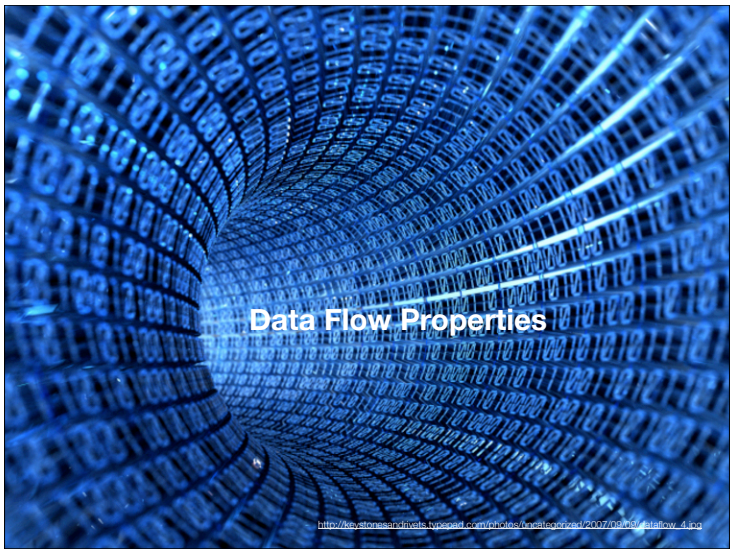
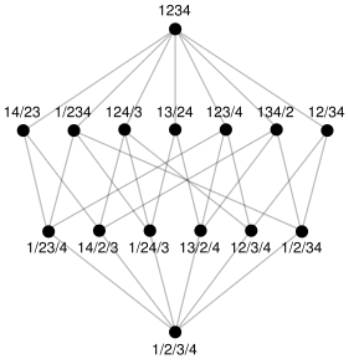
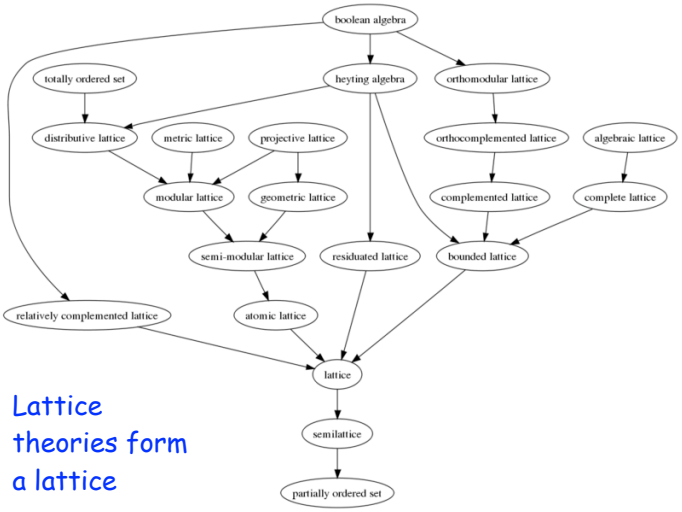


Examples

- Math
  - natural numbers related by divisibility
  - subsets related by inclusion
  - functions
  - nodes in a directed acyclic graph
- Others
  - intervals, Cartesian squares
  - ancestry relation among family members



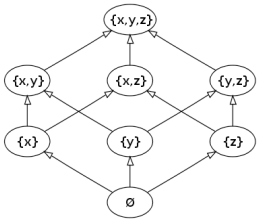
http://en.wikipedia.org/wiki/Lattice_(order)



Lattice theories form a lattice

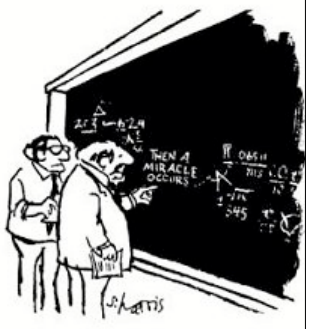
Lattice

- A semi-lattice
  - $\wedge$: idempotent, commutative, and associative
    - $a \wedge a = a$
    - $a \wedge b = b \wedge a$
    - $a \wedge (b \wedge c) = (a \wedge b) \wedge c$
  - a partially ordered set (poset)
    - every pair has a greatest lower bound and a lowest upper bound
    - $a \geq b \Leftrightarrow a \wedge b = b$
    - $a > b \Leftrightarrow a \geq b$ and $a \neq b$
- A power set forms a semi-lattice
  - under union or intersection



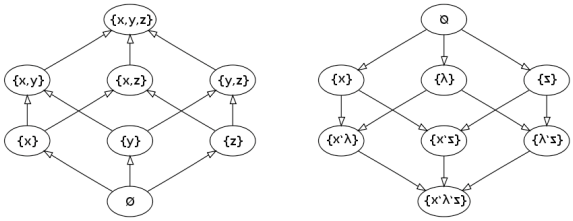
Properties of Partial Order

- Reflexive
  - $a \leq a$
- Antisymmetric
  - if $a \leq b$ and $b \leq a$ then $a = b$
- Transitive
  - if $a \leq b$ and $b \leq c$ then $a \leq c$

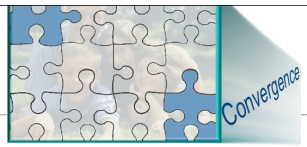


http://www.nkschools.org/1599208151462354 3/lib/1599208151462354 3/Miracle_proof.jpg

Semi-lattice for Avail and Live



- Remember the initialization for Avail and Live?
- Do you see any relation with the structure of the lattice?



