Numeric Abstract Domains

Mooly Sagiv
http://www.cs.tau.ac.il/~msagiv/courses/pa16.html
Tel Aviv University
640-6706

Adapted from Antoine Mine

Subjects

Goals

- Infer inductive invariants on numeric values
- Abstract sets of points in $P(R^n)$
- Applications:
  - Array bound
  - Termination
    - » infer ranking functions with value in N
  - Cost Analysis
    - » time, memory consumption are numeric quantities
  - Pointer analysis with pointer arithmetic
    - » pointer ≈ offest
  - String analysis in C
    - » Length, index

Numeric Semantics

Arithmetic Expressions & Commands

- <exp> ::= V $V \in Var$|
  | - <exp>
  | <exp> op <exp> $op \in \{+, -, \times, /\}$|
  | [c, c'] $c, c' \in R \cup \{-\infty, \infty\}$
- <com> ::= V := <exp> $V \in Var$
  | assume <exp> relop 0
  | assert <exp> relop 0
  $relop \in \{=, \neq, <, >, \leq, \geq\}$
- Control Flow Graph G(N, E, s) where $E \subseteq N \times N$ is annotated with commands
  - $s \in N$ is the start node

Example Program

1: X := [1, 10] ;

2: Y := 100;

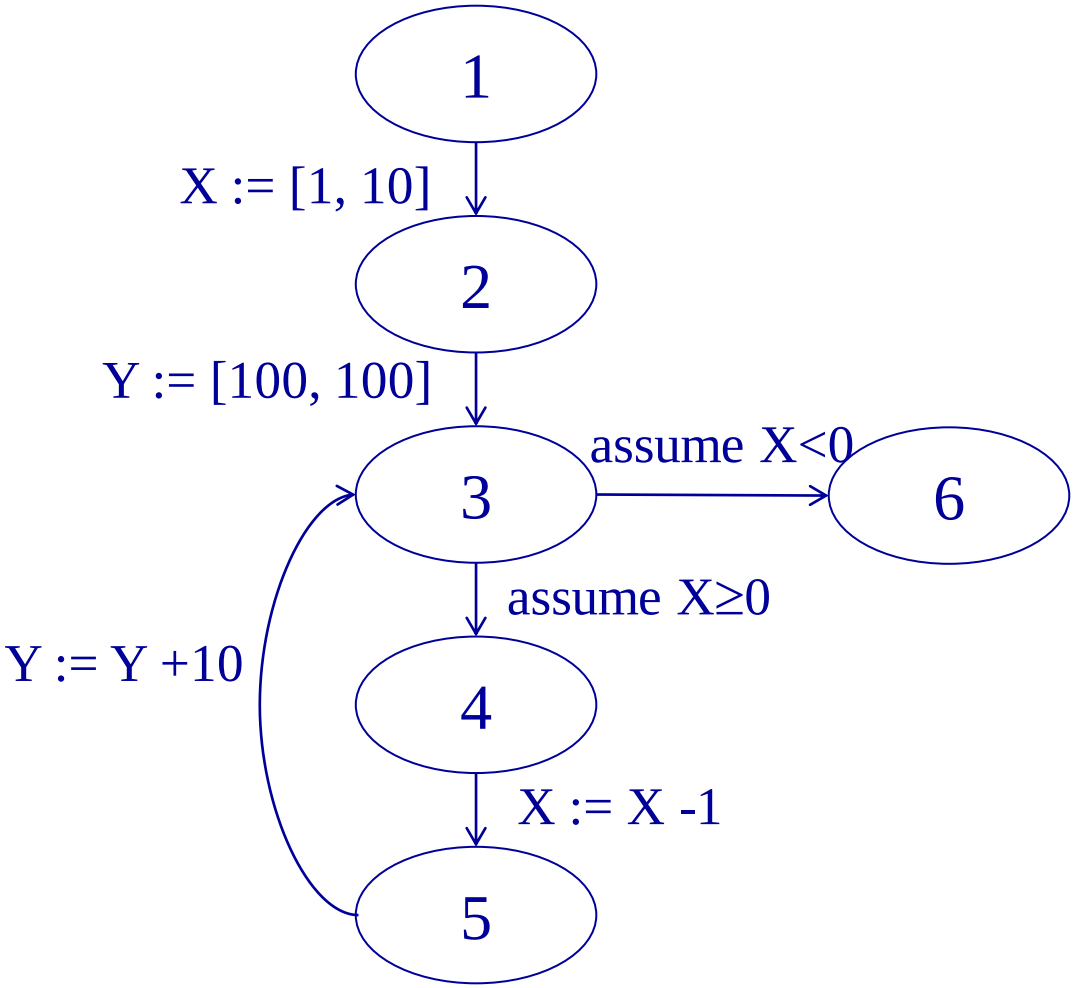
while 3: X>= 0 do {

  4: X := X – 1;

  5: Y := Y + 10

}

6:



Concrete Operational Semantics

Semantics of Expressions

- States $\sigma \in \Sigma = Var \rightarrow R$
- Semantics $E[\![\,]\!]$: <exp> $\rightarrow \Sigma \rightarrow R$
- $E[\![V]\!]\sigma = \sigma\, V$
- $E[\![c, c']\!]\sigma = \{ x \in R \mid c \leq x \leq c' \}$
- $E[\![-\text{<exp>}]\!]\sigma = \{ -x \mid x \in E[\![\text{<exp>}]\!]\sigma \}$
- $E[\![\text{<exp> op <exp'>}]\!]\sigma = \{x \text{ op } x' \mid x \in E[\![\text{<exp>}]\!]\sigma, x' \in E[\![\text{<exp'>}]\!]\sigma\}$

  $\text{op} \in \{+, -, \times\}$
- $E[\![\text{<exp> / <exp'>}]\!]\sigma =$

Semantics of Commands

