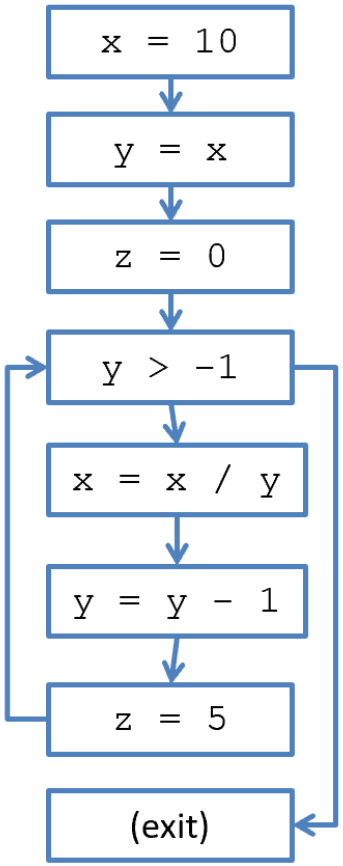
SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Dataflow Analysis
Model Checking
Formal Reasoning
Quiz

```
1   x = 10;
    y = x;
3   z = 0;
    while (y > -1) {
5     x = x / y;
      y = y - 1;
7     z = 5;
    }
```

Zero Analysis:

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Dataflow Analysis
Model Checking
Formal Reasoning
Quiz

```
x = 10;
y = x;
z = 0;
while (y > -1) {
  x = x / y;
  y = y - 1;
  z = 5;
}
```



SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Dataflow Analysis
Model Checking
Formal Reasoning
Quiz

Applying Zero Analysis

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Dataflow Analysis
Model Checking
Formal Reasoning
Quiz

Termination: Fixed Point

- Analysis values will not change, no matter how many times loop executes
- Proof: our analysis is deterministic
  - We run through the loop with the current analysis values, none of them change
  - Therefore, no matter how many times we run the loop, the results will remain the same
- We have computed the dataflow analysis results for any number of loop iterations

Example final result: `x:NZ, y:MZ, z:MZ`

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Dataflow Analysis
Model Checking
Formal Reasoning
Quiz

Abstraction Advantages

- Number of possible states gigantic
  - n 32 bit variables results in states

    e.g:
  - With loops, states can change indefinitely
- Zero Analysis narrows the state space
  - Zero or not zero (and possibly maybe zero!) results in e.g.
  - When this limited space is explored, then we are done
  - Extrapolate over all loop iterations
- Null pointer analysis works same way over the CFG

What are the tradeoffs?

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Dataflow Analysis
Model Checking
Formal Reasoning
Quiz

- Build model of a program and exhaustively evaluate that model against a specification
  - Check properties hold
  - Produce counter examples
- Common form: uses temporal logic formula to describe properties
- Used for:

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Dataflow Analysis
Model Checking
Formal Reasoning
Quiz

- Tool: SPIN (Simple Promela (Process Meta Language) INterpreter) `http://spinroot.com`
- Well established and freely available model checker
- Model processes for concurrent and distributed systems
- Verify linear temporal logic properties

```
active proctype Hello() {
  printf("Hello World\n");
}
```

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Dataflow Analysis
Model Checking
Formal Reasoning
Quiz

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Dataflow Analysis
Model Checking
Formal Reasoning
Quiz

```
1   active proctype Hello() {
      printf("Hello from Process 1\n");
3   }

5   active proctype World() {
      printf("Hello from Process 2\n");
7   }
```

Possible outputs:

