# Special Topics in Logic and Security 1

## Abstract Domains

Paul Irofti

Master Year II, Sem. I, 2023-2024

# Multiple variable

What happens when we deal with multiple variables?

```
int A[4][8] = {...};
int i, j;
int sum = 0;

for (i = 0; i < 4; i++)
    for (j = 0; j < 8; j++)
        sum += A[i][j];

printf("sum = %d\n", sum);
```

# Inequalities

**Goal:** numerical constraints on the variables domain $\mathcal{X}$.

Let $\boldsymbol{x}$ be the vector consisting of all the variables in $\mathcal{X}$.

**Lin**: Let $\mathit{Lin}^{\mathbb{R}}$ be the set of all linear expressions of the form $\boldsymbol{ax}$, where $\boldsymbol{a} \in \mathbb{R}^n$ are the variable coefficients.

**Ineq**: Let $\mathit{Ineq}^{\mathbb{R}}$ be the set of linear inequalities of the form $\boldsymbol{ax} \leq c$, where $c \in \mathbb{R}$ is a constant.

**Examples:**

- $6x_3 \leq x_1 + 5 \iff \begin{bmatrix} -1 & 0 & 6 & 0 & \dots & 0 \end{bmatrix} \boldsymbol{x} \leq 5$
- $x_2 = 7 \iff x_2 \leq 7 \wedge x_2 \geq 7$
- $x_2 \geq 7 \iff -x_2 \leq -7$
- Integer expressions: $e_1 < e_2 \iff e_1 \leq e_2 - 1$

# Inequalities: Geometric Properties

Each inequality $\boldsymbol{ax} \leq c \in \mathit{Ineq}^{\mathbb{R}}$ creates the half-plane $[\![\boldsymbol{ax} \leq c]\!] = \{x \in \mathbb{R}^{|\mathcal{X}|} \mid \boldsymbol{ax} \leq c\}$.

A set of inequalities $I \subseteq \mathit{Ineq}^{\mathbb{R}}$ produces a closed convex space $[\![I]\!] = \bigcap_{\iota \in I} [\![\iota]\!]$.

Let $\mathcal{S} = \{[\![I]\!] \mid I \subseteq \mathit{Ineq}^{\mathbb{R}}\}$ be the set of all convex sets.

### Definition

As with ranges, we define $\overline{\Upsilon}$ as the operation on $S_1, S_2 \in \mathcal{S}$ that produces the space $S = S_1 \overline{\Upsilon} S_2$ s.t. $S_1 \subseteq S$ and $S_2 \subseteq S$.

### Theorem

$(\mathcal{S}, \subseteq, \overline{\Upsilon}, \cap)$ *forms a lattice.* ***Proof:*** *Exercise!*

### Remark

*The solution to a set of equations, like the example from last course for dealing with ranges, exists and can be solved by the fixed point theorem: as we solved for $\imath$, we can solve for multiple variables at once.*

# Proof: $(\mathcal{S}, \subseteq, \overline{\Upsilon}, \cap)$ forms a lattice

Let $S, S_1, S_2, S_3 \in \mathcal{S}$ $(\mathcal{S}, \subseteq)$ POSET:

- reflexive: $S \in \mathcal{S} \implies S \subseteq S \quad \forall S$
- anti-symmetric: $S_1, S_2 \in \mathcal{S}$ and $S_1 \subseteq S_2, S_2 \subseteq S_1 \implies S_1 = S_2$ ;
  Let $p \in S_1 \implies p \in S_2 (S_1 \subseteq S_2)$
  If $r \in S_2$ and $r \notin S_1 \implies S_2 \not\subseteq S_1$. False!
- transitivity: $S_1 \subseteq S_2, S_2 \subseteq S_3 \implies S_1 \subseteq S_3$ ;
  Can I draw a line that passes through $S_1$ but does not pass through $S_3$?

Lattice:

- $\overline{\Upsilon} : \mathcal{S} \times \mathcal{S} \mapsto \mathcal{S}, \ S_1 \overline{\Upsilon} S_2 = \sup\{S_1, S_2\}$; what is the sup of two convex sets?
- $\cap : \mathcal{S} \times \mathcal{S} \mapsto \mathcal{S}, \ S_1 \cap S_2 = \inf\{S_1, S_2\}$; what is the intersections of two convex sets?
- associative: $(S_1 \overline{\Upsilon} S_2) \overline{\Upsilon} S_3 = S_1 \overline{\Upsilon} (S_2 \overline{\Upsilon} S_3)$ ; $(S_1 \cap S_2) \cap S_3 = S_1 \cap (S_2 \cap S_3)$
- commute: $S_1 \overline{\Upsilon} S_2 = S_2 \overline{\Upsilon} S_1$ ; $S_1 \cap S_2 = S_2 \cap S_1$
- absorb: $S_1 \overline{\Upsilon} (S_1 \cap S_2) = S_1$ ; $S_1 \cap (S_1 \overline{\Upsilon} S_2) = S_1$

# Computational issues

Infinite number of inequalities. Polyhedra.

