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 $X_\ell$ 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.

$$X_{\ell_0} = \top$$

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

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 $X^i_\ell \triangleq$ value of $X_\ell$ at $i$th iteration.

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

$$X^{k+1}_{\ell_0} = \top$$

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

If $\forall \ell. \ X^k_\ell = X^{k+1}_\ell$, 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:

\[
\begin{align*}
x & := 0 \\
& \quad \xleftarrow{+ +} \\
x & < 0
\end{align*}
\]

**Fixed point equations:**

\[
\begin{align*}
X_{\ell_0} & = \top \\
X_{\ell_1} & = sp(X_{\ell_0}, x' = 0) \lor sp(X_{\ell_1}, x' = x + 1) \\
X_{\ell_e} & = sp(X_{\ell_1}, x < 0 \land x' = x)
\end{align*}
\]

**Iterates:**

\[
\begin{align*}
X^0_{\ell_0} & := \top, X^0_{\ell_1} := \bot, X^0_{\ell_e} := \bot \\
X^1_{\ell_0} & := \top, X^1_{\ell_1} := (x = 0), X^1_{\ell_e} := \bot \\
X^2_{\ell_0} & := \top \\
X^2_{\ell_1} & := X^1_{\ell_1} \lor sp(X^1_{\ell_1}, x' = x + 1) \lor sp(X^1_{\ell_0}, x' = 0) \\
& := (x = 0) \lor sp(x = 0, x' = x + 1) \lor sp(\top, x' = 0) \\
& := (x = 0 \lor x = 1 \lor x = 0) := (0 \leq x \leq 1) \\
X^2_{\ell_e} & := sp(X^1_{\ell_1}, x < 0 \land x' = x) \\
& := sp(x = 0, x < 0 \land x' = x) := \bot
\end{align*}
\]
Example: diverging analysis with \( sp(\text{contd.}) \)

Iterates (contd.):

\[
x := 0
\]
\[
\ell_0 
\]
\[
x + +; \quad \ell_1
\]
\[
x < 0 \quad \ell_e
\]

\[
X^3_{\ell_0} := \top, \quad X^3_{\ell_1} := (0 \leq x \leq 2), \quad X^3_{\ell_e} := \bot
\]
\[
\vdots
\]
\[
X^n_{\ell_0} := \top, \quad X^n_{\ell_1} := (0 \leq x \leq n - 1), \quad X^n_{\ell_e} := \bot
\]

...will never converge

How to compute fixed point effectively?
Abstract post $sp\#$

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 \textit{sp} by \textit{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

$$X^{k+1}_{\ell_0} = \top$$

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

After convergence, $X_\ell$ will be a superset of reachable states at $\ell$. 
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 $x < y \overset{\Delta}{=} (x \leq y \land x \neq y)$

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** — $C = \text{sets of valuations} \subseteq \mathbb{Q}^V$
  - 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

Definition 4.3
An abstraction function $\alpha : C \rightarrow D$ maps each set $c \in C$ to $\alpha(c) \supseteq c$.

Definition 4.4
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 $Q^V$
Example: abstraction – intervals

Example 4.3

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

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

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

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 $\alpha$ 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 $D$, $\alpha$, and $\gamma$

Now on we will ignore that $D$ is set of sets. We assume $D$ is a topped poset $(D, \subseteq, T)$

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

Therefore,

$$(C, \subseteq) \xleftrightarrow{\gamma} (D, \subseteq)$$

We always choose $D$, $\alpha$, and $\gamma$ such that the above galois connection holds.
Topic 4.3

Examples of abstraction
Sign abstraction

\[ C = p(Q) \]
\[ D = \{ +, -, 0, \perp, \top \} \]
\[ \alpha(p) = + \text{ if } \min(p) > 0 \]
\[ \alpha(p) = - \text{ if } \max(p) < 0 \]
\[ \alpha(0) = 0 \]
\[ \alpha(\emptyset) = \perp \]
\[ \alpha(p) = \top, \text{ otherwise} \]
Congruence abstraction

\[ C = \mathbb{Z} \]
\[ D = \{0, \ldots, n - 1\} \]
\[ \alpha(c) = c \mod n \]
Cartesian predicate abstraction

Cartesian predicate abstraction is defined by a set of predicates

\[ P = \{ p_1, \ldots, p_n \} \]
\[ C = p(\mathbb{Q} | \mathbb{V}) \]
\[ D = \bot \cup p(P) \quad // \quad \emptyset \text{ represents } \top \]
\[ \bot \subseteq S_1 \subseteq S_2 \text{ if } S_2 \subseteq S_1 \]
\[ \alpha(c) = \{ p \in P \mid c \Rightarrow p \} \]
\[ \gamma(S) = \bigwedge S \]

Example:
\[ \mathbb{V} = \{ x, y \} \]
\[ P = \{ x \leq 1, -x - y \leq -1, y \leq 5 \} \]
\[ \alpha(\{(0, 0)\}) = \{ x \leq 1, y \leq 5 \} \]
\[ \alpha((x - 1)^2 + (y - 3)^2 = 1) = \{ -x - y \leq -1, y \leq 5 \} \]
Boolean predicate abstraction

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

\[ C = \mathbf{p}(\mathbf{Q}|\mathbf{V}|) \]

\[ D = \text{boolean formulas over predicates in } P \]

\[ F_1 \sqsubseteq F_2 \text{ if } F_1 \Rightarrow F_2 \]

\[ \alpha(c) = \text{strongest boolean formula over } P \text{ that contains } c \]

\[ \gamma(F) = F \]

Example:
\[ V = \{ x, y \} \]
\[ P = \{ x \leq 1, -x - y \leq -1, y \leq 5 \} \]
\[ \alpha(-2x - y \leq -2) = -x - y \leq -1 \lor \neg(x \leq 1) \]
End of Lecture 4