Example

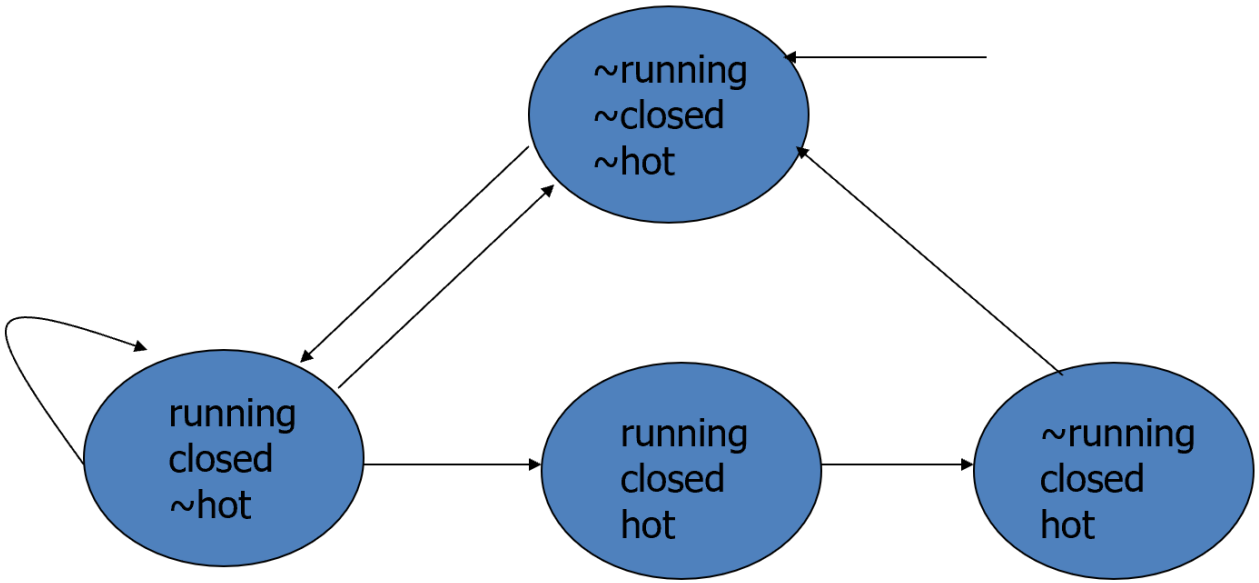
- A simple microwave oven system
- States of the system correspond to values of 3 boolean variables
  - Either door is closed or not closed
  - Either microwave is running or it is stopped
  - Either the food in the microwave is warm or it is cold

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Dataflow Analysis
Model Checking
Formal Reasoning
Quiz

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Dataflow Analysis
Model Checking
Formal Reasoning
Quiz

Example (cont'd)

Model as transition system



Example (cont'd)

- Using Temporal Logic, one can say
  - **Specification**: microwave doesn't heat the food up until the door is closed
  - Equivalent to: hot holds until closed
  - **Formula**: $f = (\text{ hot})\ \mathbf{U}\ \text{closed}$
- Given f and the model
  - A model checking algorithm can automatically return if the model satisfies f
  - If not, a counterexample is returned, showing a path of execution where the system fails to satisfy the formula
- Pros:
- Trade-offs:

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Dataflow Analysis
Model Checking
Formal Reasoning
Quiz

Hoare Logic

- Formal system used for reasoning about the correctness of a program
  - Uses pre and post conditions
- The Hoare triple: $\{P\}\ S\ \{Q\}$
  - P, Q: predicates, S: program
  - If we start in a state where P is true and execute S, then S will terminate in a state where Q is true

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Dataflow Analysis
Model Checking
Formal Reasoning
Quiz

Hoare Triples

```
1   {P}          S                    {Q}

3   {x=y}        x := x + 3           {x = y + 3}

5   {x > -1}     x := x * 2 + 3       {x > 1}

7   {true}       x := 5               {x=5}
```

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Dataflow Analysis
Model Checking
Formal Reasoning
Quiz

- Hoare logic defines inference rules for program constructs
  - Assignments: $\{x{=}1\}\ x := x + 1\ \{x{=}2\}$
  - Conditionals:

    $$\{P\}\ \text{if } x > 0 \text{ then } y := z \text{ else } y := -z\ \{y > 5\}$$

