CSCI 742 - Compiler Construction

Lecture 36
Available Expressions Analysis
Instructor: Hossein Hojjat

May 1, 2017
Recap: Live Variable Analysis

- Compute which variables are live at each program point
- Live variable information flows backward
- Derive a system of constraints between live variable sets

\[
\text{in}(S) = (\text{out}(S) \setminus \text{def}(S)) \cup \text{use}(S)
\]

\[
\text{out}(S) = \bigcup_{S_i \in \text{succ}(S)} \text{in}(S_i)
\]

- Solve constraints in an iterative algorithm
Idea: some computation may be a redundant repetition of earlier computation within the same program

An expression like $x+y$ is available at a statement $S$ if

- Every path from the entry node to $S$ compute $x+y$ before reaching $S$
- There are no assignments to $x$ or $y$ since the last time $x+y$ was computed on the paths to $S$

Optimization: If an expression is available, don’t need to recompute it
Example: Available Expression

- Is the expression available?

\[ x = y + z \]
\[ w = v - u \]
\[ u = x \times z \]
\[ t = x \times z \]
\[ y = x + z \]
\[ s = z + u \]
\[ u = x \times z + y + z + w \]
Example: Available Expression

- Is the expression available?

\[
x = y + z \\
w = v - u \\
u = x \times z \\
t = x \times z \\
w = x - t \\
y = x + z \\
s = z + u \\
u = x \times z + y + z + w
\]
Example: Available Expression

- Is the expression available?

\begin{align*}
x &= y + z \\
w &= v - u \\
u &= x \times z \\
w &= x - t \\
t &= x \times z \\
s &= z + u \\
u &= x \times z + y + z + w \\
y &= x + z
\end{align*}
• Is the expression available?

\[ x = y + z \]
\[ w = v - u \]
\[ u = x \times z \]
\[ t = x \times z \]
\[ w = x - t \]
\[ y = x + z \]
\[ s = z + u \]
\[ u = x \times z + y + z + w \]
Example: Available Expression

• Is the expression available?

\[
\begin{align*}
x &= y + z \\
w &= v - u \\
u &= x \times z \\
t &= x \times z \\
w &= x - t \\
y &= x + z \\
s &= z + u \\
u &= x \times z + y + z + w
\end{align*}
\]
Example: Available Expression

- Is the expression available?

\[
\begin{align*}
x &= y + z \\
w &= v - u \\
u &= x \times z \\
t &= x \times z \\
w &= x - t \\
y &= x + z \\
s &= z + u \\
u &= x \times z + y + z + w
\end{align*}
\]
Example: Available Expression

- Is the expression available?

```
x = y + z
w = v - u
u = x \times z
t = x \times z
w = x - t
y = x + z
s = z + u
u = x \times z + y + z + w
```
Example: Available Expression

- Is the expression available?

\[
\begin{align*}
x &= y + z \\
w &= v - u \\
u &= x \times z \\
t &= x \times z \\
w &= x - t \\
y &= x + z \\
s &= z + u \\
u &= x \times z + y + z + w
\end{align*}
\]
Example: Available Expression

- Is the expression available?

\[
\begin{align*}
x &= y + z \\
w &= v - u \\
u &= x \times z \\
t &= x \times z \\
y &= x + z \\
s &= z + u \\
u &= x \times z + y + z + w
\end{align*}
\]
Example: Available Expression

- Is the expression available?

\[
\begin{align*}
x &= y + z \\
w &= v - u \\
u &= x \times z \\
t &= x \times z \\
y &= x + z \\
s &= z + u \\
u &= x \times z + y + z + w
\end{align*}
\]
• Is the expression available?

\begin{align*}
x &= y + z \\
w &= v - u \\
u &= x \times z \\
t &= x \times z \\
w &= x - t \\
y &= x + z \\
s &= z + u \\
u &= x \times z + y + z + w
\end{align*}
Example: Available Expression

- Is the expression available?