- States $\sigma \in \Sigma = Var \rightarrow R$
- Semantics $C[\![\,]\!]$: <com> $\rightarrow P(\Sigma) \rightarrow P(\Sigma)$
- $C[\![V := \langle exp \rangle]\!]Z = \{\sigma\,[V \mapsto x] \mid \sigma \in Z,$
  $x \in E[\![\langle exp \rangle]\!]\sigma\ \}$
- $C\,[\![\text{assume } \langle exp \rangle \text{ relop } 0]\!]Z = \{\sigma \mid \sigma \in Z,$
  $\forall x \in E[\![\langle exp \rangle]\!]\sigma : x \text{ relop } 0\ \}$
- $C\,[\![\text{assert } \langle exp \rangle \text{ relop } 0]\!]Z$

Distributivity

- $C[\![exp]\!]$ is distributive
- $C[\![exp]\!](\cup Z) = \cup_{\sigma \in Z} C[\![exp]\!]\{\sigma\}$

Concrete Semantics of Programs

- $[\![G(s, N, E)]\!] : P(\Sigma) \rightarrow N \rightarrow P(\Sigma)$
  - The set of reachable states
  - $D = <P(\Sigma), \subseteq, \cup, \cap, \varnothing, \Sigma)$
- The smallest simultaneous solution to the set of equations $[\![G(s, N, E)]\!]\ \iota$

$$CS_s = \iota$$

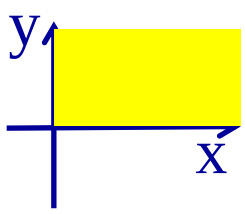
$$CS_n = \cup_{<m, c, n> \in E} C[\![c]\!]\ CS_m \qquad n \neq s$$

- Uniquely defined from Tarski's theorem but not computable

Numeric Abstract Domains

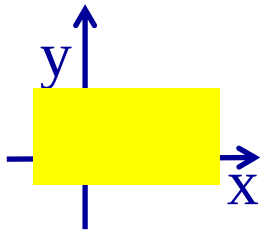
- Representation: a set $D^{\#}$ of representable abstract values
- $<D^{\#}, \sqsubseteq^{\#}, \sqcup^{\#}, \sqcap^{\#}, \bot^{\#}, \top^{\#})$
  - relating the amount of information given by abstract values
- A concretization function
  $\gamma: D^{\#} \rightarrow D = P(\Sigma) = P(Var \rightarrow R)$
- Required algebraic properties:
  - γ need to be monotonic: $d \sqsubseteq^{\#} d' \Rightarrow \gamma\, d \sqsubseteq^{\#} \gamma\, d'$
  - Strictness $\gamma \bot^{\#} = \varnothing$
  - $\gamma \top^{\#} = Var \rightarrow R$
- γ need not be one-to-one

