Program verification 2019

Lecture 4: Understand abstraction

Instructor: Ashutosh Gupta

IITB, India

Compile date: 2019-01-15

Topic 4.1

Fixed point computation and Abstraction

Reachability as fixed point equation

Consider program $P = (V, L, \ell_0, \ell_e, E)$

Let X_ℓ be a variable representing the reachable valuations at location $\ell \in L$

We may compute reachability using *sp* via the following fixed point equation

$$X_{\ell_0} = \top$$
$$\forall \ell' \in L \setminus \{\ell_0\}.\ X_{\ell'} = \bigvee_{(\ell,\rho,\ell') \in E} sp(X_\ell, \rho)$$

We will use the following fixed point equation that has same fixed points as above.

$$X_{\ell_0} = \top$$
$$\forall \ell' \in L \setminus \{\ell_0\}.\ X_{\ell'} = X_{\ell'} \vee \bigvee_{(\ell,\rho,\ell') \in E} sp(X_\ell, \rho)$$

Note: For now, we are ignoring the constraints posed by the error location.

Fixed point computation

Initial assignment to variables and iteratively compute the fixed point

Let $X_\ell^i \triangleq$ value of X_ℓ at ith iteration.

In our setting, initially: $X_{\ell_0}^0 \triangleq \top$ and $X_\ell^0 \triangleq \bot$ for each $\ell \neq \ell_0$
and at each iteration

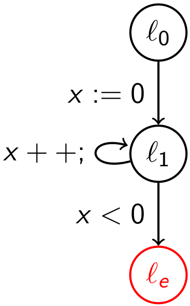
$$X_{\ell_0}^{k+1} = \top$$
$$\forall \ell' \in L \setminus \{\ell_0\}.\ X_{\ell'}^{k+1} = X_{\ell'}^k \vee \bigvee_{(\ell,\rho,\ell') \in E} sp(X_\ell^k, \rho)$$

If $\forall \ell.\ X_\ell^k = X_\ell^{k+1}$, then we say that the iterations have converged at iteration k and we have computed the fixed point.

Example: diverging analysis with *sp*

Example 4.1

Consider program:



Fixed point equations:

$$X_{\ell_0} = \top$$
$$X_{\ell_1} = sp(X_{\ell_0}, x' = 0) \vee sp(X_{\ell_1}, x' = x + 1)$$
$$X_{\ell_e} = sp(X_{\ell_1}, x < 0 \wedge x' = x)$$

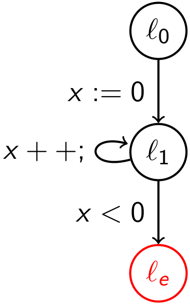
Iterates:

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

$$X^1_{\ell_0} := \top, X^1_{\ell_1} := (x = 0), X^1_{\ell_e} := \bot$$

$$X^2_{\ell_0} := \top$$
$$X^2_{\ell_1} := X^1_{\ell_1} \vee sp(X^1_{\ell_1}, x' = x + 1) \vee sp(X^1_{\ell_0}, x' = 0)$$
$$:= (x = 0) \vee sp(x = 0, x' = x + 1) \vee sp(\top, x' = 0)$$
$$:= (x = 0 \vee x = 1 \vee x = 0) := (0 \leq x \leq 1)$$
$$X^2_{\ell_e} := sp(X^1_{\ell_1}, x < 0 \wedge x' = x)$$
$$:= sp(x = 0, x < 0 \wedge x' = x) := \bot$$

Example: diverging analysis with *sp*(contd.)



Iterates(contd.):

$X^3_{\ell_0} := \top, X^3_{\ell_1} := (0 \leq x \leq 2), X^3_{\ell_e} := \bot$

$\vdots$

$X^n_{\ell_0} := \top, X^n_{\ell_1} := (0 \leq x \leq n-1), X^n_{\ell_e} := \bot$

...will never converge

How to compute fixed point effectively?

Now we introduce the key method of verification

Let us define

$$sp^{\#} : \Sigma(V) \times \Sigma(V, V') \rightarrow \Sigma(V)$$

Abstract post must satisfy the following condition over labels of P

$$sp(F, \rho) \Rightarrow sp^{\#}(F, \rho)$$

It is up to us how we choose $sp^{\#}$ that satisfies the above condition

Important: We have defined $sp^{\#}$ using formulas. However, any data type (domain) can work that is capable of representing set of states.

Abstract Fixed point

Replace sp by $sp^{\#}$ for faster convergence

initially: $X^0_{\ell_0} \triangleq \top$ and $X^0_{\ell} \triangleq \bot$ for each $\ell \neq \ell_0$
and at each iteration

$$X^{k+1}_{\ell_0} = \top$$
$$\forall \ell' \in L \setminus \{\ell_0\}.\ X^{k+1}_{\ell'} = X^k_{\ell'} \vee \bigvee_{(\ell,\rho,\ell') \in E} sp^{\#}(X^k_{\ell}, \rho)$$

After convergence, X_ℓ will be a superset of reachable states at ℓ.

Definition alert: Partial order and poset

Definition 4.1

On a set X, $\leq \subseteq X \times X$ is a partial order if

- *reflexive:* $\Delta_X \subseteq \leq$
- *anti-symmetric:* $\leq \cap \leq^{-1} \subseteq \Delta_X$
- *transitive:* $\leq \circ \leq \subseteq \leq$

We will use $x \leq y$ to denote $(x, y) \in \leq$
Let $x < y \triangleq (x \leq y \wedge x \neq y)$

Definition 4.2

A poset $(X, \leq)$ is a set equipped with partial order $\leq$ on X

Example 4.2

$(\mathbb{N}, \leq)$

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?

Abstracting and concretization function

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

Definition 4.3

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

Definition 4.4

A concretization function $\gamma : D \to C$ *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 4.3

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 $\alpha(c) \triangleq [\mathit{inf}(c), \mathit{sup}(c)]$ *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.

Topic 4.3

Examples of abstraction

Sign abstraction

Sign abstraction

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

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

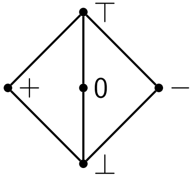
$\alpha(p) = +$ if $min(p) > 0$

$\alpha(p) = -$ 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

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

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

$D = \bot \cup \mathfrak{p}(P)$ // $\emptyset$ 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 $P = \{p_1, \dots, p_n\}$

$C = \mathfrak{p}(\mathbb{Q}^{|V|})$
$D =$ 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:
$V = \{\mathtt{x}, \mathtt{y}\}$
$P = \{\mathtt{x} \leq 1, -\mathtt{x} - \mathtt{y} \leq -1, \mathtt{y} \leq 5\}$
$\alpha(-2x - y \leq -2) = -\mathtt{x} - \mathtt{y} \leq -1 \vee \neg(\mathtt{x} \leq 1)$

End of Lecture 4