System Dependence Graphs for Java Programs

Pavlos Milaszewicz & ~~Ito Franchilo Mikael Alcuaz~~

TECHNISCHE UNIVERSITÄT DARMSTADT

```
public class bar {
 void foo() {
   int x = source();
   if (x < MAX) {
     int y = 2 * x;
     sink(y);
   }
 }

 int source() {
   ...
   return 1;
 }

 void sink(int x) {...}
 .
 .
}
```

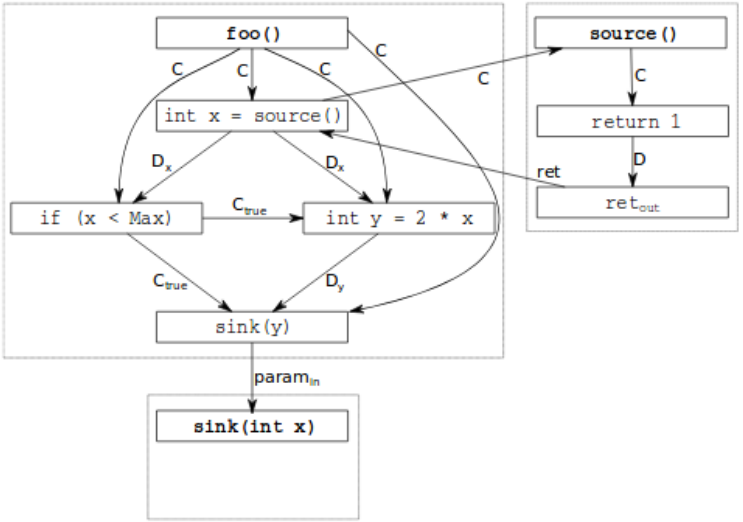


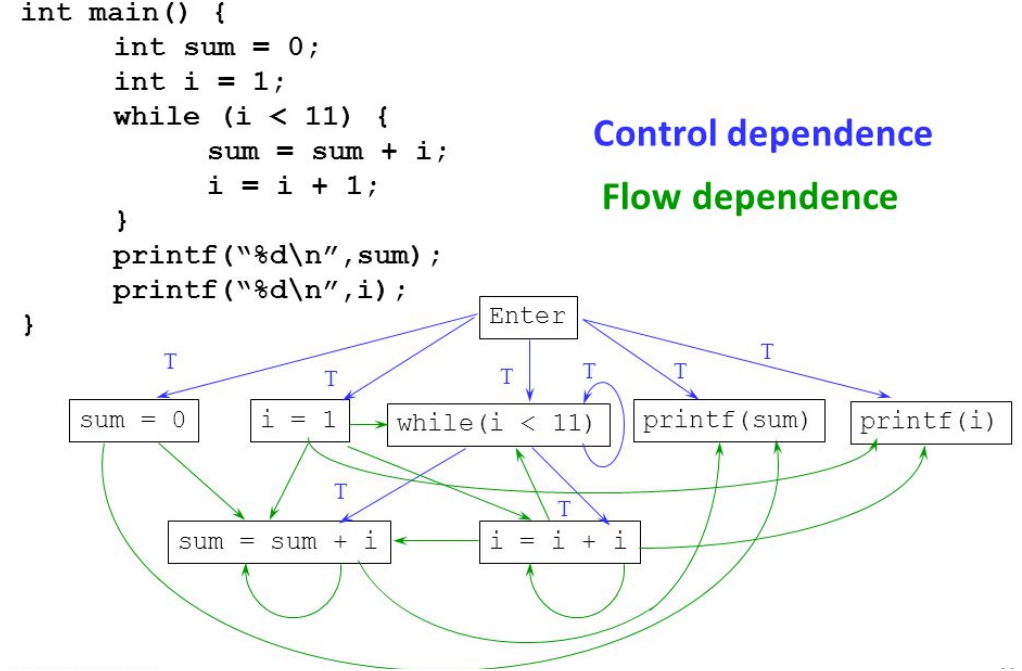
Figure 1: A code snippet and its corresponding SDG

Program Dependence Graph (PDG)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Graph based representation capturing data and control flow information.
- Intra-procedural.
- Program slicing and optimization applications.
- **Built-in PDG generator via Soot.**

```
int main() {
      int sum = 0;
      int i = 1;
      while (i < 11) {
            sum = sum + i;
            i = i + 1;
      }
      printf("%d\n",sum);
      printf("%d\n",i);
}
```



Source: http://slideplayer.com/slide/4859950/

System Dependence Graph (SDG)

TECHNISCHE UNIVERSITÄT DARMSTADT

- An extension of PDG's.
- Models both inter/intra-procedural program dependencies.
- Vulnerability analysis and bug finding applications.
- **Built-in SDG generator not currently available.**

```
public class bar {
 void foo() {
   int x = source();
   if (x < MAX) {
     int y = 2 * x;
     sink(y);
   }
 }

 int source() {
   ...
   return 1;
 }

 void sink(int x) {...}
 .
 .
}
```

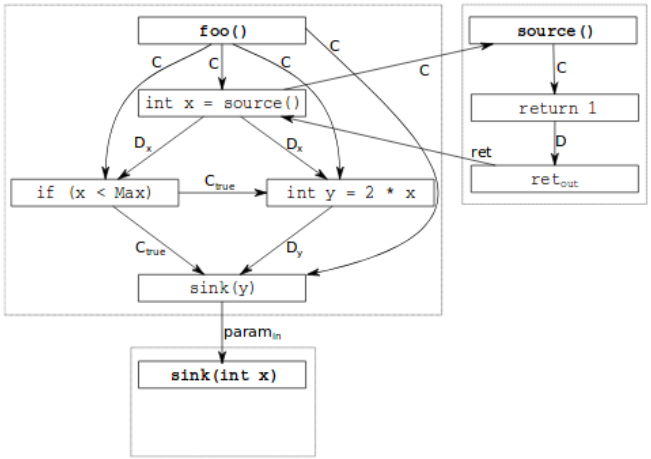


Figure 1: A code snippet and its corresponding SDG

Project Goals

TECHNISCHE UNIVERSITÄT DARMSTADT

- Extend Soot PDG generation to include support for SDG’s by implementing a static analysis tool.
- Evaluate the static analysis against a set of real-world examples.

Soot SDG’s - Approach & Implementation

TECHNISCHE UNIVERSITÄT DARMSTADT

- Consider forward branched Soot flow (as opposed to backward, non branched, etc.) analysis.
- Base case: a method with no intra-function invocation(s). ←Basically a plain PDG.
- Consider how to detect one or more intra-function calls.
- Generate additional (mini) Soot PDG(s) for the intra-function call(s).
- Append extra PDG’s to the main PDG by analyzing the location of the in and out flows.

Overall Progress

TECHNISCHE UNIVERSITÄT DARMSTADT

- Static analysis SDG generator ←~ 50% completed, problems encountered with poorly documented Soot API, out of date tutorials, etc.
- (TODO) Test analysis against real world examples and evaluate results.
- (TODO) Project report.

Possible Project Extensions

TECHNISCHE UNIVERSITÄT DARMSTADT

1) Consider corner cases (e.g. anonymous lambda functions, recursive calls, etc.).

Questions

TECHNISCHE UNIVERSITÄT DARMSTADT