Numeric Abstract Domain Examples



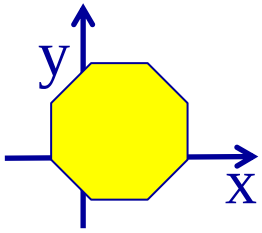
signs

$x \geq 0$



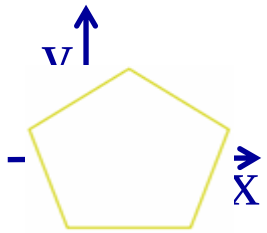
intervals

$x \in [a, b]$



octagons

$\pm x \pm y \leq c$



polyhedra

$\Sigma a_i x_i \leq c$

Requirements on abstract operators

- Algorithmic requirements
  - For each c $\in$ <cmd>, $c^{\#}[\![c]\!]: D^{\#} \rightarrow D^{\#}$ is computable
  - Algorithm for $\sqcup^{\#}$
    - » Used for merging control paths and iterations
  - Algorithm for $\sqcap$
    - » Used for assume
  - Algorithm for $\sqsubseteq^{\#}$
    - » Used for checking termination

Abstract Semantics of Programs

- $[\![G(s, N, E)]\!] : D^{\#} \rightarrow N \rightarrow D^{\#}$
  - The set of reachable abstract states
  - $D^{\#} = <D^{\#}, \sqsubseteq^{\#}, \sqcup^{\#}, \sqcap^{\#}, \bot^{\#}, \top^{\#})$
- The smallest simultaneous solution to the set of equations $[\![G(s, N, E)]\!]^{\#} \iota^{\#}$

$$AS_s = \iota^{\#}$$

$$AS_n = \sqcup^{\#}_{<m, c, n> \in E} C^{\#}[\![c]\!] AS_m \qquad n \neq s$$

- Uniquely defined from Tarski's theorem

Soundness

- The smallest simultaneous solution to the set of equations $[\![G(s, N, E)]\!]\ \iota$
  - CS

