CS846
Machine Learning for Software Engineering

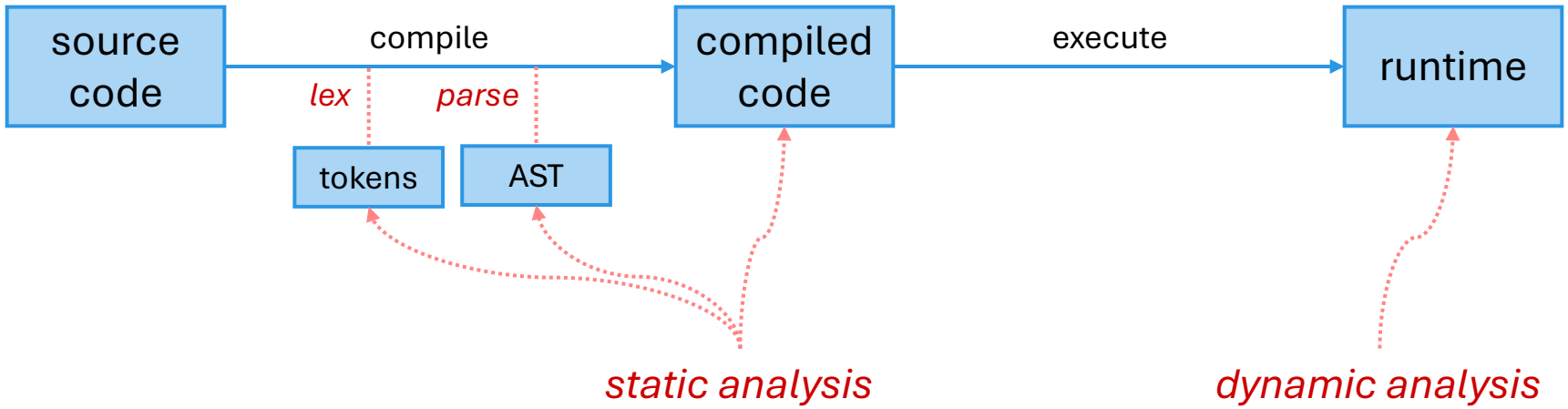
Pengyu Nie

Static Analysis

Source code analysis

Bytecode analysis

Program Analyses Overview



Examples of Static Analysis

- Kinds of data
  - call graph
  - data flow graph / def-use / taint analysis
  - type checking (esp. for dynamic typed languages)
  - path condition / symbolic execution
- Use cases
  - ML model
  - Linter

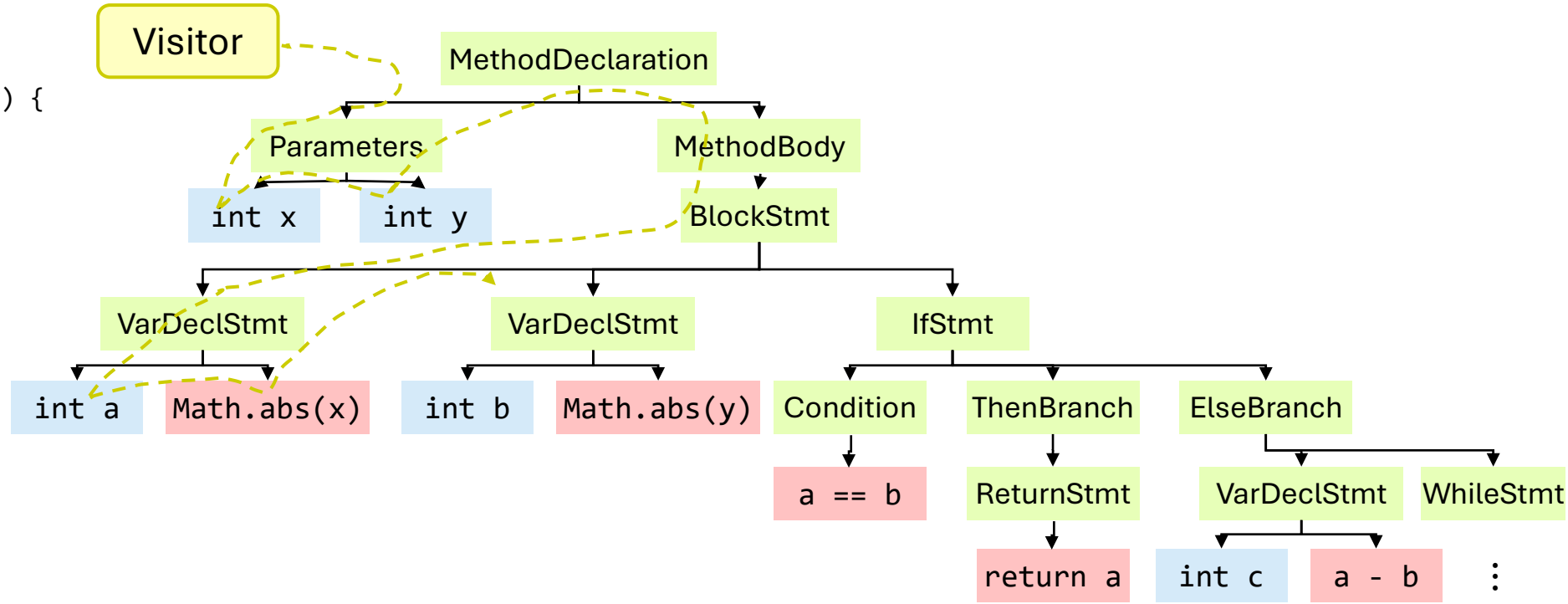
```
(seutil) pynie@PRODIGY-T16Gen2:~/projects/pytest-inline/src$ ruff check

inline/plugin.py:1258:33: F841 [*] Local variable `e` is assigned to but never used
     |
1256 |                 # TODO: still need to find the right way to import without errors. mode=ImportMode.importlib did not work
1257 |                 module = import_path(self.path, root=self.config.rootpath)
1258 |             except Exception as e:
     |                                 ^ F841
1259 |                 # (ImportError, ModuleNotFoundError, TypeError, NameError, FileNotFoundError)
1260 |                 if self.config.getvalue("inlinetest_ignore_import_errors"):
     |
     = help: Remove assignment to unused variable `e`
```

