Program verification 2016

Lecture 4: Abstract interpretation

Instructor: Ashutosh Gupta

TIFR, India

Compile date: 2016-04-11

Where are we?

- Abstraction for fixed point computation
- Lattice theory and Galios connection
- Conditions for effective fixed point computation

Lecture plan

- Abstract interpretation
- Abstract domains
- Widening/Narrowing
- Box domain
- Polyhedral domain
- Octagonal domain

Topic 4.1

Galois connection

Galois connection

Definition 4.1

For posets $(X, \leq)$ and $(Y, \sqsubseteq)$, a pair of maps (α, γ) of maps $\alpha : X \to Y$ and $\gamma : Y \to X$ is a galois connection *if*

$$\forall x \in X \forall y \in Y.\ \alpha(x) \sqsubseteq y \Leftrightarrow x \leq \gamma(y)$$

which is usually written

$$(X, \leq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (Y, \sqsubseteq)$$

α and γ are called upper and lower adjoints respectively.

Unique adjoints

Theorem 4.1

In $(X, \leq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (Y, \sqsubseteq)$*,* α *uniquely defines* γ *and vice-versa.*

$$\alpha(x) = \sqcap\{y | x \leq \gamma(y)\} \qquad \gamma(y) = \vee\{x | \alpha(x) \sqsubseteq y\}$$

Proof.

- By definition of meet, $\sqcap\{y|\alpha(x) \sqsubseteq y\}$ exists and

$$\alpha(x) = \sqcap\{y|\alpha(x) \sqsubseteq y\}$$

- By def. of galois connection

$$\alpha(x) = \sqcap\{y|x \leq \gamma(y)\}$$

Symmetrically for $\gamma(x)$. □

Properties of galois connection

Let $(X, \leq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (Y, \sqsubseteq)$ then

1. $\forall x \in X . x \leq \gamma \circ \alpha(x)$
2. $\forall y \in Y . \alpha \circ \gamma(y) \sqsubseteq y$
3. α is monotone
4. γ is monotone
5. $\alpha \circ \gamma \circ \alpha = \alpha$
6. $\gamma \circ \alpha \circ \gamma = \gamma$
7. α is onto $\Leftrightarrow$ γ is one-to-one $\Leftrightarrow$ $\alpha \circ \gamma = 1_X$
8. γ is onto $\Leftrightarrow$ α is one-to-one $\Leftrightarrow$ $\gamma \circ \alpha = 1_Y$

Exercise 4.1

Prove the above properties

Exercise 4.2

Prove properties 1-4 also define galois connection

Topic 4.2

Abstract interpretation

Abstract interpretation

- Concrete objects of analysis or domain — C = sets of valuations $\subseteq \mathbb{Q}^V$
  - not all sets are concisely representable in computer
  - too (infinitely) many of them
- Abstract domain — only simple to represent sets $D \subseteq C$
  - D should allow efficient algorithms for desired operations
  - far fewer, but possibly infinitely many
  - Sets in $C \setminus D$ are not precisely representable in D

How to use D to capture semantics of a program?

Note: C naturally forms a complete lattice

$$(C, \subseteq, \emptyset, \mathbb{Q}^V, \cup, \cap)$$

Abstracting and concretization function

This is not the most general definition!
Any partial order can replace $\supseteq$.

Definition 4.2

An abstraction function $\alpha : C \rightarrow D$ *maps each set* $c \in C$ *to* $\alpha(c) \supseteq c$.

Definition 4.3

A concretization function $\gamma : D \rightarrow C$ *maps each set* $d \in D$ *to* d.

The above definitions become more meaningful, if we think of D as the representation of sets on a computer instead of the sets themselves.

Lemma 4.1

D contains $\mathbb{Q}^V$

Example: abstraction – intervals

Example 4.1

Let us assume $V = \{\mathtt{x}\}$

Consider $D = \{\bot, \top\} \cup \{[a, b] | a, b \in \mathbb{Q}\}$.

D forms a lattice.

Ordering among elements of D are defined as follows:
$\bot \sqsubseteq [a, b] \sqsubseteq \top$ *and* $[a_1, b_1] \sqsubseteq [a_2, b_2] \Leftrightarrow a_2 \leq a_1 \wedge b_1 \leq b_2$

Let $\alpha(c) \triangleq [inf(c), sup(c)]$ *and* $\gamma([a, b]) \triangleq [a, b]$

- $\alpha(\{0, 3, 5\}) = [0, 5]$
- $\alpha((0, 3)) = [0, 3]$
- $\alpha([0, 3] \cup [5, 6]) = [0, 6]$
- $\alpha(\{1/x | x \geq 1\}) = [0, 1]$

Minimal abstraction principle

It is always better to choose smaller abstraction.

Choose $\alpha(c)$ as small as possible, therefore more precise abstraction

Therefore, if $d \in D$ then $\alpha(d) = d$ and α must be monotonic

There may be multiple minimal abstractions.

Even worse, there may be no minimal approximation,
e. g., approximating a circle with a polytope
(In this lecture, we assume minimal abstractions exist.)

