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
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

```javascript
var x = a + b;
var y = a * b;
while (y > a + b) {
    a = a + 1;
    x = a + b;
}
```
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 \( \text{gen} : \text{Stmt} \rightarrow \mathcal{P}(Expr) \)

- 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:

```plaintext
var x = a * b;
generates a * b
```
**kill** Function

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

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

**Example:**
\[
a = 23; \text{kills } a \times b
\]
Control flow graph

\[
\begin{align*}
\text{entry} & \quad \xrightarrow{1} \quad x = a+b \\
\text{entry} & \quad \xrightarrow{2} \quad y = a\times b \\
\text{entry} & \quad \xrightarrow{3} \quad y > a+b \\
\text{entry} & \quad \xrightarrow{4} \quad a = a-1 \\
\text{entry} & \quad \xrightarrow{5} \quad x = a+b
\end{align*}
\]

Non-trivial expressions:

\[
\begin{align*}
a + b \\
a \times b \\
a - 1
\end{align*}
\]

Transfer function for each statement

<table>
<thead>
<tr>
<th>Statement, s</th>
<th>gen(s)</th>
<th>kill(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{a+b}</td>
<td>\emptyset</td>
</tr>
<tr>
<td>2</td>
<td>{a \times b}</td>
<td>\emptyset</td>
</tr>
<tr>
<td>3</td>
<td>{a+b}</td>
<td>\emptyset</td>
</tr>
<tr>
<td>4</td>
<td>\emptyset</td>
<td>{a+b, a \times b, a-1}</td>
</tr>
<tr>
<td>5</td>
<td>{a+b}</td>
<td>\emptyset</td>
</tr>
</tbody>
</table>

7 nodes, 7 edges