Global Value Numbering with SSA

X, Y are *dynamically equivalent* at P if they have the same values whenever control reaches P on execution.

Undecidable  Develop static notion  Congruence

X congruent to Y  X dynamically equivanlent to Y

Go beyond Basic Block

Ex.

```
J := 5
K := 5
```

Conclude J,K congruent

---

Value Graph for Basic Block

```
A := 3
B := 3
C := A + 1
D := B + 1
if (C > 3)...
```

C and D are *congruent:*

have identical operators, and like operands are congruent

Like value-numbering
Why SSA?

\[ \begin{align*}
& J := 5 \\
& K := 5 \\
\rightarrow \text{congruent} \\
\end{align*} \]

\[ \begin{align*}
& J := 6 \\
& K := 7 \\
\rightarrow \text{not congruent} \\
\end{align*} \]

\[ \begin{align*}
& J_1 := 5 \\
& K_1 := 5 \\
\rightarrow \text{not congruent} \\
\end{align*} \]

\[ \begin{align*}
& J_2 := 6 \\
& K_2 := 7 \\
\rightarrow \text{congruent} \\
\end{align*} \]

\[ J_1, K_1 \text{ congruent} \]

\[ J_2, K_2 \text{ not congruent} \]

What about control flow?

\[ \begin{align*}
& I := 5 \\
\rightarrow \text{congruent} \\
\end{align*} \]

\[ \begin{align*}
& J_1 := 5 \\
& K_1 := 5 \\
\rightarrow \text{congruent} \\
\end{align*} \]

\[ B \]

\[ I_1, J_1, K_1 \text{ congruent at } B \text{ if assignments dominate } B \]
Value Graph for SSA

Nodes  Constants, operators, phi-functions

Directed Edges  From use to node where value generated

Labels  Constant, operator, function symbols

Ex.

(l > 29)

J1 := 1  J2 := 2
K1 := 1  K2 := 2
J3 := (J1, J2)  K3 := (K1, K2)

BB # for phi function

Value Graph for SSA (example)
Congruence

A is congruent to node B if

1. A is the same node as B, or
2. A and B are constant nodes, with the same constant value, or
3. A and B are operator nodes, with the same operator, and their like operands are congruent

Vars X and Y are equivalent at P if their nodes are congruent and defining assignments dominate P.

Ex.

$J_1, K_1, L_1$ equivalent
$J_2, K_2, L_2$ equivalent
$J_3, K_3$ equivalent, but not with $L_3$

Get equivalence classes of variables

---

Loop Example

Value graph for $J_2$ is identical

BUT cycles prohibit finding it!
Algorithm Overview

1. Compute SSA.
2. Build value graph for SSA.
3. Optimistically assume all nodes with same label are congruent. Determine congruence of nodes by partitioning algorithm.
4. Check for equivalence.

\textit{Partitioning:} \hspace{1cm} (O(E \log E))

1. Put all nodes with same label in same partition.
2. Two nodes are in same partition at step \( i+1 \), if at step \( i \), they are in the same partition and the destination of their edges are in the same partition.

Taking Control Flow into account
Loop Example

Detecting Congruence:
1. Same initial values
2. Same modifications in loop
3. Same no. of iterations

pred, asst that dominates exit

asst outside loop, bottom of loop

Other Extensions

Incorporate arrays, pointers

Update, Access functions

Take commutativity into account

Ex. a*b same as b*a

Combine with hash-based approach

(Cooper et. al.)