- there are convex sets $S \in \mathcal{S}$ s.t. $|I| \notin \mathbb{N},\ \forall\ I \subseteq \mathit{Ineq}$ and $[\![I]\!] = S$.
- implementations can store convex spaces generated by a finite set of inequalities that are also called polyhedra

**Example**

- the sequence of regular polyhedra (equilateral triangle, square, heaxgon, dodecadon, ...) converges towards a disc;
- we can formulate the ascending chain $S_1 \subseteq S_2 \subseteq S_3 \ldots$
- $S_i$ is a polyhedra; $\bigcup_i S_i$ is not because a disc can not be represented by a finite set of inequalities

**Remark**

*The polyhedra lattice is incomplete. The fixed point theorem converges in a convex space that is not a polyhedra!*

# Computational issues

Unlimited coefficients growth.

- $Lin^{\mathbb{R}}$ and $Ineq^{\mathbb{R}}$ are defined in $\mathbb{R}$
- floating point numbers represent finite elements from $\mathbb{R}$; more precise, the numbers on a computing device are elements of $\mathbb{Q}$

## Example

- let $x_i \in \mathbb{Q}$ be the sequence defined by $x_0 = 1$ and $x_{n+1} = (x_n + 2/x_n)/2$
- $S_j = [\![\{1 \leq x \leq x_j\}]\!]$ includes $x_0, \ldots, x_j$
- we can formulate the ascending chain $S_0 \subseteq S_1 \subseteq \ldots$
- the chain converges to $[\![\{1 \leq x \leq \sqrt{2}\}]\!]$

## Remark

*The fixed point theorem can generate inequalities that involve coefficients and constants of infinite size. Constraining the coefficients and constants to the set of rational numbers leads to an incomplete domain!*

# The widening operator

**Definition**

**Widening** is an acceleration technique that allows the execution of the iterations of the fixed point theorem to stop in finite time through the elimination of certain inequalities.

When widening is applied to an ascending chain, we obtain an inequalities set that can be finitely described.

**Definition**

**Redefine:** Let *Lin* be the set of coefficients with $\boldsymbol{a} \in \mathbb{Z}^n$ and *Ineq* the generated inequalities with *Lin* and constants $c \in \mathbb{Z}$.

The half-space defined by an inequality becomes $[\![\boldsymbol{ax} \leq c]\!] = \{\boldsymbol{x} \in \mathbb{Q}^{|\mathcal{X}|} \mid \boldsymbol{ax} \leq c\}$.

# Polyhedra widening operator $\nabla$

The convex spaces resulting through widening are $Poly = \{[\![I]\!] \mid I \in Ineq \wedge |I| \in \mathbb{N}\}$ or the set of convex polyhedra generated in finite time.

Thus the widening operator is $\nabla : Poly \times Poly \to Poly$ with the following properties:

1. $P \subseteq P \nabla Q, \forall P, Q$
2. $Q \subseteq P \nabla Q, \forall P, Q$
3. For all chains $P_0 \subseteq P_1 \subseteq \ldots$, the chain $R_0 = P_0$ and $R_{i+1} = R_i \nabla P_{i+1}$ is stable ($\exists i$ s.t. $\bigcup_{j \in \mathbb{N}} R_j \subseteq R_i$)

Examples:

- $1 \subseteq \frac{3}{2} \subseteq \frac{17}{12} \subseteq \cdots \subseteq x_{n+1}$
- $R_1 = R_0 \nabla P_1 = P_0 \nabla P_1$
  $R_2 = R_1 \nabla P_2 = (P_0 \nabla P_1) \nabla P_2$
  $R_3 = R_2 \nabla P_3 = ((P_0 \nabla P_1) \nabla P_2) \nabla P_3$
  $\vdots$

# Properties. Down-sides.

The lattice $(\mathit{Poly}, \leq_P, \vee_P, \wedge_P)$:

- $\leq_P$ is the inclusion operator $\subseteq$
- $\vee_P = \overline{\Upsilon}$ is the join operation for polyhedra
- $\wedge_P$ is the meet operation for sets

The lattice is incomplete because the join and meet operations, when applied to an arbitrary number of polyhedra, can lead to a non-polyhedra object.

