Program Analysis

Data Flow Analysis (Part 1)

Prof. Dr. Michael Pradel

Software Lab, University of Stuttgart

Winter 2020/2021

Big Picture

- **Static versus dynamic analysis**
- **Many ways of formulating and implementing analyses**
- **This lecture: One popular way of 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
- Solving data flow problems
- Inter-procedural analysis
- Sensitivities

Available Expression Analysis

Goal: For each program point, compute which expressions 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 expressions 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 expressions e if
  - it modifies any of the variables used in e
- Otherwise, function returns empty set

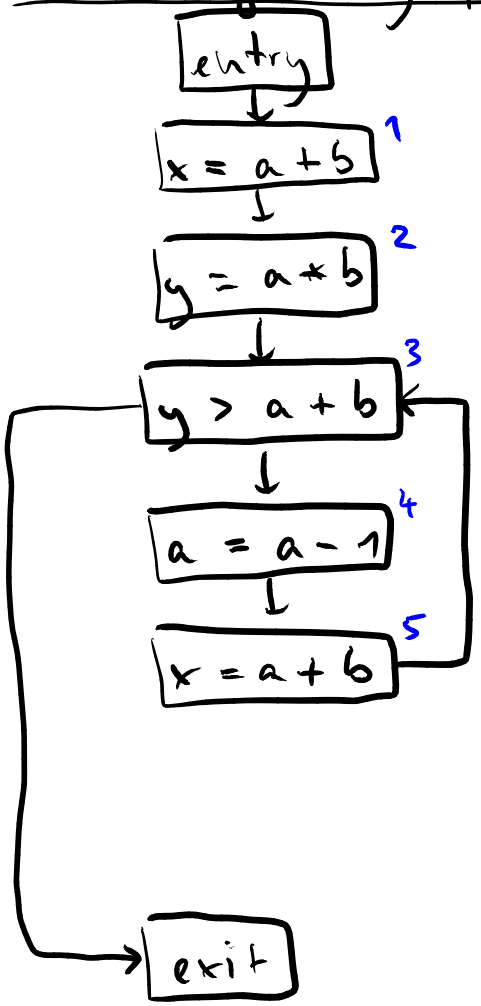
Example:

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

Example

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

Control flow graph



Non-trivial expressions:

$a + b$

$a * b$

$a - 1$

Transfer function f. 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	$\{a+b\}$	$\emptyset$

Propagating Available Expressions

- Initially, no available expressions
- Forward analysis: Propagate available expressions in the direction of control flow
- For each statement s, outgoing available expressions are:
  incoming avail. exprs. minus $kill(s)$ plus $gen(s)$
- When control flow splits, propagate available expressions both ways
- When control flows merge, intersect the incoming available expressions

Data flow equations

$AE_{entry}(s)$ .. avail. expr. at entry of s

$AE_{exit}(s)$ .. avail. expr. at exit of s

$AE_{entry}(1) = \emptyset$

$AE_{entry}(2) = AE_{exit}(1)$

$AE_{entry}(3) = AE_{exit}(2) \cap AE_{exit}(5)$

$AE_{entry}(4) = AE_{exit}(3)$

$AE_{entry}(5) = AE_{exit}(4)$.

$AE_{exit}(1) = AE_{entry}(1) \cup \{a+b\}$

$AE_{exit}(2) = AE_{entry}(2) \cup \{a*b\}$

$AE_{exit}(3) = AE_{entry}(3) \cup \{a+b\}$

$AE_{exit}(4) = AE_{entry}(4) \setminus \{a+b, a*b, a-1\}$

$AE_{exit}(5) = AE_{entry}(5) \cup \{a+b\}$

Solution of these equations

s	$AE_{entry}(s)$	$AE_{exit}(s)$
1	$\emptyset$	$\{a+b\}$
2	$\{a+b\}$	$\{a+b, a*b\}$
3	$\boxed{\{a+b\}}$	$\{a+b\}$
4	$\{a+b\}$	$\emptyset$
5	$\emptyset$	$\{a+b\}$

see example

Quiz

```
var m = x - y;
if (random()) {
  while (m > 0) {
    x = y + 1;
  }
} else {
  n = x - y;
}
z = x - y;
```

Quiz

```
var m = x - y;
if (random()) {
  while (m > 0) {
    x = y + 1;
  }
} else {
  n = x - y;
}
z = x - y;
```

Is `x - y` an available expression when entering this statement?

Quiz

```
var m = x - y;
if (random()) {
  while (m > 0) {
    x = y + 1;
  }
} else {
  n = x - y;
}
z = x - y;
```

No, because modifying `x` kills `x - y`

Is `x - y` an available expression when entering this statement?