Properties of D, α, and γ

Now on we will ignore that D is set of sets. We assume D is a topped poset

$$(D, \sqsubseteq, \top)$$

- α is monotone (due to minimality principle)
- γ is monotone
- $c \subseteq \gamma \circ \alpha(c)$
- $\alpha \circ \gamma(d) \sqsubseteq d$ (due to minimality principle)

Therefore,

$$(C, \subseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (D, \sqsubseteq)$$

We always choose D, α, and γ such that the above galois connection holds.

Best approximation

Definition 4.4

α performs best approximation if $\forall c \in C, d \in D.\ c \subseteq \gamma(d) \Rightarrow \alpha(c) \sqsubseteq d$.

The above is one of the galois conditions. So, $\alpha(c) = \sqcap\{d \in D | c \subseteq \gamma(d)\}$.

Theorem 4.2

An abstract domain is complete lattice iff best approximations exists.

Proof.

If abstract domain is complete lattice then $\sqcap\{d \in D | c \subseteq \gamma(d)\}$ always exists.
For the other direction, consider $S \subseteq D$.

1. Since $\bigcap\gamma(S)$ and best approximations exists, $\alpha(\bigcap\gamma(S)) = \sqcap\{d | \bigcap\gamma(S) \subseteq \gamma(d)\}$
2. $(\forall c \in S.\ c \in \{d | \bigcap\gamma(S) \subseteq \gamma(d)\}) \Rightarrow \alpha(\bigcap\gamma(S)) \in lb(S)$
3. Assume $d \in lb(S)$. Due to monotone γ, $\gamma(d) \in lb(\gamma(S))$. Therefore, $\gamma(d) \subseteq \bigcap\gamma(S)$
4. Due to monotone α, $\alpha \circ \gamma(d) \sqsubseteq \alpha(\bigcap\gamma(S))$
5. Since $\alpha \circ \gamma = 1_D$, $d \sqsubseteq \alpha(\bigcap\gamma(S))$. Therefore, $\alpha(\bigcap\gamma(S)) = \sqcap S$ □

Note: If we do not have best approximation then we are breaking conditions of galois connection, namely monotone α.

Onto abstraction

Due to the principle of minimal abstraction, α must be onto

$$\forall p \in D.\ \alpha(p) = p \qquad (\text{assuming } D \subseteq C)$$

Therefore, one-to-one γ

However, in practice we may relax the onto condition on α. A set can be represented multiple ways on a computer.

Therefore, multiple abstract objects may have same concretization.

Exercise 4.3

Make an exercise

Topic 4.3

Examples of abstraction

Sign abstraction

Sign abstraction

$C = \mathfrak{p}(\mathbb{Q})$

$D = \{+, -, 0, \bot, \top\}$

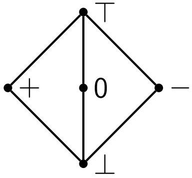
$\alpha(p) = +$ if $min(p) > 0$

$\alpha(p) = -$ if $max(p) < 0$

$\alpha(0) = 0$

$\alpha(\emptyset) = \bot$

$\alpha(p) = \top$, otherwise



Congruence abstraction

Congruence abstraction

$C = \mathbb{Z}$

$D = \{0, \ldots, n-1\}$

$\alpha(c) = c \bmod n$

Cartesian predicate abstraction

Cartesian predicate abstraction is defined by a set of predicates

$P = \{p_1, \ldots, p_n\}$

$C = \mathfrak{p}(\mathbb{Q}^{|V|})$

$D = \bot \cup \mathfrak{p}(P)$ // $\emptyset$ represents $\top$

$\bot \sqsubseteq S_1 \sqsubseteq S_2$ if $S_2 \subseteq S_1$

$\alpha(c) = \{p \in P | c \Rightarrow p\}$

$\gamma(S) = \bigwedge S$

Example:

$V = \{\mathtt{x}, \mathtt{y}\}$

$P = \{\mathtt{x} \leq 1, -\mathtt{x} - \mathtt{y} \leq -1, \mathtt{y} \leq 5\}$

$\alpha(\{(0, 0)\}) = \{\mathtt{x} \leq 1, \mathtt{y} \leq 5\}$

$\alpha((\mathtt{x} - 1)^2 + (\mathtt{y} - 3)^2 = 1) = \{-\mathtt{x} - \mathtt{y} \leq -1, \mathtt{y} \leq 5\}$

Boolean predicate abstraction

Boolean predicate abstraction is also defined by a set of predicates
$P = \{p_1, \ldots, p_n\}$

$C = \mathfrak{p}(\mathbb{Q}^{|V|})$
$D =$ boolean formulas over predicates in P
$F_1 \sqsubseteq F_2$ if $F_1 \Rightarrow F_2$
$\alpha(c) =$ strongest boolean formula over P that contains c
$\gamma(F) = F$