Visitor Design Pattern

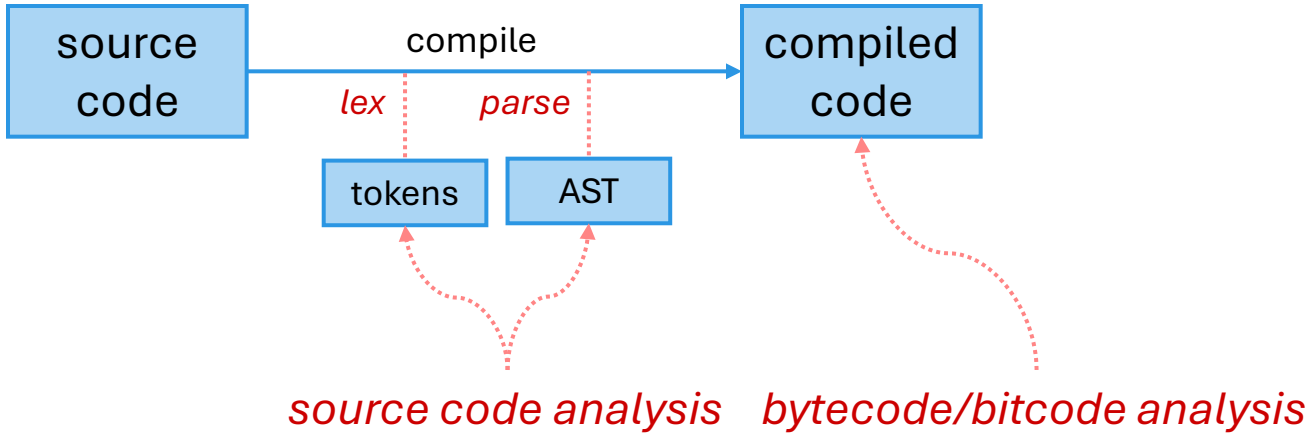
- Example: data flow graph -> finding variable def & use statements
- Suitable for traversing tree structure

```
public int foo(int x, int y) {
  int a = Math.abs(x);
  int b = Math.abs(y);
  if (a == b) {
    return a;
  } else {
    int c = a - b;
    while (a - b > 0) {
      a++;
      b--;
    }
    return a;
  }
}
```



Source Code vs. Bytecode Analysis

- Applicable for compiled languages (e.g., Java, C/C++)
- Compiler has performed many analyses / optimizations for you
  - type resolving
  - macro expansion
- Easier to extract some kinds of data (e.g., call graph)



Bytecode Analysis Resources (for Java)

- Libraries
  - ASM https://asm.ow2.io/
  - ByteBuddy https://bytebuddy.net/#/
- References
  - List of bytecode instructions Wikipedia https://en.wikipedia.org/wiki/List_of_Java_bytecode_instructions
  - Java specifications https://docs.oracle.com/javase/specs/
    source code: "The Java Language Specification, Java SE xxx Edition"
    bytecode: "The Java Virtual Machine Specification, Java SE xxx Edition"