The widening operator together with the incomplete lattice restrain the number of fixed points that can be attained.

**Definition**

A stable polyhedra obtained at convergence is generally a post-fixpoint: a polyhedra that contains the polyhedra of the fixed point. An approximation.

**Example**

- let the sequence $x_i \in \mathbb{Q}$ defined by $x_0 = 1$ and $x_{n+1} = (x_n + 2/x_n)/2$
- the chain converges to $[\![\{1 \leq x \leq \sqrt{2}\}]\!] \notin \mathit{Poly}$
- post-fixpoint: $[\![\{1 \leq x \leq 2\}]\!]$ or even $[\![\{1 \leq x\}]\!]$

# Operations

Assigning a value to a variable.

Let $P \in \mathit{Poly}$ and $x \in \mathcal{X}$. The instruction $x = 42$ corresponds to the polyhedra $P \wedge_P [\![\{x = 42\}]\!]$ which is in fact the polyhedra $P$ only that the value of $x$ is fixed at 42.

**Exercise:** Show that the operation $P \wedge_P [\![\{x = 42\}]\!]$ implements the semantic of the instruction `if (x == 42)`. Show an example!

Updating the value of a variable.

For $x$ to gain a new value, the old value has to go away: we use the projection operator $\exists_x : \mathit{Poly} \to \mathit{Poly}$:

$$\exists_{x_i}(P) = \{[x_1, \ldots, x_{i-1}, x, x_{i+1}, \ldots, x_n] \mid \boldsymbol{x} \in P, x \in \mathbb{R}\}$$

The operator removes any information regarding $x \in \mathcal{X}$ from $P \in \mathit{Poly}$.

**Exercise:** How do you implement the semantic of the instruction `x=42`? What about `x=x+1`?

# Special Operations

How do you implement the instruction `x=x+1`?

- $\exists_x(P) \wedge_P [\![\{x = x + 1\}]\!]$ is not feasible
- we use the temporary variable $t \in \mathcal{X}^T$ and then assign $t$ to $x$
- $\mathcal{X}^T \subseteq \mathcal{X}$ represents the abstract set of temporary variables that do not correspond to any of the program variables
- thus any instruction `x=e`, where $e$ is a linear expression

**Exercise:** Show that `x=e` can be implemented as

$$\exists_t([\![\{x = t\}]\!] \wedge_P \exists_x(P \wedge_P [\![\{t = e\}]\!]))$$

We denote this operation by $P \rhd x := e$.

# Special Operations

The notation $P \rhd x := e$ is sufficient for all assignment operations except division and *shift*.

Let $P \rhd x := y \gg n$ with $n \in \mathbb{N}$ be the right *shift* operations by $n$ bits.

- overwrite: $x$ is updated s.t. $P$ contains the integer solutions $x = \lfloor y/2^n \rfloor$
- the equivalent linear equation becomes:: $2^n x = y - d$ with $d \in \{0, \ldots, 2^n - 1\}$
- which leads to the model:

$$\exists_t(\llbracket \{x = t\} \rrbracket \wedge_P \exists_x(P \wedge_P \llbracket \{y - (2^n - 1) \leq 2^n t \leq y\} \rrbracket))$$

**Exercise:** Geometrically illustrate the polyhedra resulting from the operation $P' = P \rhd y := x \gg 2$. What happens when we are given $x = 8$ prior? ($P' \wedge_P \llbracket \{x = 8\} \rrbracket$.)

**Exercise:** How can we model the division operation?

# Optimization Techniques

How can I find the minimum value of a given expression $\boldsymbol{ax} \in \mathit{Lin}$ such that $x \in P$?

Let there be the set $C = \{c \in \mathbb{Z} \mid P \wedge_P [\![\{\boldsymbol{ax} \leq c\}]\!] \neq \emptyset\}$ containing all the constants $c$ for which the half-plane $\boldsymbol{ax} \leq c$ intersects the polyhedra $P$.

We define the function $\mathit{minExp} : \mathit{Lin} \times \mathit{Poly} \to (\mathbb{Z} \cup \{-\infty\})$

$$\mathit{minExp}(\boldsymbol{ax}, P) = \begin{cases} \min(C), & \text{if exists} \min(C) \\ -\infty, & \text{otherwise} \end{cases}$$

This is a linear programming problem of the kind

$$\min \boldsymbol{c}^T\boldsymbol{x} \;\text{ s.t. }\; \boldsymbol{Ax} \leq \boldsymbol{b}, \quad \boldsymbol{x} \geq 0$$

that can be solved by standard optimization algorithms such as the simplex algorithm.