Value Numbering (Global)

**Notation**

Notation: \( x \equiv y \iff VN(x) = VN(y) \)

**Basic Idea**

If \( x_1 \equiv x_2 \) and \( y_1 \equiv y_2 \), then \( x_1 \ op \ y_1 \equiv x_2 \ op \ y_2 \).

Problem: When does congruence hold?

\[
\begin{align*}
a &= b; \\
a &= c; \\
a &= d;
\end{align*}
\]
Value Numbering (Global)

<table>
<thead>
<tr>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notation: $x \equiv y \iff VN(x) = VN(y)$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Basic Idea</th>
</tr>
</thead>
<tbody>
<tr>
<td>If $x_1 \equiv x_2$ and $y_1 \equiv y_2$, then $x_1 \text{ op } y_1 \equiv x_2 \text{ op } y_2$.</td>
</tr>
</tbody>
</table>

Problem: When does congruence hold?

a = b;
a = c;
a = d;

Solution: SSA form

$a_1 = b$;
a$_2 = c$;
a$_3 = d$;
\[ a_1 = b; \]
\[ a_2 = c; \]
\[ a_3 = d; \]

**Desired property**

\[ VN(x) = VN(y) \implies x = y \text{ at every program point } p \]

dominated by the (unique) defs of \( x \) and \( y \).
Value Numbering (Global)

**Congruent Expressions**

\[ x \equiv x \]
\[ c_1 \equiv c_2 \]
\[ x_1 \text{ op}_1 x_2 \equiv y_1 \text{ op}_2 y_2 \]
\[ \phi_1(x_1, \ldots, x_n) \equiv \phi_2(y_1, \ldots, y_n) \]

for all variables \( x \).
if \( c_1 \) and \( c_2 \) are constants and \( c_1 = c_2 \).
if \( \text{op}_1 = \text{op}_2 \), \( x_1 \equiv y_1 \), and \( x_2 \equiv y_2 \).
if \( \phi_1 \) and \( \phi_2 \) are in the same basic block, and \( \forall i. x_i \equiv y_i \).
Pessimistic vs. Optimistic Value Numbering

Pessimistic approach
1. Initially assume all expressions are not congruent.
2. Merge sets of expressions determined to be congruent.

Optimistic approach
1. Initially assume all expressions are congruent.
2. Split sets of expressions determined to not be congruent.
Algorithm GVN():

1: for all operators $\text{op}$ (including $\phi$ operators) do
2: create new partition $p$
3: for all assignments $x = y_1 \text{ op } y_2$ do
4: add $x$ to $p$
5: add $p$ to worklist
6: while worklist not empty do
7: remove some partition $p$ from worklist
8: for all $x, q, i$ such that $x \in q$ and $i$’th operand of $x$ in $p$ do
9: if $\exists y \in q$ st $i$’th operand of $y$ not in $p$ then
10: create new partition $r$
11: for all $y \in q$ st $i$’th operand of $y$ not in $p$ do
12: move $y$ from $q$ to $r$
13: add $q$ and $r$ to worklist