Example:
$V = \{\mathtt{x}, \mathtt{y}\}$
$P = \{\mathtt{x} \leq 1, -\mathtt{x} - \mathtt{y} \leq -1, \mathtt{y} \leq 5\}$
$\alpha(-2x - y \leq -2) = -\mathtt{x} - \mathtt{y} \leq -1 \vee \neg(\mathtt{x} \leq 1)$

Topic 4.4

Abstract fixed point

Abstract operations

For a concrete operation $f : C^n \to C$, we define an abstract operation $f^\# : D^n \to D$ as follows

$$f^\#(x_1, \ldots, x_n) = \alpha \circ f(\gamma(x_1), \ldots, \gamma(x_n))$$

For example,

- $x \sqcup y = \alpha(\gamma(x) \cup \gamma(y))$
- $x \sqcap y = \alpha(\gamma(x) \cap \gamma(y))$
- $sp^\#(d, \rho) = \alpha \circ sp(\gamma(d), \rho)$

Computing approximate least fixed point

Let $f : C \to C$ be a monotonic operator.

Our goal is to compute $lfp_a(f)$, which is in general impossible.

Instead, we compute an approximation of $lfp_a(f)$ using α.

Theorem 4.3

Let $(C, \subseteq, \emptyset, \mathbb{Q}^V, \cup, \cap)$ *and* $(D, \sqsubseteq, \bot, \top, \sqcup, \sqcap)$ *are complete lattices,*

$$(C, \subseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (D, \sqsubseteq),$$

and $f : C \to C$ *and* $f^{\#}$ *are continuous operators then*

$$lfp_a(f) \subseteq \gamma(lfp_{\alpha(a)}(f^{\#}))$$

Exercise 4.4

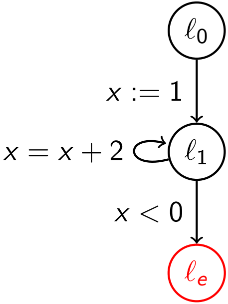
Prove the above theorem *Hint: First show iterates on both the sides are related*

Example : abstract fixed point computation

Example 4.2

Let us use sign abstraction to analyze the program
$D = \{\top, +, -, 0, \bot\}$

Consider program:



$X^0_{\ell_0} := \alpha(\top), X^0_{\ell_1} := \alpha(\bot), X^0_{\ell_e} := \alpha(\bot)$
$X^0_{\ell_0} := \top, X^0_{\ell_1} := \bot, X^0_{\ell_e} := \bot$

$$X^1_{\ell_1} = X^0_{\ell_1} \sqcup sp^{\#}(x' = 1, X^0_{\ell_0}) \sqcup sp^{\#}(x' = x + 2, X^0_{\ell_1})$$
$$= \bot \sqcup \alpha(sp(x' = 1, \gamma(X^0_{\ell_0}))) \sqcup \alpha(sp(x' = x+2, \gamma(X^0_{\ell_1})))$$
$$= \alpha(sp(x' = 1, \gamma(\top))) \sqcup \alpha(sp(x' = x + 2, \gamma(\bot)))$$
$$= \alpha(sp(x' = 1, \gamma(\top))) \sqcup \alpha(\bot)$$
$$= \alpha(x = 1) \sqcup \alpha(\bot) = + \sqcup \alpha(\bot) = +$$

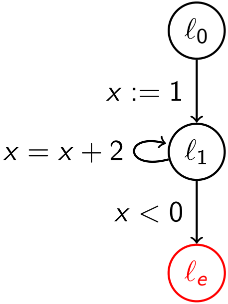
$X^1_{\ell_0} := \top, X^1_{\ell_1} := +, X^1_{\ell_e} := \bot$

Exercise 4.5

Calculate $X^1_{\ell_e}$

Example : abstract fixed point computation (contd.)

Consider program:



$$X^2_{\ell_1} = X^1_{\ell_1} \sqcup sp^{\#}(x' = 1, X^1_{\ell_0}) \sqcup sp^{\#}(x' = x + 2, X^1_{\ell_1})$$
$$= + \sqcup \alpha(sp(x' = 1, \gamma(X^1_{\ell_0}))) \sqcup \alpha(sp(x' = x+2, \gamma(X^1_{\ell_1})))$$
$$= + \sqcup \alpha(sp(x' = 1, \gamma(\top))) \sqcup \alpha(sp(x' = x + 2, \gamma(+)))$$
$$= + \sqcup \alpha(sp(x' = 1, \gamma(\top))) \sqcup \alpha(x > 2)$$
$$= + \sqcup \alpha(x = 1) \sqcup \alpha(x > 2) = + \sqcup + \sqcup + = +$$
$$X^2_{\ell_0} := \top, X^2_{\ell_1} := +, X^2_{\ell_e} := \bot$$

Fixed point reached

Exercise 4.6

Calculate $X^2_{\ell_e}$

Demo - The Interproc Analyzer

http://pop-art.inrialpes.fr/interproc/interprocweb.cgi

Exercise 4.7

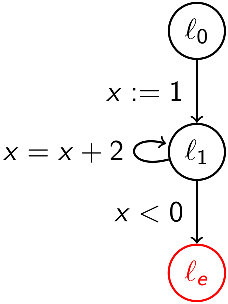
Run Interproc on the following code

```
var i:int;
begin
  i = 0;
  while (i<=10) do
    i = i+2;
  done;
end
```

