Compiler Optimizations and Program Analysis

Unnikrishan C

August 19, 2019

Outline

Recap

- Static Analysis gives overapproximation of the program properties at each program point.

Recap

- Static Analysis gives overapproximation of the program properties at each program point.
- We saw an example of identifying whether variable are odd or even at a program point.

Recap

- Static Analysis gives overapproximation of the program properties at each program point.
- We saw an example of identifying whether variable are odd or even at a program point.
- The ideal value was not obtained. Values obtained were *over approximated*.

Recap

- Static Analysis gives overapproximation of the program properties at each program point.
- We saw an example of identifying whether variable are odd or even at a program point.
- The ideal value was not obtained. Values obtained were *over approximated*.
- Values obtained also depends on lattice taken for analysis.

Powerset of $\{1,2,3\}$- Relation$=\subseteq$

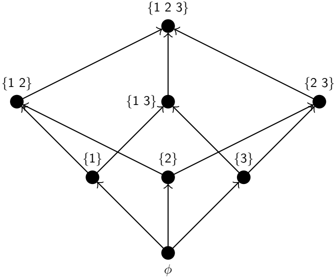
- Some domains are totally ordered $1 \leq 2 \leq 3 \leq ..$ (Set of all Natural Numbers)

Powerset of $\{1,2,3\}$- Relation$=\subseteq$

- Some domains are totally ordered $1 \leq 2 \leq 3 \leq ..$ (Set of all Natural Numbers)
- Some domains are partially ordered: powerset of the set $\{1,2,3\}$, with relation $\subseteq$

Powerset of $\{1,2,3\}$- Relation$=\subseteq$

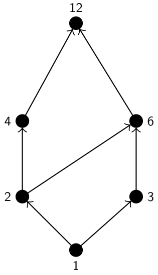
- Some domains are totally ordered $1 \leq 2 \leq 3 \leq ..$ (Set of all Natural Numbers)
- Some domains are partially ordered: powerset of the set $\{1,2,3\}$, with relation $\subseteq$



-

Divisors of 12= {1,2,3,4,6,12} and relation=divides

- $(a \rightarrow b) \Rightarrow$ (b÷ a) ==0.
- Hasse Diagram shown below.



Partially Ordered Set (POSET)

- A `POSET` is a non-empty set S with a partial order $\leq$ on S.

Partially Ordered Set (POSET)

- A `POSET` is a non-empty set S with a partial order $\leq$ on S.
- $\leq$ is a binary relation on S with properties

Partially Ordered Set (POSET)

- A `POSET` is a non-empty set S with a partial order $\leq$ on S.
- $\leq$ is a binary relation on S with properties
  - reflexive: $x \leq x, \forall\, x \in S$.

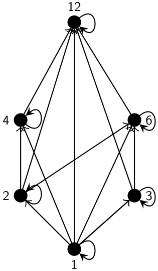
Partially Ordered Set (POSET)

- A `POSET` is a non-empty set S with a partial order $\leq$ on S.
- $\leq$ is a binary relation on S with properties
  - reflexive: $x \leq x, \forall\, x \in S$.
  - transitive: $x \leq y$ and $y \leq z, \Rightarrow x \leq z, \forall\, x, y, z \in S$.

Partially Ordered Set (POSET)

- A `POSET` is a non-empty set S with a partial order $\leq$ on S.
- $\leq$ is a binary relation on S with properties
  - reflexive: $x \leq x, \forall\, x \in S$.
  - transitive: $x \leq y$ and $y \leq z, \Rightarrow x \leq z, \forall\, x, y, z \in S$.
  - antisymmetric: $x \leq y$ and $y \leq x \Rightarrow x{=}{=}y, \forall\, x, y \in S$.

Transitive - S= divisors of 12 and binary relation $\leq$ = divides



- In a POSET (R,$\leq$)

- In a POSET (R,$\leq$)
  - X is a subset of R (X$\subseteq$ R).
  - An element u $\in$ R is an **upper bound** of a set of elements X $\subseteq$ R, if x $\leq$ u for all x $\in$ X .

- In a POSET (R,$\leq$)
  - X is a subset of R (X$\subseteq$ R).
  - An element u $\in$ R is an **upper bound** of a set of elements X $\subseteq$ R, if x $\leq$ u for all x $\in$ X .
  - u is the **least upper bound**,or **lub** or **join**) of X if u is an **upper bound** for X , and for every upper bound y of X , we have u $\leq$ y .

- In a POSET $(R, \leq)$
  - X is a subset of R ($X \subseteq R$).
  - An element $u \in R$ is an **upper bound** of a set of elements $X \subseteq R$, if $x \leq u$ for all $x \in X$.
  - u is the **least upper bound**,or **lub** or **join**) of X if u is an **upper bound** for X , and for every upper bound y of X , we have $u \leq y$.
  - We write $u = \sqcup X$.

- In a POSET $(R, \leq)$
  - X is a subset of R ($X \subseteq R$).
  - An element $u \in R$ is an **upper bound** of a set of elements $X \subseteq R$, if $x \leq u$ for all $x \in X$.
  - u is the **least upper bound**,or **lub** or **join**) of X if u is an **upper bound** for X , and for every upper bound y of X , we have $u \leq y$.
  - We write $u = \sqcup X$.
  - Similarly, $v = \sqcap X$ is the **greatest lower bound** or **glb** or **meet** of X .

Lattices

- A lattice is a partially order set in which every pair of elements has an **lub** and a **glb**

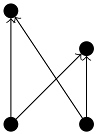
Lattices

- A lattice is a partially order set in which every pair of elements has an **lub** and a **glb**
- A **complete lattice** is a lattice in which every subset of elements has a **lub** and **glb**.

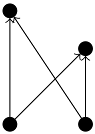
Lattices

- A lattice is a partially order set in which every pair of elements has an **lub** and a **glb**
- A **complete lattice** is a lattice in which every subset of elements has a **lub** and **glb**.
- What is *semilattice*?. Read Dragon Book, covered in the previous class.

The below relation is Lattice, POSET, both or None?



The below relation is Lattice, POSET, both or None?



- It is `POSET`, not a `lattice`.

LLVM- Some inputs

- Generate IR in binary format: **clang –emit-llvm file.c -o file.bc**

LLVM- Some inputs

- Generate IR in binary format: **clang –emit-llvm file.c -o file.bc**
- Make binary IR to readable format (new): **llvm-dis file.bc** (generates `hello.ll`)

LLVM- Some inputs

- Generate IR in binary format: **clang –emit-llvm file.c -o file.bc**
- Make binary IR to readable format (new): **llvm-dis file.bc** (generates `hello.ll`)
- Open IR using command (new): **vi file.ll**

LLVM- Some inputs

- Generate IR in binary format: **clang –emit-llvm file.c -o file.bc**
- Make binary IR to readable format (new): **llvm-dis file.bc** (generates `hello.ll`)
- Open IR using command (new): **vi file.ll**
- The LLVM IR pass discussed in one of the previous class is added in course repository.