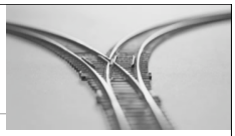
Requirement

- Monotonicity
  (1) $u \leq v \Rightarrow f(u) \leq f(v)$
  (2) $f(u \wedge v) \leq f(u) \wedge f(v)$
- Avail
  - $f(x) = (x \cap c_1) \cup c_2$
- Live
  - $f(x) = (x \cap c_1) \cup c_2$

Monotonicity

Are $u \leq v \Rightarrow f(u) \leq f(v)$ and $f(u \wedge v) \leq f(u) \wedge f(v)$ equivalent?

Monotone Means Convergence



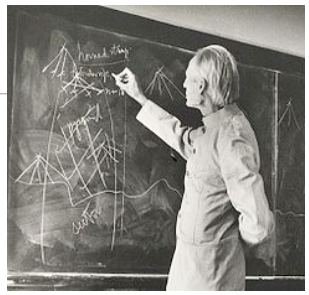
- Kam and Ullman, JACM 1976
- Proof: every step of the iterative algorithm, increment time by 1. $A^t[n]$ is the result of block n at time t.
  Initially, $A^0[n] = T$ and $A^0[n_0] = \perp$
  Induction to prove $A^{m+1}[n] \leq A^m[n]$

Distributive and Rapid Data Flow

- Distributive
  - $f(u \wedge v) = f(u) \wedge f(v)$
  - unique fixed point
- Rapid
  - $f(g(\perp)) > g(\perp) \wedge f(x) \wedge x$
  - converge in $d(G)+3$ iterations

Brouwer Fixed Point Theorem

- A continuous function from and to the same compact convex domain must have a fixed point
- Kakutani's extension to set-valued logic
- Used by Nash to prove the existence of Nash equilibrium



Luitzen Egbertus Jan Brouwer (1881-1966), a Dutch mathematician and philosopher

The 31st Annual
ACM SIGPLAN - SIGACT
Symposium
on
Principles of Programming Languages
Venice, Italy
January 14-16, 2004

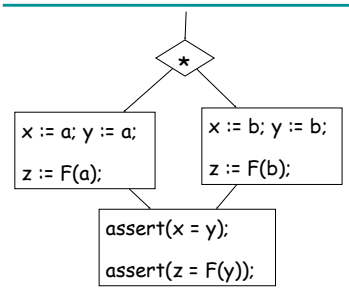
Global Value Numbering using Random Interpretation

Sumit Gulwani George C. Necula
CS Department
University of California, Berkeley

Global Value Numbering

- Problem
  - To detect equivalences of expressions in a program
  - To obtain a complete algorithm under the assumptions:
    - Conditionals are non-deterministic
    - Operators are uninterpreted
      - $F(e_1,e_2) = F'(e_1',e_2')$, $F=F'$, $e_1=e_1'$, $e_2=e_2'$
- Existing algorithms
  - Precise but expensive
  - Efficient but imprecise
- Use randomization to obtain a precise, efficient but probabistically sound algorithm

Example



$x = \phi(a,b)$

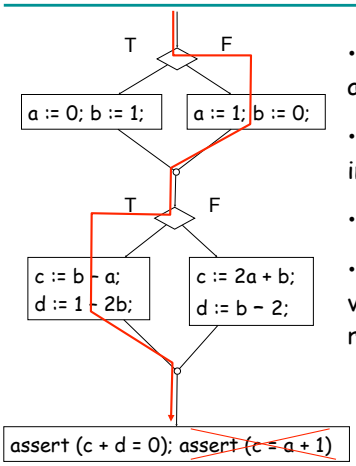
$y = \phi(a,b)$

$z = \phi(F(a),F(b))$

$F(y) = F(\phi(a,b))$

- Typical algorithms treat ϕ as uninterpreted
  - Hence cannot verify the second assertion
- The randomized algorithm interprets ϕ
  - Similar to the randomized algorithm for linear arithmetic