\[
x = y + z \\
w = v - u \\
u = x \times z \\
t = x \times z \\
w = x - t \\
y = x + z \\
s = z + u \\
u = x \times z + y + z + w
\]
Example: Available Expression

- How can we use available expression?
How can we use available expression?

\[
\begin{align*}
x &= y + z \\
w &= v - u \\
u &= x \times z \\
t &= u \\
w &= x - t \\
s &= z + u \\
y &= x + z \\
u &= x \times z + y + z + w
\end{align*}
\]
How can we use available expression?

\[ x = y + z \]
\[ w = v - u \]
\[ u = x \times z \]
\[ t = u \]
\[ w = x - t \]
\[ s = z + u \]
\[ y = x + z \]
\[ u = x \times z + y + z + w \]
How can we use available expression?

\[ x = y + z \]
\[ w = v - u \]
\[ u = x \times z \]
\[ t = u \]
\[ w = x - t \]
\[ s = z + u \]
\[ y = x + z \]
\[ u = u + y + z + w \]
Available Expression Analysis: Forward

- Available expression analysis is a **forward** data-flow analysis
- Information from past instructions must be propagated forward through the program to discover which expressions are available

```java
int t = x*y; println(x*y);
if(x*y!=6) ...;
...
t int z = x * y;
}
```

- **Unlike** variable liveness, expression availability flows forward through the program
- **Like** liveness, each instruction has an effect on the availability information as it flows past
Available Expression Analysis

- A statement makes an expression **available** when it generates (computes) its current value.
A statement makes an expression **unavailable** when it kills (invalidates) its current value.

\[
\begin{align*}
\{x+y, v+5, z*t, t-1\} & \quad \text{kills } x+y \\
v = 11; \quad \{v+5, z*t, t-1\} & \quad \text{kills } v+5 \\
\{z*t, t-1\} & \\
t = 7; \quad \{z*t, t-1\} & \quad \text{kills } z*t, t-1 \\
\{\} &
\end{align*}
\]
Available Expression Analysis

- As in Live Variable Analysis, we create functions $\text{gen}(S)$ and $\text{kill}(S)$ which give the sets of expressions the statement $S$ generates and kills.
- Assignment to a variable $x$ kills all expressions in the program which contain occurrences of $x$ ($\mathcal{E}_x$).

\[
\text{gen(println(x+5))} = \{x+5\} \quad \text{kill(x = 3)} = \{\mathcal{E}_x\}
\]

\[
\text{gen(println(x+5))} = \{y-1\} \quad \text{kill(x = 3)} = \{y-1, x+5\}
\]

\[
\text{gen(println(x+5))} = \{y-1, x+5\} \quad \text{kill(x = 3)} = \{y-1\}
\]
Available Expression Analysis

- If a statement both generates and kills expressions, remove the killed expressions after adding the generated ones.

\[
\begin{align*}
\text{gen}(x = x + y) &= \{x+y\} \\
\text{kill}(x = x + y) &= \{E_x\}
\end{align*}
\]

In general:

- \( \text{in}(S) \): set of available expressions immediately before statement \( S \)
- \( \text{out}(S) \): set of available expressions immediately after statement \( S \)

\[
\text{out}(S) = (\text{in}(S) \cup \text{gen}(S)) \setminus \text{kill}(S)
\]
Multiple Successors

- An expression is available at beginning of statement $S$ if it is available at the end of all predecessor statements

\[
in(S) = \bigcap_{S_i \in \text{pred}(S)} \text{out}(S_i)
\]
Data-flow Equations

- Start with CFG and derive a system of constraints between sets of available expressions

\[
\begin{align*}
out(S) &= (in(S) \cup gen(S)) \setminus kill(S) \\
\text{in}(S) &= \bigcap_{S_i \in \text{pred}(S)} out(S_i)
\end{align*}
\]

Solve constraints:

- Start with empty set of available expressions at start and universal set of available expressions everywhere else
- Iteratively apply constraints
- Stop when we reach a fixed point
Exercise

Compute the set of available expressions at each point of the program

\[ a = x - y \]
\[ [x = 0] \]
\[ A_0 = {} \]
\[ A_1 = \{ \text{all} \} \]
\[ A_2 = \{ \text{all} \} \]
\[ A_3 = \{ \text{all} \} \]
\[ A_4 = \{ \text{all} \} \]
\[ A_5 = \{ \text{all} \} \]
\[ A_6 = \{ \text{all} \} \]
\[ A_7 = \{ \text{all} \} \]
\[ A_8 = \{ \text{all} \} \]
\[ A_9 = \{ \text{all} \} \]
\[ A_{10} = \{ \text{all} \} \]
Exercise

Compute the set of available expressions at each point of the program

- $A_0 = \{\}$
  - $a = x - y$
  - $[x = 0]$
  - $A_1 = \{x - y\}$
  - $[x \neq 0]$
  - $A_2 = \{\text{all}\}$
  - $x = z$
  - $A_3 = \{\text{all}\}$
  - $b = x - y$
  - $i = x - y$
  - $A_4 = \{\text{all}\}$
  - $[i < n]$
  - $A_5 = \{\text{all}\}$
  - $[-(i < n)]$
  - $A_6 = \{\text{all}\}$
  - $c = x - y$
  - $A_7 = \{\text{all}\}$
  - $d = x - y$
  - $A_8 = \{\text{all}\}$
  - $i = i + c$
  - $A_9 = \{\text{all}\}$
  - $A_{10} = \{\text{all}\}$
Exercise

Compute the set of available expressions at each point of the program

\[ a = x - y \]
\[ [x \neq 0] \]
\[ x = z \]
\[ x = 0 \]
\[ b = x - y \]
\[ i = x - y \]
\[ [i < n] \]
\[ [\neg(i < n)] \]
\[ c = x - y \]
\[ d = x - y \]
\[ i = i + c \]
\[ A_0 = \{\} \]
\[ A_1 = \{x - y\} \]
\[ A_2 = \{x = 0, x - y\} \]
\[ A_3 = \{\text{all}\} \]
\[ A_4 = \{\text{all}\} \]
\[ A_5 = \{\text{all}\} \]
\[ A_6 = \{\text{all}\} \]
\[ A_8 = \{\text{all}\} \]
\[ A_9 = \{\text{all}\} \]
\[ A_{10} = \{\text{all}\} \]
Exercise

Compute the set of available expressions at each point of the program

\[ a = x - y \]

\[ A_0 = \{ \} \]

\[ A_1 = \{ x - y \} \]

\[ A_2 = \{ x = 0, x - y \} \]

\[ x = z \]

\[ A_3 = \{ \} \]

\[ b = x - y \]

\[ A_4 = \{ \text{all} \} \]

\[ i = x - y \]

\[ A_5 = \{ \text{all} \} \]

\[ A_7 = \{ \text{all} \} \]

\[ c = x - y \]

\[ A_6 = \{ \text{all} \} \]

\[ d = x - y \]

\[ A_9 = \{ \text{all} \} \]

\[ i = i + c \]

\[ A_8 = \{ \text{all} \} \]

\[ A_{10} = \{ \text{all} \} \]
Compute the set of available expressions at each point of the program

\( a = x - y \)
\([x = 0]\)
\( A_0 = \{\} \)

\( A_2 = \{x = 0, x - y\} \)

\( x = z \)
\( A_3 = \{} \)

\( \ell : [x ≠ 0] \)
\( out(\ell) = \{x - y, x ≠ 0\} \)

\( i = x - y \)
\( A_4 = \{\text{all}\} \)

\( i < n \)
\( A_5 = \{\text{all}\} \)

\( A_7 = \{\text{all}\} \)

\( c = x - y \)
\( A_9 = \{\text{all}\} \)

\( i = i + c \)

