CS738 - Compiler Optimizations

Assignment 3

Shruti Sharma

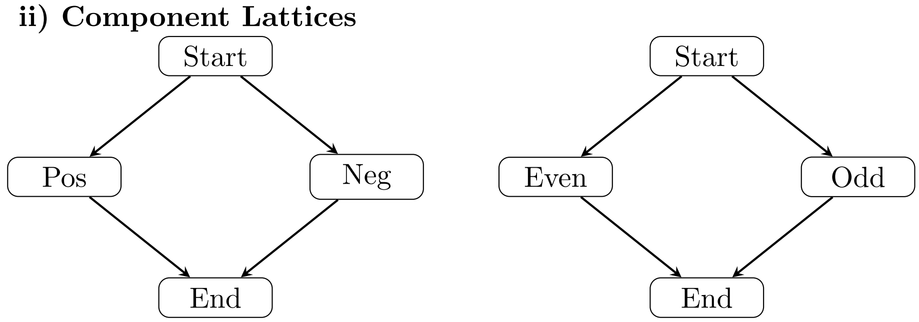
Oct 2020

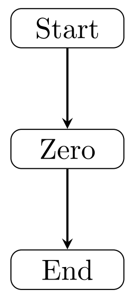
A1) Design of Data Flow Analysis

- Operations - addition(+) , multiplication(*)
- Variables - Integer
- Categories - {start, pos even, pos odd, neg even, neg odd, zero, end} , where start means nothing is known about the variable and end means variable's category is defined.

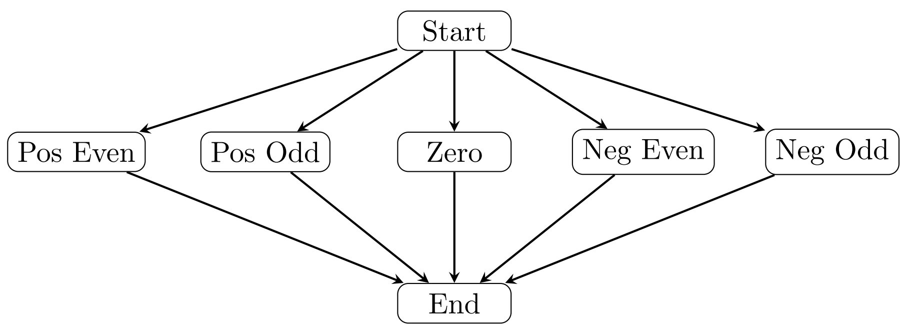
i) Direction of Analysis - *Forward*

ii) Component Lattices





Product lattice



iii) Transfer Functions

1. variable=constant eg. S: x = a

   OUT[S] = {x$\rightarrow$ category(a)}
   where, category(a) = {pos even, pos odd, neg even, neg odd, zero}

2. variable = variable eg. S: x=y

   OUT[S] = {x$\rightarrow$ category(IN[y])}
   where, category(IN[y]) = {start, pos even, pos odd, neg even, neg odd, zero, end}

3. S: x=y+z

   OUT[S] = {x$\rightarrow$ calc_+(y+z)}
   where, calc_+ is

calc$_+$							
y/z	start	pos even	pos odd	neg even	neg odd	zero	end
start	start	start	start	start	start	start	end
pos even	start	pos even	pos odd	end	end	pos even	end
pos odd	start	pos odd	pos even	end	end	pos odd	end
neg even	start	end	end	neg even	neg odd	neg even	end
neg odd	start	end	end	neg odd	neg even	neg odd	end
zero	start	pos even	pos odd	neg even	neg odd	zero	end
end	end	end	end	end	end	end	end

4. S: x=y*z

   OUT[S] = {x→ calc$_*$(y*z)}
   where, calc$_*$ is

calc$_*$							
y/z	start	pos even	pos odd	neg even	neg odd	zero	end
start	start	start	start	start	start	start	end
pos even	start	pos even	pos even	neg even	neg even	zero	end
pos odd	start	pos even	pos odd	neg even	neg odd	zero	end
neg even	start	end	end	neg even	neg odd	zero	end
neg odd	start	end	end	neg odd	neg even	zero	end
zero	start	zero	zero	zero	zero	zero	end
end	end	end	end	end	end	end	end

iv) Meet operation - Meet operation can be seen from component lattices. Like, Start $\wedge$ X = X , End $\wedge$ X = End

v) Mononicity of flow functions - Yes, they are monotonic except addition.

