

# **Program Analysis**

## **Program Slicing (Part 1)**

**Prof. Dr. Michael Pradel**

**Software Lab, University of Stuttgart**

**Winter 2020/2021**

# Outline

---

- 1. Introduction**
- 2. Static Slicing**
- 3. Thin Slicing**
- 4. Dynamic Slicing**

Mostly based on these papers:

- *Program Slicing*, Weiser., IEEE TSE, 1984
- *Thin Slicing*, Sridharan et al., PLDI 2007
- *Dynamic Program Slicing*, Agrawal and Horgan, PLDI 1990
- *A Survey of Program Slicing Techniques*, Tip, J Prog Lang 1995

# Program Slicing

---

Extract an **executable subset of a program** that (potentially) **affects the values at a particular program location**

- **Slicing criterion** = program location + variable
- An observer focusing on the slicing criterion **cannot distinguish** a run of the program from a run of the slice

# Example

---

```
var n = readInput();
var i = 1;
var sum = 0;
var prod = 1;
while (i <= n) {
    sum = sum + i;
    prod = prod * i;
    i = i + 1;
}
console.log(sum);
console.log(prod);
```

# Example

---

```
var n = readInput();  
var i = 1;  
var sum = 0;  
var prod = 1;  
while (i <= n) {  
    sum = sum + i;  
    prod = prod * i;  
    i = i + 1;  
}  
console.log(sum);  
console.log(prod);
```

**Slice for value  
of sum at this  
statement?**

# Example

---

```
var n = readInput();  
var i = 1;  
var sum = 0;  
var prod = 1;  
while (i <= n) {  
    sum = sum + i;  
    prod = prod * i;  
    i = i + 1;  
}  
console.log(sum);  
console.log(prod);
```

**Slice for value  
of sum at this  
statement?**

# Example

---

```
var n = readInput();
var i = 1;
var sum = 0;
var prod = 1;
while (i <= n) {
    sum = sum + i;
    prod = prod * i;
    i = i + 1;
}
console.log(sum);
console.log(prod);
```

**Slice for value  
of prod at this  
statement**



# Example

---

```
var n = readInput();  
var i = 1;  
var sum = 0;  
var prod = 1;  
while (i <= n) {  
    sum = sum + i;  
    prod = prod * i;  
    i = i + 1;  
}  
console.log(sum);  
console.log(prod);
```

**Slice for value  
of n at this  
statement**





# Why Do We Need Slicing?

---

## Various applications, e.g.

- **Debugging**: Focus on parts of program relevant for a bug
- **Program understanding**: Which statements influence this statement?
- **Change impact analysis**: Which parts of a program are affected by a change? What should be retested?
- **Parallelization**: Determine parts of program that can be computed independently of each other

# Slicing: Overview

---

## Forward vs. backward

- Backward slice (our focus): Statements that **influence** the slicing criterion
- Forward slice: Statements that **are influenced** by the slicing criterion

## Static vs. dynamic

- Statically computing a minimum slice is undecidable
- Dynamically computed slice focuses on particular execution/input