Example: interval abstraction

Example 4.3

Consider program:



Let us use interval abstraction:

$X^0_{\ell_0} := \top, X^0_{\ell_1} := \bot, X^0_{\ell_e} := \bot$

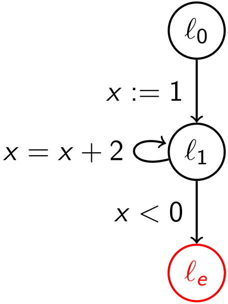
$$X^1_{\ell_1} := X^0_{\ell_1} \sqcup sp^{\#}(x' = 1, X^0_{\ell_0}) \sqcup sp^{\#}(x' = x + 2, X^0_{\ell_1})$$
$$= \bot \sqcup \alpha(sp(x' = 1, \gamma(X^0_{\ell_0}))) \sqcup \alpha(sp(x' = x+2, \gamma(X^0_{\ell_1})))$$
$$= \alpha(sp(x' = 1, \gamma(\top))) \sqcup \bot$$
$$= \alpha(sp(x' = 1, \gamma(\top))) = \alpha(x = 1) = [1, 1]$$

$X^1_{\ell_0} := \top, X^1_{\ell_1} := [1, 1], X^1_{\ell_e} := \bot$

Example: interval abstraction(contd.)

Example 4.4

Consider program:



$X^2_{\ell_1} = X^1_{\ell_1} \sqcup sp^{\#}(x' = 1, X^1_{\ell_0}) \sqcup sp^{\#}(x' = x + 2, X^1_{\ell_1})$
$= [1,1] \sqcup \alpha(sp(x' = 1, \gamma(\top))) \sqcup \alpha(sp(x' = x + 2, \gamma([1,1])))$
$= [1,1] \sqcup [3,3] = [1,3]$

$X^2_{\ell_0} := \top, X^2_{\ell_1} := [1,3], X^2_{\ell_e} := \bot$

$X^3_{\ell_0} := \top, X^3_{\ell_1} := [1,5], X^3_{\ell_e} := \bot$

... *the process will go on forever*

Acceleration

Many interesting abstract domains are of infinite size.

Abstraction may only provide simple calculations, but not convergence.

For convergence we need acceleration using a special operator call widening.

If we do too much widening then we may need narrowing

Widening

Definition 4.5

A *widening* $\nabla : D \times D \to D$ *on a poset* $(D, \sqsubseteq)$ *satisfies*

- $\forall x, y \in D.\ x \sqsubseteq x \nabla y \wedge y \sqsubseteq x \nabla y$
- *for an increasing chain* $x_0 \sqsubseteq x_1 \ldots$, *the increasing chain*

$$y^0 \triangleq x^0 \qquad y^n \triangleq y^{n-1} \nabla x^n$$

is not strictly increasing.

Definition 4.6

widening iterates $(\mathrm{I}^k, k < n)$ *for monotone function* f *from* $a \in prefp(f)$

- $\mathrm{I}^0 \triangleq a$
- $\mathrm{I}^{n+1} \triangleq \mathrm{I}^n \qquad$ *if* $f(\mathrm{I}^n) \sqsubseteq \mathrm{I}^n$
- $\mathrm{I}^{n+1} \triangleq \mathrm{I}^n \nabla f(\mathrm{I}^n) \qquad$ *if* $f(\mathrm{I}^n) \not\sqsubseteq \mathrm{I}^n$

Theorem 4.4

There exists $k \in \mathbb{N}$, $f(\mathrm{I}^k) \sqsubseteq \mathrm{I}^k$ *and* $lfp_a(f) \sqsubseteq \mathrm{I}^k$.

Example : widening for interval domain

$[a, b] \triangledown \bot = [a, b]$
$\bot \triangledown [a, b] = [a, b]$
$[a, b] \triangledown [a', b'] = [((a' < a)? -\infty : a), ((b' > b)?\infty : b)]$

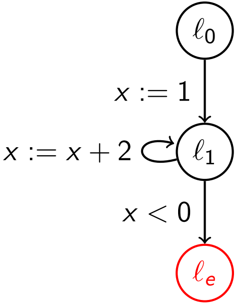
$[2, 3] \triangledown [-3, 2] = [-\infty, 3]$
$[2, 3] \triangledown [4, 6] = [2, \infty]$
$[2, 3] \triangledown [1, 6] = [-\infty, \infty]$

Exercise 4.8

a. Show $\triangledown$ for interval domain satisfies the definition of widening
b. Show $\triangledown$ is not symmetric and monotone