  - Loops: $\{P\}$ while B do S $\{Q\}$
- Using these rules, we can reason about program correctness (if we have defined proper pre- and post-conditions)
- Common technology in safety-critical systems programming

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Dataflow Analysis
Model Checking
Formal Reasoning
Quiz

Tool: ESC (Extended Static Checking for Java) [FLL+02]

- Uses the **Simplify** theorem prover to evaluate each routine in a program
- Embedded assertions/annotations
- Checker readable comments
- Based on the Java Modeling Language (JML)

```
/*@ requires i != 0 *@/
public void div(int i, int j) {
  return j/i;
}
```

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Dataflow Analysis
Model Checking
Formal Reasoning
Quiz

Theorem provers

- Alt-Ergo
- Prover9
- Z3
- CVC
- KIV
- PVS
- Agda
- Isabelle
- ...

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Dataflow Analysis
Model Checking
Formal Reasoning
Quiz

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Dataflow Analysis
Model Checking
Formal Reasoning
Quiz

The Quiz - again

- Reminder: quizzes like this give a big increase in long-term recall

`https://bit.ly/3F3XqGk`

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

The Quiz

1. What is **static analysis**?
2. What's the relationship, in terms of practical value, between static analysis and dynamic testing?
3. What's the most commonly used form of static analysis?
4. What can code quality analysis tools (e.g. Lint, FindBugs, FXCop etc) do for you as a developer?
5. Contract verification (as supported by SPARK) supports very powerful and accurate analysis. What are the main drawbacks to using it?
6. In practice, what's the biggest barrier that enterprise-scale projects face when they think about implementing more advanced static analysis?

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

The Quiz - Possible Answers

1. What is static analysis?

2. What's the relationship, in terms of practical value, between static analysis and dynamic testing?

3. What's the most commonly used form of static analysis?

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

The Quiz - Possible Answers (cont'd)

4. What can code quality analysis tools (e.g. Lint, FindBugs, FXCop etc) do for you as a developer?

5. Contract verification (as supported by SPARK and Eiffel) supports very powerful and accurate analysis. What are the main drawbacks to using it?

SE 3SO3: Reviews and Evaluation

A. Marinache

Preliminaries

Type Checking

Code Quality Analysis

Contract Verification

Advanced Notions

Quiz

The Quiz - Possible Answers (cont'd)

6. In practice, what's the biggest barrier that enterprise-scale projects face when they think about implementing more advanced static analysis?

SE 3SO3: Reviews and Evaluation
A. Marinache
Preliminaries
Type Checking
Code Quality Analysis
Contract Verification
Advanced Notions
Quiz

References I


collective, Quantitative comparison of unit testing vs. static typing?, March 30 2012.


Cormac Flanagan, K Rustan M Leino, Mark Lillibridge, Greg Nelson, James B Saxe, and Raymie Stata, Extended static checking for java, Proceedings of the ACM SIGPLAN 2002 Conference on Programming language design and implementation, 2002, pp. 234–245.


Martin Fowler, Dynamic typing, March 14 2005.

SE 3SO3: Reviews and Evaluation

A. Marinache

Preliminaries

Type Checking

Code Quality Analysis

Contract Verification

Advanced Notions

Quiz

References II


Hong Jin Kang, Khai Loong Aw, and David Lo, Detecting false alarms from automatic static analysis tools: How far are we?, Proceedings of the 44th International Conference on Software Engineering, 2022, pp. 698–709.


Richard F Paige, Louis M Rose, Xiaocheng Ge, Dimitrios S Kolovos, and Phillip J Brooke, Automated safety analysis for domain-specific languages, Workshop on Non-Functional System Properties in Domain Specific Modeling Languages, Citeseer, 2008.

SE 3SO3: Reviews and Evaluation

A. Marinache

Preliminaries

Type Checking

Code Quality Analysis

Contract Verification

Advanced Notions

Quiz