Review: Randomized Algorithm for Linear Arithmetic

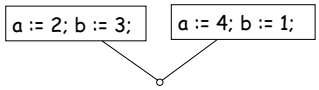


- Between random testing and abstract interpretation
- Choose random values for input variables
- Execute both branches
- Combine the values of a variable at join points using a random affine combination

Review: The Affine Join Operation

- Affine combination of v_1 and v_2 w.r.t. weight w

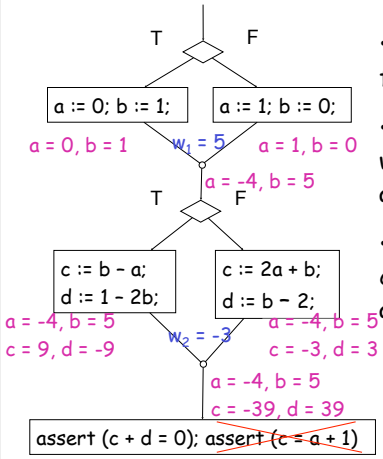
$\phi_w(v_1,v_2) \equiv w\ v_1 + (1-w)\ v_2$



$a = \phi_7(2,4) = -10$

$b = \phi_7(3,1) = 15$ $\quad (w = 7)$

Review: Example



- Choose a random weight for each join independently.
- All choices of random weights verify the first assertion
- Almost all choices contradict the second assertion

Uninterpreted Functions

$e := y \mid F(e_1,e_2)$

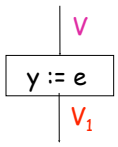
- Choose a random interpretation for F
- Non-linear interpretation
  - E.g. $F(e_1,e_2) = r_1e_1^2 + r_2e_2^2$
  - Preserves all equivalences in straight-line code
  - But not across join points
- Lets try linear interpretation

The Random Interpreter R

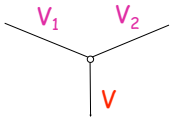
V: Variables ! Vectors

V(e): defined inductively as $V(F(e_1,e_2)) = R_1V(e_1) + R_2V(e_2)$

$V^j(e)$: the j^{th} component of vector V(e)

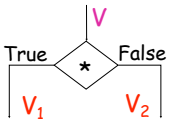


$V_1 = V[y \leftarrow V(e)]$



$V^j(y) = \phi_w(V_1^j(y), V_2^j(y))$

for all y, j



$V_1 = V$

$V_2 = V$

Outline

- Two key ideas in the algorithm
  - The affine join operation
  - K-linear interpretations
- Correctness of the algorithm
- Termination of the algorithm

Completeness and soundness of R

- We compare the random interpreter R with a suitable abstract interpreter A
- R mimics A with high probability
  - R is as complete as A
  - R is (probabilistically) as sound as A

Completeness Theorem

- If S) $e_1 = e_2$, then $V(e_1) = V(e_2)$
- Proof:
  - Uninterpreted operators are modeled as linear functions
  - The affine join operation preserves linear relationships

Soundness Theorem

- If S) $e_1 \neq e_2$, then with high probability $V(e_1) \neq V(e_2)$
- Error probability · $\left(\frac{2n^2}{d}\right)^t$
  - n: number of function applications
  - d: size of set from which random values are chosen
  - t : number of repetitions
- If n = 100, d ¼ 2^{32}, t $\frac{1}{2^{100}}$ then error probability ·

Outline

- Two key ideas in the algorithm
  - The affine join operation
  - K-linear interpretations
- Correctness of the algorithm
- Termination of the algorithm

Loops and Fixed Point Computation

- The lattice of sets of equivalences has finite height n. Thus, the abstract interpreter A converges to a fixed point.
- Thus, the random interpreter R also converges (probabilistically)
- We can detect convergence by comparing the set of symbolic relationships implied by vectors in two successive iterations

Related Work

- Efficient but imprecise algorithms
  - Congruence partitioning [Rosen, Wegman, Zadeck, POPL 88]
  - Rewrite rules [Ruthing, Knoop, Steffen, SAS 99]
    - Balanced algorithms [Gargi PLDI 2002]
- Precise but inefficient algorithms
  - Abstract interpretation on uninterpreted functions [Kildall 73]
- Affine join operation
  - Random interpretation for linear arithmetic [Gulwani, Necula POPL 03]

Conclusion and Future Work

- Key ideas in the paper

$$\phi(e_1,e_2) = w\, e_1 + (1-w)\, e_2$$

  - Linearity , Preserves equivalences across a join point
    - $F(e_1,e_2) = R_1\, e_1 + R_2\, e_2$
  - Vectors) Introduce no false equivalence
- Random interpretation vs. deterministic algorithms
  - Linear arithmetic
    - $O(n^2)$ vs. $O(n^4)$ [POPL 2003]
  - Uninterpreted functions
    - $O(n^3)$ vs. $O(n^5 \log n)$ [this talk]
- Future work
  - Inter-procedural analysis using random interpretation