Example: widening in action

Consider program:



$$X^0_{\ell_0} := \top, X^0_{\ell_1} := \bot, X^0_{\ell_e} := \bot$$

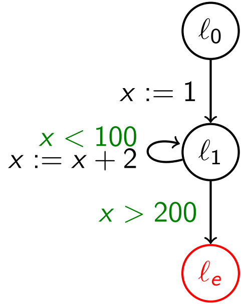
$$X^1_{\ell_0} := \top, X^1_{\ell_1} := [1,1], X^1_{\ell_e} := \bot$$

$$X^2_{\ell_1} = X^1_{\ell_1} \triangledown (sp^{\#}(x' = 1, X^1_{\ell_0}) \sqcup sp^{\#}(x' = x + 2, X^1_{\ell_1}))$$
$$= [1,1] \triangledown ([1,1] \sqcup [3,3]) = [1,1] \triangledown [1,3] = [1,+\infty]$$

$$X^2_{\ell_0} := \top, X^2_{\ell_1} := [1,+\infty], X^2_{\ell_e} := \bot$$

Example: too much widening

Now consider:



$$X^0_{\ell_1} := \bot, X^0_{\ell_e} := \bot$$

$$X^1_{\ell_1} := [1,1], X^0_{\ell_e} := \bot$$

$$X^2_{\ell_1} = [1,1] \nabla ([1,1] \sqcup sp^{\#}(x < 100 \wedge x' = x + 2, X^1_{\ell_1}))$$
$$= [1,1] \nabla ([1,1] \sqcup [3,3]) = [1,+\infty]$$
$$X^2_{\ell_1} := [1,+\infty], X^2_{\ell_e} := \bot$$

$$X^3_{\ell_e} = X^2_{\ell_e} \nabla (sp^{\#}(x > 200 \wedge x' = x, X^2_{\ell_1}))$$
$$X^3_{\ell_e} = \bot \nabla (sp^{\#}(x > 200 \wedge x' = x, [1,+\infty]))$$
$$X^3_{\ell_e} = \bot \nabla [200,+\infty] = [200,+\infty]$$
$$X^2_{\ell_1} := [1,+\infty], X^3_{\ell_e} = [200,+\infty]$$

... reaching error location

Narrowing

Unfortunate misnomer!!
Narrowing is not the dual of widening!

Definition 4.7

A narrowing $\triangle : D \times D \to D$ *on a poset* $(D, \sqsubseteq)$ *satisfies*

- $\forall x, y \in D.\ y \sqsubseteq x \Rightarrow y \sqsubseteq x \triangle y \sqsubseteq x$
- *for an decreasing chain* $\dots x_1 \sqsubseteq x_0$, *the decreasing chain*

$$y^0 \triangleq x^0 \qquad y^n \triangleq y^{n-1} \triangle x^n$$

is not strictly decreasing.

Definition 4.8

narrowing iterates $(\mathrm{I}^k, k < n)$ *for monotone function* f *from* $a \in postfp(f)$

- $\mathrm{I}^0 \triangleq a$
- $\mathrm{I}^{n+1} \triangleq \mathrm{I}^n \qquad$ *if* $f(\mathrm{I}^n) = \mathrm{I}^n$
- $\mathrm{I}^{n+1} \triangleq \mathrm{I}^n \triangle f(\mathrm{I}^n) \quad$ *if* $\mathrm{I}^n \sqsubseteq f(\mathrm{I}^n)$

Theorem 4.5

For all $x \in X.\ x = f(x) \sqsubseteq a \Rightarrow \exists k.\ x \sqsubseteq \mathrm{I}^k = \mathrm{I}^{k+1} \sqsubseteq a$

Example: narrowing for interval abstraction

$\bot \triangle [a, b] = \bot$
$[a, b] \triangle [a', b'] = [((a = -\infty)?a' : a), ((b = \infty)?b' : b)] \quad \text{if } [a', b'] \sqsubseteq [a, b]$

$[1, 3] \triangle [1, 2] = [1, 3]$
$[2, 3] \triangle [4, 6] = \text{(undefined)}$
$[-\infty, 6] \triangle [1, 3] = [1, 6]$

Theorem 4.6
Show $\triangle$ for interval abstraction is truly narrowing operator.

Using narrowing after widening

Let us suppose we have monotonic $f : D \to D$, $a \in \mathit{prefp}(f)$, widening ∇, and narrowing $\triangle$.

- Apply widening iterates to obtain b such that $a \sqsubseteq b \in \mathit{postfp}(f)$
- Then, apply narrowing iterates to obtain c such that $c = f(c) \sqsubseteq b$

Exercise 4.9

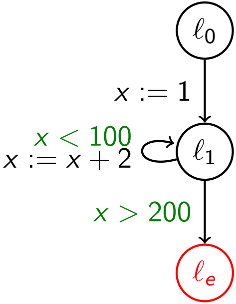
Show $a \sqsubseteq c$.