$$CS_s = \iota$$

$$CS_n = \bigcup_{<m, c, n> \in E} C[\![c]\!]\ CS_m \qquad n \neq s$$

- Any solution AS set of equations $[\![G(s, N, E)]\!]^{\#}\ \iota^{\#}$

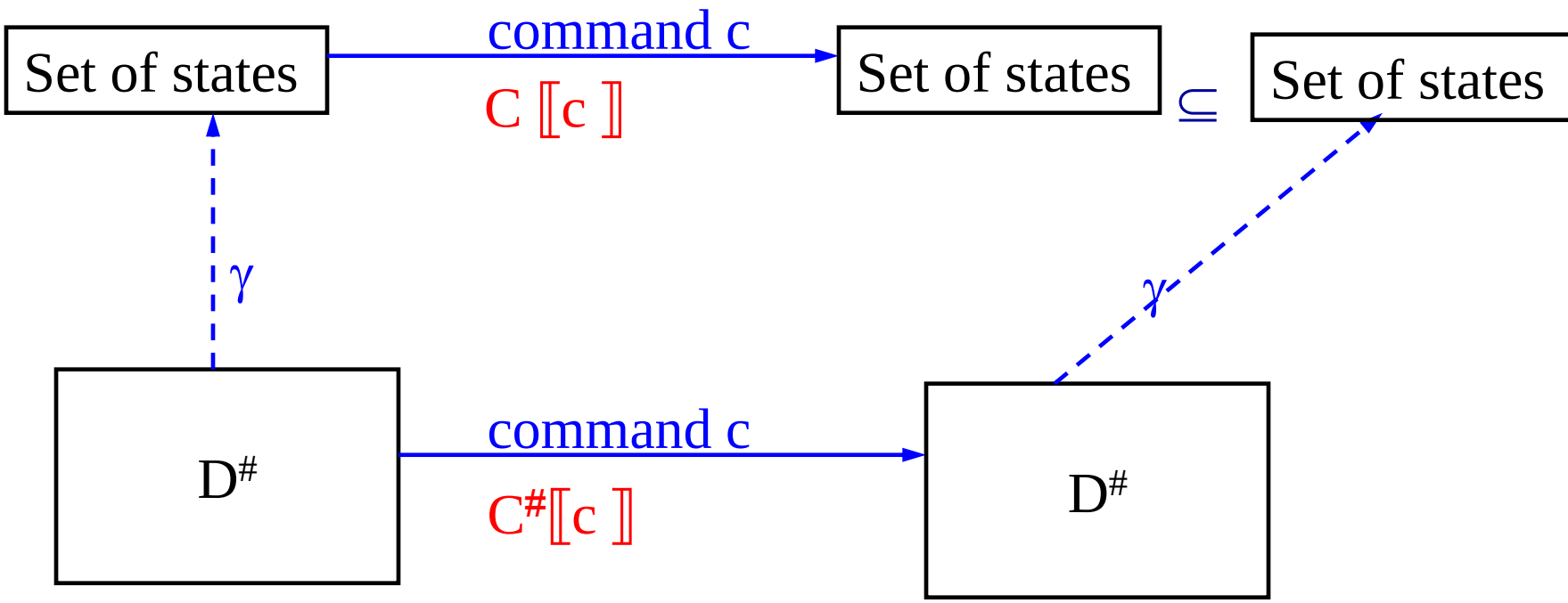
$$AS_s = \iota^{\#}$$

$$AS_n = \bigsqcup^{\#}_{<m, c, n> \in E} C^{\#}[\![c]\!]\ AS_m \qquad n \neq s$$

- $CS_n \subseteq \gamma\ AS_n$ for all $n \in N$

Soundness requirement

- $\iota \subseteq \gamma\ \iota^{\#}$
- For each c $\in$ <cmd>, d $\in$D#, c$[\![c]\!](\gamma\ d) \subseteq \gamma\ (c^{\#}[\![c]\!]\ d)$



Optimality (induced operation)

- Requires existence of abstraction α: D $\rightarrow$ $D^{\#}$ such that $<\alpha, \gamma>$ form a Galois connection
- Define $c[\![c]\!]^{\#} = \lambda d.\ \alpha\ (c[\![c]\!]\ (\gamma\ d)$
- α may not exist
- $c[\![c]\!]^{\#}$ may be hard to compute

Widening

- Accelerate the termination of Chaotic iterations by computing a more conservative solution
- Can handle lattices of infinite heights
- $\nabla : D^{\#} \times D^{\#} \rightarrow D^{\#}$ such that
  - $d \sqcup^{\#} d' \sqsubseteq d \nabla d'$
  - For every increasing chain $d^{\#}_1 \sqsubseteq d^{\#}_2 \sqsubseteq \ldots$,
    - » The sequence $s0 = d^{\#}_0$ and $s_{i+1} = s_i \nabla d^{\#}_i$ is finite

Chaotic Iterations with widening

for each n in N do AS[v] := $\perp^{\#}$

AS[s] = $\iota^{\#}$ WL = {s}

while (WL $\neq$ $\emptyset$) do

select and remove an element m $\in$ WL

for each n, such that. (m, c, n) $\in$E do

temp = $c[\![c]\!]^{\#}$ AS[m]

if m is a loop header

then new := AS(n) $\bigtriangledown$ temp

else new := AS(n) $\sqcup^{\#}$ temp

if (new $\neq$ AS[n]) then

AS[n] := new;