- x=a, x=y are constant transfer functions which are always monotonic.
- x=y*z can be proved monotonic by fixing z to be one thing from category defined above and varying y over entire category one by one. Then, show that y*z remain non-increasing for the opertion *. Do this for all z choices. Similarly, fix y and vary z.
  The same can be seen from above tables. Moving left to right, it is non-increasing. Same is the case on moving top to bottom.
- Although for addtion, this isn't the case so it is non-monotonic.

vi) Top and Bottom elements

- Top element = Start
- Bottom element = End

A2) Back Edges in a Flow Graph

Proof - Consider a back edge x→y. Therefore, y dominates x. So, a path from *Entry* to x goes through y. In the DFST, y will come before x. So, DF order of y will be lower than DF order of x. Thus, x→y is a retreating edge since only retreating edges go from high to low in DF order.

A3) Natural Loops in a Flow Graph

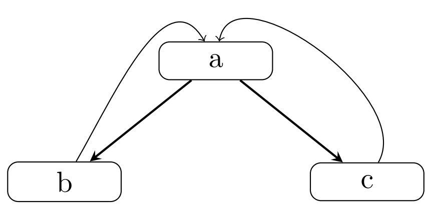
a) Let $a_1 \rightsquigarrow a_3$ be a path that goes from a_1 to a_3 without going through a_2.
Let $a_1 \rightsquigarrow a_2 \rightsquigarrow a_3$ be a path that goes from a_1 to a_3 through a_2.

Proof : Assume that natural loops for back edges $a_1 \rightarrow b_1$ (let, L1) and $a_2 \rightarrow b_2$ (let, L2) intersect (not disjoint) and are neither identical nor nested. Therefore, there is a node n that belongs to both L1 and L2, a node n1 that belongs to L1 but not L2, and a node n2 that belongs to L2 but not L1.

Therefore both b_1 and b_2 dominate n because if node n is in natural loop of a back edge $a_1 \rightarrow b_1$ then b_1 dominates n. Thus, either b_1 dominates b_2 or b_2 dominates b_1. Let's assume b_1 dominates b_2. Clearly, b_2 can not dominate b_1 so b_1 does not belong to L2 implying that paths from b_1 to a_2, if exists, must go through b_2.

Now consider a path, $n_2 \rightsquigarrow a_2 \rightarrow b_2 \rightsquigarrow n \rightsquigarrow a_1$ such a path must exist, and does not involve b_1. But then, n_2 belongs to L1 - a contradiction.

b) Example for same header natural loops



A4) Knaster-Tarski Fixed Point Theorem

Let $Z \subseteq \text{fix}(f)$.
To show : lub(Z) and glb(Z) are in fix(f). Therefore, show that $\text{lub}(Z) \in \text{fix}(f)$.

Consider $y = \wedge Z$ and $X = \{x \mid x \in S, x \leq y\}$.

Since, $(S, \wedge, \vee)$ is a complete lattice, therefore $y \in S$.
$y \leq y \Rightarrow y \in X$
Therefore, y is lub of $(X, \wedge, \vee)$.
Similarly, $\text{glb}(S, \wedge, \vee) = \text{glb}(X, \wedge, \vee)$.
Also, $X \subseteq S$

Therefore, $(X, \wedge, \vee)$ is a complete lattice.

To show : $f{:}X \rightarrow X$.
Consider $x \in X$ and $z \in Z$.
Since, $y = \wedge Z$, therefore, $x \leq y \leq z$
So, $f(x) \leq f(z)$
Since, z is fixed point of f, so, $f(z) = z$.
So, $f(x) \leq z$
Since, $z \in Z$, and $y = \wedge Z$, so, $f(x) \leq y$
Since, $f(x) \in S \Rightarrow f(x) \in X$
Therefore, $x \in X \Rightarrow f(x) \in X$.
Or, $f{:}X \rightarrow X$.

$(X, \wedge, \vee)$ is a complete lattice. $f{:}X \rightarrow X$ is monotonic on X. So, by Knaster Tarski Theorem, f has greatest fixed point z in X. And, $z \in \text{fix}(f)$, $z \leq y$, $y = \wedge Z$, therefore, $\text{lub}(Z) = z \in \text{fix}(f)$