Example: narrowing interval domain

Result of widening iterates:

$X^3_{\ell_1} := [1, +\infty], X^3_{\ell_e} := [200, +\infty]$

Now consider:



$$
\begin{aligned}
X^4_{\ell_1} &:= X^3_{\ell_1} \triangle ([1,1] \sqcup sp^{\#}(x < 100 \wedge x' = x + 2, X^3_{\ell_1}))\\
&= [1,\infty] \triangle ([1,1] \sqcup sp^{\#}(x < 100 \wedge x' = x + 2, [1,\infty]))\\
&= [1,\infty] \triangle ([1,1] \sqcup [3,101])\\
&= [1,\infty] \triangle ([1,101]) = [1,101]
\end{aligned}
$$

$$
\begin{aligned}
X^4_{\ell_e} &:= X^3_{\ell_e} \triangle (sp^{\#}(x > 200 \wedge x' = x, X^3_{\ell_1}))\\
&= [200,+\infty] \triangle (sp^{\#}(x > 200 \wedge x' = x, [1,+\infty]))\\
&= [200,+\infty] \triangle [200,+\infty] = [200,+\infty]
\end{aligned}
$$

$X^4_{\ell_1} := [1,101], X^4_{\ell_e} := [200,\infty]$

$$
\begin{aligned}
X^5_{\ell_1} &:= X^4_{\ell_1} \triangle ([1,1] \sqcup sp^{\#}(x < 100 \wedge x' = x + 2, X^4_{\ell_1}))\\
&= [1,101]
\end{aligned}
$$

$$
\begin{aligned}
X^5_{\ell_e} &:= X^4_{\ell_e} \triangle (sp^{\#}(x > 200 \wedge x' = x, X^4_{\ell_1}))\\
&= [200,+\infty] \triangle (sp^{\#}(x > 200 \wedge x' = x, [1,101]))\\
&= [200,+\infty] \triangle \bot = \bot
\end{aligned}
$$

Fixed point reached

$X^5_{\ell_1} := [1,101], X^5_{\ell_e} := \bot$

Widening and narrowing policy

We need not apply narrowing/widening of at every iteration or for every variable.

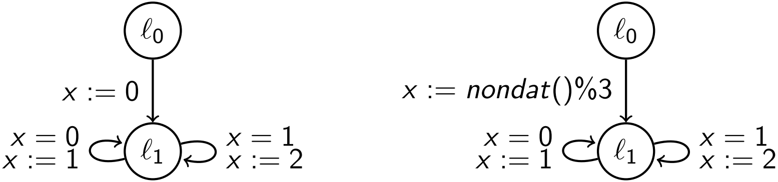
- use narrowing/widening operators only at cut points
- use narrowing/widening operators at every ith iteration

Exercise : widening chaos

The proposed machinery may have unpredictable behaviors!!

Exercise 4.10

Apply widening iterates of interval domain on the following examples



Abstract domain

An abstract domain consists of

- a lattice $(D, \sqsubseteq, \sqcup, \sqcap)$,
- a abstraction function $\alpha : C \to D$ and a concretization function $\gamma : D \to C$ such that
$$(D, \sqsubseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (C, \subseteq),$$
- a widening operator $\nabla : D \times D \to D$, and
- a narrowing operator $\triangle : D \times D \to D$.

Box domain

If the program has multiple variables then the product of interval domains for each variable is the box domain. All operators naturally extend.

However, $sp^{\#}$ can be optimized. Instead of computing $sp^{\#}$ in three steps, one may compute it directly using interval arithmetic.

Example 4.5

$sp^{\#}(x := y + x, ([2,3]_x, [1,4]_y)) = (([2,3] + [1,4])_x, [1,4]_y)$
$= ([3,7]_x, [1,4]_y)$

Polyhedral domain

Let us assume $V = \{x_1, \dots, x_n\}$.

$D = \{AV \leq b | A \in \mathbb{Q}^{m \times n} \wedge b \in \mathbb{Q}^{m \times 1}\}$

D has natural complete lattice structure.

However, there is no canonical representation of polyhedra

We will first discuss the representation to use to implement various operators efficiently.

Exercise 4.11

Define a complete lattice over polyhedra

Dual representation

Representation by constraints:
(A, b) where $A \in \mathbb{Q}^{m \times n}$ and $b \in \mathbb{Q}^{m \times 1}$ representing

$$\gamma((A, b)) = \{v | Av \leq b\}$$

Representation by generators:
(Q, R) where $Q = \{v_1, \dots, v_p\}$ is a set of vertices and $R = \{r_1, \dots, r_m\}$ set of rays in $\mathbb{Q}^n$.

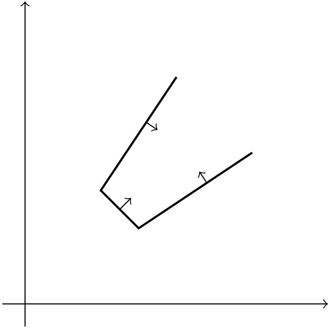
$$\gamma((Q, R)) = \{\sum_{i=1}^{p} \lambda_i v_i + \sum_{j=1}^{m} \mu_j r_j | \forall i \in 1..r.\ \mu_i \geq 0 \wedge$$

$$\forall i \in 1..p.\ \lambda_i \geq 0 \wedge \sum_{i=1}^{p} \lambda_i = 1 \wedge \}$$

Why dual representations?
Some operations are efficient if both the representations are available.

