263-2810: Advanced Compiler Design

3.3 Dominance frontier

*Thomas R. Gross*

Computer Science Department
ETH Zurich, Switzerland
Outline

- 3.1 Graphs
- 3.2 Approaches to insertion of $\phi$ nodes
- 3.3 Dominance frontier
- 3.4 Algorithm for insertion of $\phi$ nodes
- 3.5 Algorithm for variable renaming
- 3.6 Example
3.3.1 Introduction

- Control flow graph summarizes program.
- Dominance frontier computed for control flow graph.
  - Deal with definitions of a variable in the control flow node
  - Definition d in node Q → insert φ function in node N in dominance frontier of Q.

- Given a node Q in the control flow graph. The dominance frontier DF(Q) is the set of nodes \{ N_k \} such that
  - Q \not\in dom N_k
  - At least one predecessor P of N_k is dominated by Q: Q dom P
  - DF(Q) = \{ N \mid \exists \text{ path from Q to N, } \neg(Q \supset N), \exists \text{ predecessor P of N s.t. } Q \supset P \}
Dominance frontiers

\begin{center}
\begin{tikzpicture}
  \node [draw, shape=rectangle, fill=blue!20] (Q) at (0,0) {Q};
  \node [draw, shape=rectangle, fill=blue!20] (B) at (1,0) {B};
  \node [draw, shape=rectangle, fill=green!20] (N1) at (0,-1) {$N_1$};
  \draw [->] (Q) -- (N1);
  \draw [->] (N1) -- (B);
\end{tikzpicture}
\end{center}
Dominance frontiers

![Diagram of dominance frontiers with nodes Q, B, N, and N1, showing the relationships between them.](image)
DF(Q) = \{ N \mid \exists \text{ path from } Q \text{ to } N, \neg(Q \nrightarrow N), \exists \text{ predecessor } P \text{ of } N \text{ s.t. } Q \nrightarrow P \}\}

DF(X), DF(Y) \text{ may overlap}
3.3.2 Computing DF(X)

- How can we compute DF(X) for a node in the CFG?

- Consider a node Y that is a child of X and a (direct) successor of X
3.3.2 Computing DF(X)

- Consider a node Y that is a child of X in the DT and a (direct) successor of X in the CFG.
- \( \text{DF}(Y) = \{ Z_1, Z_2 \} \), \( \text{DF}(X) = \{ Z_1, Z_2 \} \) and \( Y \) child of \( X \)

- Are all nodes \( N \in \text{DF}(Y) \) also \( \in \text{DF}(X) \)?
- $DF(Y) = \{Z_1, Z_2\}$, $DF(X) = \{Z_1, Z_2\}$ and $Y$ child of $X$
- Are all nodes $N \in DF(Y)$ also $\in DF(X)$?
DF(child) → DF(parent)

- A node $N \in DF(child)$ is $\in DF(parent)$ if $\neg (parent \gg N)$
- $DF(Y) = \{Z_1, Z_2\}$, $DF(X) = \{Z_1\}$

- $DF_{UP}(Y) = \{N \mid N \in DF(Y) \text{ and } \neg(Parent(Y) \gg N)\}$
3.3.2 Computing $DF(X)$

- $DF(Y) = \{Z_1, Z_2\}$
- $DF_{UP}(Y) = \{Z_1, Z_2\}$
- $\text{DF}(Y) = \{ Z_1, Z_2 \}$
- $\text{DF}_{\text{UP}}(Y) = \{ Z_1 \}$
DF(X)

- Node X with children $Y_1$, $Y_2$, ...
- DF(X) contains those nodes from DF($Y_1$), DF($Y_2$), ... that are *not* dominated by X
- DF(X) contains the nodes from DF$_{UP}$(Y$_1$), DF$_{UP}$(Y$_2$), ...

\[ DF(X) = \bigcup \left( DF_{UP}(Y_i) \text{ with } Y_i \text{ child of } X \right) \bigcup \text{(other_nodes)} \]
Dominance frontier

$DF(X) = \{ N \mid \exists \text{ path from } X \text{ to } N, \neg(X \gg N), \exists \text{ predecessor } P \text{ of } N \text{ s.t. } X \gg P \}$

Direct successors of $X$ that are not dominated (trivially) belong to $DF(X)$
$\text{DF}_\text{Local}(X)$

- $\text{DF}_\text{Local}(X)$ is the set of nodes $N$ that are direct successors of $X$ but not descendants

- $\text{DF}_\text{Local}(Y) = \{ Z_1, Z_2 \}$

- $\text{DF}_\text{Local}(X) = \{ Z_3 \}$
Incremental computation of $DF(X)$

- $DF(X) = DF_{Local}(X) \cup ( \bigcup (DF_{UP}(Y_i) \text{ with } Y_i \text{ child of } X))$

- **Proof**
  - $\subseteq$
  - $\supseteq$
Incremental computation of $DF(X)$

- $DF(X) = DF_{Local}(X) \cup \bigcup (DF_{UP}(Y_i) \text{ with } Y_i \text{ child of } X)$

- $DF_{Local}(X)$: Set of nodes in $DF(X)$ determined by (CFG) successors

- $DF_{UP}(Y)$: Set of nodes in $DF(X)$ contributed by children $Y$ of $X$
Efficient computation

Compute $\text{DF}_{\text{Local}}(X)$: inspect all direct CFG successors
  - Usually a small number

Assume $\text{DF}(Y)$ has been computed for all children $Y_i$ of $X$

Get $\text{DF}_{\text{up}}(Y_i)$: all nodes $N \in \text{DF}(Y_i)$ with $\neg (X \gg N)$
Only direct successors contribute to $DF_{Local}(X)$

- Descendants $D_i$ (that are children) contribute through $DF_{UP}(D_i)$
- Efficient computation
- Compute $DF_{Local}(X)$: inspect all direct CFG successors
  - Usually a small number
- Assume $DF(Y)$ has been computed for all children $Y_i$ of $X$
- Get $DF_{up}(Y_i)$: all nodes $N \in DF(Y_i)$ with $\neg (X \gg N)$
Computing $DF_{\text{Local}}(X)$

$$DF_{\text{Local}}(X) = \{ N \mid \text{N is a successor of } X,$$

$$\neg (X \gg N)$$

$$\exists N' \text{ s. t. } X \gg N' \text{ and } N' \text{ predecessor of } N \}$$

- Really easy if $X$ is a leaf node of the DT
  - No children in dominator tree $\implies$ only direct successors must be inspected
Computing $DF_{Local}(X)$

$DF_{Local}(2) = \{ \}$
DF for leafs

- $\text{DF} (X) = \text{DF}_{\text{Local}}(X) \cup \bigcup (\text{DF}_{\text{UP}}(Y_i) \text{ with } Y_i \text{ child of } X))$
- Leaf nodes have no children
- X leaf node: $\text{DF} (X) = \text{DF}_{\text{Local}}(X)$
Computing $DF_{\text{Local}}(X)$

$DF_{\text{Local}}(2) = \{ 3 \}$

$DF(2) = \{ 3 \}$