# Program Analysis

# Path Profiling (Part 3)

**Prof. Dr. Michael Pradel**

**Software Lab, University of Stuttgart**

**Winter 2020/2021**

# Outline

1. **Motivation and Challenges**
2. **Ball-Larus algorithm for DAGs**
3. **Generalization and Applications** 

Mostly based on this paper:

- *Efficient path profiling*, Ball and Larus, MICRO 1996

Other reading material:

- *Whole program paths* , Larus, PLDI 1999
- *HOLMES: Effective statistical debugging via efficient path profiling*, Chilimbi et al., ICSE 2009

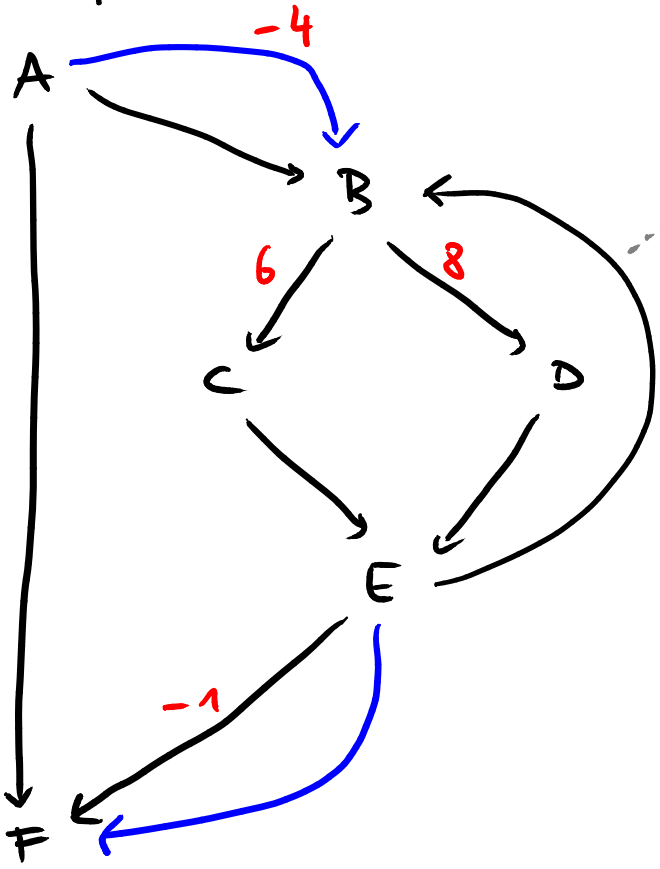
# Generalizing to Cyclic CFGs

- **For each backedge $n \rightarrow m$, add dummy edges**
  - $Entry \rightarrow m$
  - $n \rightarrow Exit$
- **Remove backedges and add DAG-based increments**
- **In addition, add instrumentation to each backedge**
  - `count[r]++; r=0`

# Generalizing to Cyclic CFGs (2)

- **Leads to four kinds of paths**
  - From entry to exit
  - From entry to backedge
  - From end of backedge to beginning of (possibly another) backedge
  - From end of backedge to exit
- **Full path information can be constructed from these four kinds**

## Example: Generalizing




backedge

dummy edge

| Path | Encoding |
|---|---|
| AF | 0 |
| ABCEF | 1 |
| ABCE | 2 |
| ABDEF | 3 |
| ABDE | 4 |
| BCEF | 5 |
| BCE | 6 |
| BDEF | 7 |
| BDE | 8 |

# Applications

- **Performance optimization**
  - Frequent path should get most attention by optimizer
- **Statistical debugging**
  - Paths correlated with failure are more likely to contain the bug
- **Energy analysis**
  - Warn developers about paths and statements associated with high power consumption