Example : dual representation

$$2x - 3y \leq 0 \quad -3x + 2y \leq 0 \quad x + y \geq 25$$



$$(A, b) = \left(\begin{bmatrix} 2 & -3 \\ -3 & 2 \\ -1 & -1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ -25 \end{bmatrix} \right)$$

$$(Q, R) = (\{(15, 10), (10, 15)\}, \{(3, 2), (2, 3)\})$$

Converting constraints to generators

Chernikova algorithm iteratively computes generators for a polyhedron that is given as $AV \leq b$.

- The algorithm considers inequalities of $AV \leq b$ sequentially.
- At kth iteration, it computes the generators (Q_k, R_k) for the inequalities seen so far.
- After considering all inequalities it has generators for $AV \leq b$.
- Initially, $(Q_0, R_0) := (\{0\}, \{e_1, -e_1 \dots, e_n, -e_n\})$, which spans whole vector space

kth iteration of Chernikova algorithm

Let us suppose $aV \leq c$ is being considered at kth step. We construct (Q_k, R_k) as follows.

- if $v \in Q_{k-1}$ and $av \leq c$ then $v \in Q_k$
- if $r \in R_{k-1}$ and $ra \leq 0$ then $r \in R_k$
- if $v_1, v_2 \in Q_{k-1}$, $av_1 \leq c$, and $av_2 > c$ then $\frac{c-av_1}{av_2-av_1}v_1 + \frac{av_2-c}{av_2-av_1}v_2 \in Q_k$
- if $v \in Q_{k-1}$ and $r \in R_{k-1}$ such that $av < c$ and $ar > 0$ or $av > c$ and $ar < 0$ then $v + \frac{c-av}{ar}r \in Q_k$
- if $r_1, r_2 \in R_{k-1}$, $ar_1 > 0$, and $ar_2 < 0$ then $(ar_2)r_1 - (ar_1)r_2 \in R_k$

The algorithm generates redundant vertices and rays.
Worst case blow up 2^m, if the number of constraints is $2m$. Need to remove redundancies during the construction, e.g., Le. Verge algorithm

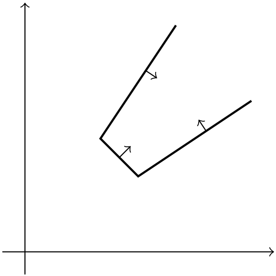
By duality, this algorithm can be used to convert generators into constraints.

Exercise 4.12

Give the duality construction

Example: chernikova algorithm

$$2x - 3y \le 0 \quad -3x + 2y \le 0 \quad x + y \ge 25$$



$(Q_0, R_0) = (\{(0,0)\}, \{(1,0), (-1,0), (0,1), (0,-1)\})$

Consider $2x - 3y \le 0$

$(Q_1, R_1) = (\{(0,0)\}, \{(3,2), (-3,-2), (1,0), (0,-1)\})$

Consider $-3x + 2y \le 0$

$(Q_2, R_2) = (\{(0,0)\}, \{(3,2), (2,3)\})$

Consider $x + y \ge 25$

$(Q_3, R_3) = (\{(10,15), (15,10)\}, \{(3,2), (2,3)\})$

Minimal representations

A constraint representation (A, b) is minimal if one can not drop a row from A and b without changing the corresponding polyhedron $\gamma((A, b))$

A generator representation (Q, R) is minimal if one can not drop a vertices or ray from Q and R without changing the corresponding polyhedron $\gamma((A, b))$

We assume that the representations are minimal. However it is not strictly needed in implementing various operations.

Lattice operations in polyhedra

- is $\bot = (Q, R)$: if Q is empty
- $(Q, R) \sqsubseteq (A, b) \triangleq \forall v \in Q.\ Av \leq b \land \forall r \in R.\ Ar \leq 0$
- $(A_1, b_1) \sqcap (A_2, b_2) \triangleq (\begin{bmatrix} A_1 \\ A_2 \end{bmatrix}, \begin{bmatrix} b_1 \\ b_2 \end{bmatrix})$
- $(Q_1, R_1) \sqcup (Q_2, R_2) \triangleq (Q_1 \cup Q_2, R_1 \cup R_2)$
- $sp^{\#}(V' = AV + b, (Q, R)) = (\{Av + b | v \in Q\}, \{Ar | r \in R\})$
- $sp^{\#}(A_2 V \leq b_2 \land V' = V, (A_1, b_1)) = (\begin{bmatrix} A_1 \\ A_2 \end{bmatrix}, \begin{bmatrix} b_1 \\ b_2 \end{bmatrix})$

Both representations are useful for efficient implementations.