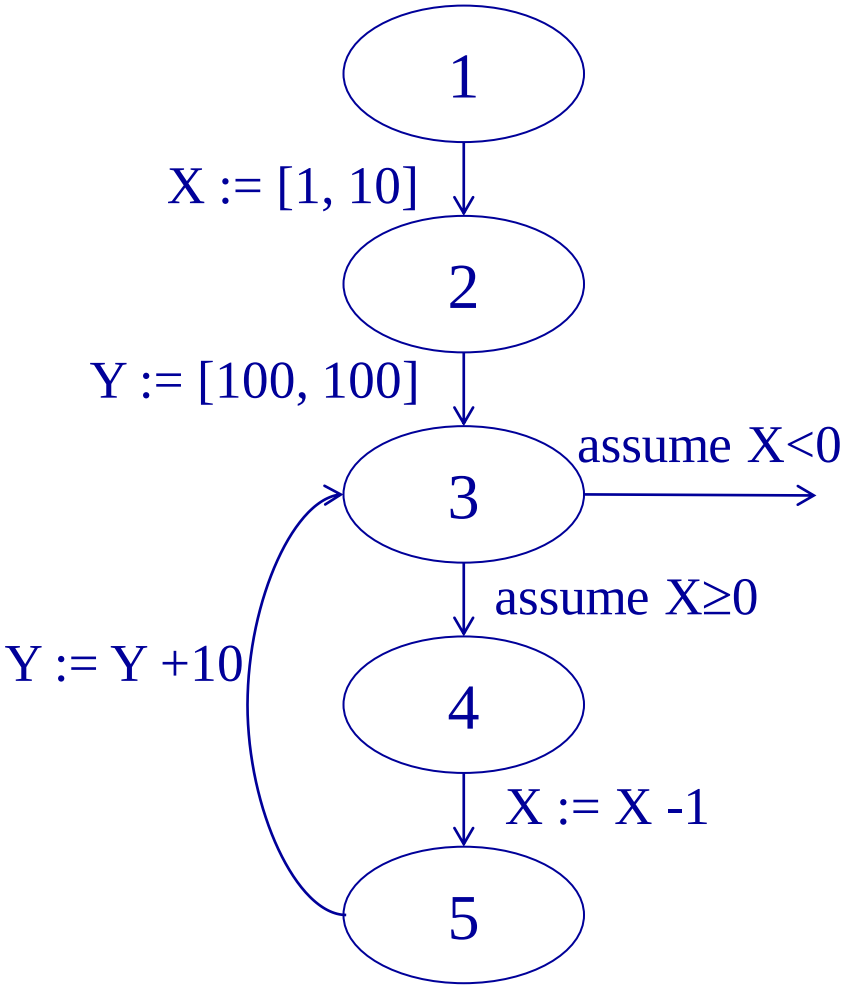
WL := WL $\cup${n}

Non-Relational Abstractions

Cartezian Abstraction (independent attribute)

- Forget the relationship between variables

Example Program



The Interval Domain

The Interval Domain [Moore'66, Cousot'76]

- $D^{\#} = \{[a, b] \mid a \leq b \in R \text{ or } a=-\infty \text{ or } b= \infty \} \cup \perp^{\#}$
- $\top^{\#} = [-\infty, \infty]$
- $d \sqcup^{\#} d' =$ if $d = \perp^{\#}$ then d'
  else if $d' = \perp^{\#}$ then d
  else let $d=[a, b]$ and $d'=[c, d]$ in
  $[\min(a, c), \max(b, d)]$
- $d \sqcap^{\#} d' =$ if $d = \perp^{\#}$ then $\perp^{\#}$
  else if $d' = \perp^{\#}$ then $\perp^{\#}$
  else let $d=[a, b]$ and $d'=[c, d]$ in
  let $l = \max(a, c)$ and $u= \min(b, d)$
  if $l > u$ then $\perp^{\#}$ else $[l, u]$
- $d \nabla d' =$ if $d = \perp^{\#}$ then d'
  else else let $d=[a, b]$ and $d'=[c, d]$ in
  $[$if $a \leq c$ then a else $-\infty$,
  if $b \geq d$ then b else $\infty]$

