Program Analysis – Lecture 4

Data Flow Analysis (Part 1)

Prof. Dr. Michael Pradel

Software Lab, University of Stuttgart

Winter 2019/2020

Big Picture

- **Static versus dynamic analysis**
- **Many ways of formulating and implementing analyses**
- **Next few lectures: One popular way to formulating a static analysis: Data flow analysis**

Data Flow Analysis

Basic idea

- Propagate analysis information along the edges of a control flow graph
- Goal: Compute analysis state at each program point
- For each statement, define how it affects the analysis state
- For loops: Iterate until fix-point reached

Outline

- First example: Available expressions
- Basic principles
- More examples
- Monotonicity framework (a.k.a. solving data flow problems)
- Sensitivities

  Inter-procedural analysis

Available Expression Analysis

Goal: For each program point, compute which expression must have already been computed, and not later modified

- Useful, e.g., to avoid re-computing an expression
- Used as part of compiler optimizations

Example

```
var x = a + b;
var y = a * b;
while (y > a + b) {
  a = a + 1;
  x = a + b;
}
```

Example

```
var x = a + b;
var y = a * b;
while (y > a + b) {
  a = a + 1;
  x = a + b;
}
```

Available every time execution reaches this point

Transfer Functions

- Transfer function of a statement: How the statement affects the analysis state
- Here: Analysis state = available expressions
- Two functions
  - gen: Available expressions generated by a statement
  - kill: Available expressions killed by a statement

gen Function

Function $gen : Stmt \rightarrow \mathcal{P}(Expr)$

- A statement generates an available expression e if
  - it evaluates e and
  - it does not later write any variable used in e
- Otherwise, function returns empty set

Example:

`var x = a * b;` **generates** `a * b`

kill Function

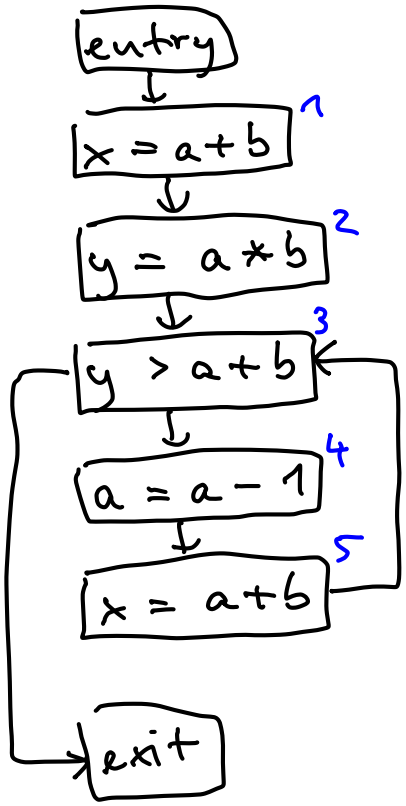
Function $kill : Stmt \rightarrow \mathcal{P}(Expr)$

- A statement kills an available expression e if
  - it modifies any of of the variables used in e
- Otherwise, function returns empty set

Example:

`a = 23;` **kills** `a * b`

Control flow graph



7 nodes, 7 edges

Non-trivial expressions:

$a+b$

$a*b$

$a-1$

Transfer function for each statement

Statement s	gen(s)	kill(s)
1	$\{a+b\}$	$\emptyset$
2	$\{a*b\}$	$\emptyset$
3	$\{a+b\}$	$\emptyset$
4	$\emptyset$	$\{a+b, a*b, a-1\}$
5	$\{a+b\}$	$\emptyset$