Polyhedral widening

Consider polyhedron $L = (A^1, b^1)$ and $M = (A^2, b^2)$.
If L is empty then, $L \triangledown M = M$.
Otherwise, $L \triangledown M = \beta_1 \cup \beta_2$, where

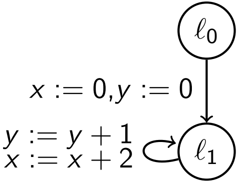
- $\beta_1 \triangleq \{aV \leq c \in L | aV \leq c \text{ contains } M\}$
- $\beta_2 \triangleq \{aV \leq c \in M |\ aV \leq c$ can replace some inequality in L without changing L $\}$

Example: Polyhedral widening

$L = \{(x, y) | 0 \leq x \wedge x \leq y \wedge y \leq x\}$
$M = \{(x, y) | 0 \leq x \wedge x \leq y \wedge y \leq x + 1\}$
$L \triangledown M = \{(x, y) | 0 \leq x \wedge x \leq y\}$

$L = \{(x, y) | 0 \leq x \wedge x \geq 0 \wedge 0 \leq y \wedge y \geq 0\}$
$M = \{(x, y) | 0 \leq y \leq x \leq 1\}$
$L \triangledown M = \{(x, y) | 0 \leq y \leq x\}$

Example: polyhedral domain



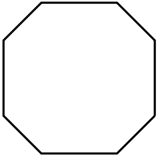
$$
\begin{aligned}
X^1_{\ell_1} &= \{0 \leq x \leq 0, 0 \leq y \leq 0\} \\
X^2_{\ell_1} &= X^1_{\ell_1} \nabla (\{0 \leq x \leq 0, 0 \leq y \leq 0\} \sqcup \\
&\qquad \{1 \leq x \leq 1, 2 \leq y \leq 2\}) \\
&= X^1_{\ell_1} \nabla \{0 \leq x \leq 2, x \leq 2y, x \geq 2y\} \\
&= \{0 \leq x, x \leq 2y, x \geq 2y\}
\end{aligned}
$$

... fixed point reached

Octagon domain

Let us assume $V = \{x_1, \ldots, x_n\}$.

$O = \{\pm \mathtt{x} \pm \mathtt{y} \leq c | \mathtt{y}, \mathtt{x} \in V, c \in \mathbb{I}\}$
where $\mathbb{I} \in \{\mathbb{Q}, \mathbb{Z}, \mathbb{R}\}$



$D = \{o_1 \wedge \cdots \wedge o_k | \forall i.\ o_i \in O\}$

Domain representation

- Since a tight ODBM represents $F \in D$ canonically, we say D is a set of tight ODBMs (recall lecture 5).
- Tight ODBMs do not represent unsat formulas therefore we need to add a special element $\bot$ to represent the least element.

Lattice operators in Octagonal domain

Let A^1 and A^2 be $2n \times 2n$ tight ODBMs.

- $is\bot?$ due to canonical representation trivial
- $A^1 \sqsubseteq A^2 \triangleq A^1 \dot{\leq} A^2$ *
- $A^1 \sqcap A^2 \triangleq (\dot{\min}(A^1, A^2))^\bullet$
- $A^1 \sqcup A^2 \triangleq (\dot{\max}(A^1, A^2))^\bullet$
- $A^1 \nabla A^2 = A^\bullet$, where $A_{ij} = (A^2_{ij} > A^1_{ij} ? \infty : A^1_{ij})$
- $sp^\#(\rho, A^1) = \alpha((\exists V' F[A^1] \wedge \rho)[V'/V])$

ρ is a polyhedron then the param to α is also polyhedron.

---

* $\dot{\leq}$ is pointwise $\leq$

Octagonal abstraction of polyhedron

$\alpha((Q,R)) = A$ is defined as follows

- $A_{(2i)(2i-1)} = (\ \exists r \in R.\ r_i > 0\ ?\ \infty :\ 2\max\{v_i | v \in V\})$
- $A_{(2i-1)(2i)} = (\ \exists r \in R.\ r_i < 0\ ?\ \infty :\ -2\min\{v_i | v \in V\})$
- $A_{(2i-1)(2j-1)} = A_{(2i)(2j)} = (\ \exists r \in R.\ r_i > r_j\ ?\ \infty :\ \max\{v_i - v_j | v \in V\})$
- $A_{(2i-1)(2j)} = (\ \exists r \in R.\ r_i + r_j > 0\ ?\ \infty :\ -\min\{v_i + v_j | v \in V\})$
- $A_{(2i)(2j-1)} = (\ \exists r \in R.\ 0 > r_i + r_j\ ?\ \infty :\ \max\{v_i + v_j | v \in V\})$
- $A_{(i)(i)} = 0$

Topic 4.5

Problems

Apply Sign abstraction

Exercise 4.13

Apply sign abstraction on the following example?

```
main (){
  x := 0;
  y := -1;
  while( x < 20 ) {
    if( x < 10 ) {
      y := y - 1;
    }else{
      y := y + 1;
    }
    x = x + 1;
  }
}
```

Apply Chernikova algorithm

Exercise 4.14

Apply Chernikova algorithm on the following polyhedron

$$\{x - y \leq 0 \land x + y \leq 4 \land 0 \leq x\}$$

Show intermediate steps