Galois Connection

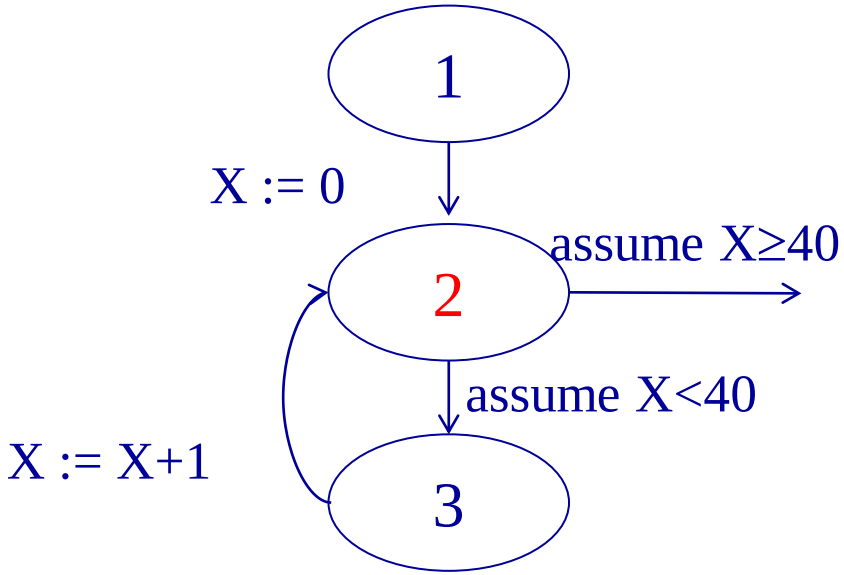
Abstract Expressions

Abstract Assignments

Optimality (Induced)

Abstract Assume

Example Program



Relational Domains

The need for relational domains

- Non-relation domains cannot represent variable relationships

```
Y :=0;
 while true do {
    X:=[-128,128]; D:=[0,16];
    S:=Y; Y:=X; R:=X-S;
     if R<=-D then Y:=S-D fi;
     if R>=D then Y:=S+D fi
}
```

X: input signal
Y: output signal
S: last output
R: Y-S
D: max allowed for |R|

The need for relational domains

- Infer strong enough inductive invariants

```
X:=0; I:=1;
while  I<5000 do {
   if  … then X:=X+1 else X:=X-1 fi;
   I:=I+1
}
```

The need for relational domains

- Modular analysis of procedures

```
Z :=X ;
if Y > Z then Z :=Y ;
if Z < 0 then Z :=0;
```

Weakly Relational Domains

The Zone Domain [Shacham'00, Mine'01]

Constrains of the form

$V_i - V_j \leq c$

$\pm V_i \leq c$

Machine Representation

- A potential constraint has the form $V_i - V_j \leq c$
- Represented as a directed graph G
  - Nodes are labeled with variables
  - An arc with weight c from V_i to V_j for each constraint $V_i - V_j \leq c$
- Difference Bound Matrix (DBM)
  - Adjacency matrix m of G
  - $mij = c < \infty \rightarrow V_i - V_j \leq c$
  - $mij = \infty \rightarrow$ No such constraints
- Concretization

Machine Representation (cont)

- Unary constraints
  - Add another variable V_0
  - m has size n+1 ×n+1
  - $V_i \leq c$ is denoted as $V_i - V_0 \leq c$, i.e., $m_{i,0} = c$
  - $V_i \geq c$ is denoted as $V_0 - V_i \leq -c$, i.e., $m_{0,i} = -c$
  - $\underline{\gamma}\, m = \{ (v_1, v_2, \ldots, v_n) \mid (0, v_1, v_2, \ldots, v_n) \in \gamma\, m\}$

	V0	V1	V2
V0	+∞	4	3
V1	-1	+∞	+∞
V2	-1	1	+∞

The DBM Lattice

Relational Domains

The Polyhedra Domain [CH'78]

$$\wedge_i \Sigma_j a_{i,j} x_{i,j} \geq c_i$$

Summary

- Numerical Domains are Powerful
- Infer interesting invariants
- Cost is an issue
- Need to combine with other domains
- Next week some applications