\( A_{10} = \{\text{all}\} \)

\( \ell' : b = x - y \)
\( out(\ell') = \{x - y\} \)

\( \neg(i < n) \)
\( A_6 = \{\text{all}\} \)

\( d = x - y \)
\( A_8 = \{\text{all}\} \)

Exercise
Exercise

Compute the set of available expressions at each point of the program

\[ a = x - y \]

\[ A_0 = \{\} \]

\[ A_1 = \{x - y\} \]

\[ A_2 = \{x = 0, x - y\} \]

\[ A_3 = \{\} \]

\[ A_4 = \{x - y\} \]

\[ A_5 = \{\text{all}\} \]

\[ A_6 = \{\text{all}\} \]

\[ A_7 = \{\text{all}\} \]

\[ A_8 = \{\text{all}\} \]

\[ A_9 = \{\text{all}\} \]

\[ A_{10} = \{\text{all}\} \]
Exercise

Compute the set of available expressions at each point of the program

\[ a = x - y \]
\[ [x = 0] \]
\[ A_2 = \{ x = 0, x - y \} \]
\[ A_3 = \{ \} \]
\[ \ell : [x \neq 0] \]
\[ \text{out}(\ell) = \{ x - y, x \neq 0 \} \]
\[ A_4 = \{ x - y \} \]
\[ A_5 = \{ x - y \} \]
\[ A_7 = \{ \text{all} \} \]
\[ A_9 = \{ \text{all} \} \]
\[ A_{10} = \{ \text{all} \} \]
\[ A_0 = \{ \} \]
\[ A_1 = \{ x - y \} \]
\[ A_8 = \{ \text{all} \} \]
Exercise

Compute the set of available expressions at each point of the program

\[ a = x - y \]
\[ [x \neq 0] \]
\[ x = z \]
\[ \ell : [x \neq 0] \]
\[ \text{out}(\ell) = \{x - y, x \neq 0\} \]
\[ \ell' : b = x - y \]
\[ \text{out}(\ell') = \{x - y\} \]
\[ i = x - y \]
\[ [i < n] \]
\[ [\neg(i < n)] \]
\[ c = x - y \]
\[ A_0 = \{\} \]
\[ A_1 = \{x - y\} \]
\[ A_2 = \{x = 0, x - y\} \]
\[ A_3 = \{\} \]
\[ A_4 = \{x - y\} \]
\[ A_5 = \{x - y\} \]
\[ A_6 = \{x - y, \neg(i < n)\} \]
\[ A_7 = \{x - y, i < n\} \]
\[ A_8 = \{x - y, \neg(i < n)\} \]
\[ A_9 = \{x - y, i < n\} \]
\[ A_{10} = \{x - y\} \]
\[ i = i + c \]
Exercise

Compute the set of available expressions at each point of the program

\[
A_0 = \{\} \\
A_1 = \{x - y\} \\
A_2 = \{x = 0, x - y\} \\
A_3 = \{\} \\
A_4 = \{x - y\} \\
A_5 = \{x - y\} \\
A_6 = \{x - y, \neg(i < n)\} \\
A_7 = \{x - y, i < n\} \\
A_8 = \{x - y, \neg(i < n)\} \\
A_9 = \{x - y, i < n\} \\
A_{10} = \{x - y\}
\]
Data-flow Equations Comparison

Live Variable Analysis

\[
\text{in-live}(S) = (\text{out-live}(S) \setminus \text{def}(S)) \cup \text{use}(S)
\]
\[
\text{out-live}(S) = \bigcup_{S_i \in \text{succ}(S)} \text{in-live}(S_i)
\]

Available Expression Analysis

\[
\text{out-avail}(S) = (\text{in-avail}(S) \cup \text{gen}(S)) \setminus \text{kill}(S)
\]
\[
\text{in-avail}(S) = \bigcap_{S_i \in \text{pred}(S)} \text{out-avail}(S_i)
\]