Abstract Interpretation

Introduction

The theory of abstract interpretation was introduced by Cousot and Cousot (POPL'77); it has been and still is being used in many different settings, ranging from compiler optimization to language semantics analysis, formal verification, and theorem proving.

From the POPL'77 paper:

“A program denotes computations in some universe of objects. Abstract interpretation of programs consists in using that denotation to describe computations in another universe of abstract objects, so that the results of abstract execution give some information about the actual computations.”
**Abstract Interpretation -- more quotes**

Cousot & Cousot, Journal of Logic and Computation, 1992:

“Abstract interpretation is a method for designing approximate semantics of programs which can be used to gather information about programs in order to provide sound answers to questions about their runtime behaviors. These semantics can then be used to design manual proof methods or to specify automatic program analyses.”

---

**Abstract interpretation -- basics**

Given:
- a concrete system with concrete (standard) semantics
- some notion of the properties we are interested in

We have to choose / construct:
1. Abstract domain
2. Correspondence between abstract and concrete objects
3. Abstract semantics

---

Cousot & Cousot, 1992:

“Theoretical point of view: The purpose of abstract interpretation is to design hierarchies of interrelated semantics at various levels of detail.”

“Practical point of view: The purpose of abstract interpretation is to design automatic program analysis tools for determining statically dynamic properties of programs.”
Abstract interpretation -- a simple example

Concrete system: multiplication of integers

Question: are the results of these multiplications less than, equal to, or greater than zero?

Concrete domain: sets of integers $\Sigma = 2^\mathbb{Z}$

Extend the semantics of multiplication to multiplication of sets:

$$S_1 \times S_2 = \{ n : \exists n_1 \in S_1, n_2 \in S_2 \land n_1 \times n_2 = n \}$$

Example: $\{1, 2\} \times \{3, 4\} = \{3, 4, 6, 8\}$

Abstract interpretation -- basics

We have to choose / construct:

1. Abstract domain
2. Correspondence between abstract and concrete objects
3. Abstract semantics

Abstract interpretation -- a simple example

Question: are the results of these multiplications less than, equal to, or greater than zero?

1. Abstract domain: $\Sigma_A = \{\text{neg, zero, pos}\}$

We have the following possibilities for $\Sigma_A$:

- $\Sigma_A = \{-1, 0, 1\}$
- $\Sigma_A = \{\ast, \equiv, \neq\}$
- $\Sigma_A = \{\triangledown, \triangledown, \#\}$
- $\Sigma_A = \{\odot, \bigotimes, \bigodot\}$
Abstract interpretation -- a simple example

Question: are the results of these multiplications less than, equal to, or greater than zero?

1. Abstract domain: \( \Sigma_A = \{ \text{neg, zero, pos} \} \)

Choose an abstract domain

Question: are the results of these multiplications less than, equal to, or greater than zero?

1. Abstract domain: \( \Sigma_A = \{ \text{neg, zero, pos} \} \)

Abstract interpretation -- a simple example

Concretization function

\( \gamma: \Sigma_A \rightarrow \Sigma \)

maps abstract objects to concrete objects

gives meaning to the abstract objects

\[ \begin{align*}
\gamma(\text{neg}) &= \{ n \in \mathbb{Z} \mid n < 0 \} \\
\gamma(\text{zero}) &= \{ 0 \} \\
\gamma(\text{pos}) &= \{ n \in \mathbb{Z} \mid n > 0 \}
\end{align*} \]

Abstraction function

\( \alpha: \Sigma \rightarrow \Sigma_A \)

maps concrete objects into abstract objects

\[ \begin{align*}
\alpha(\{0\}) &= \text{zero} \\
\alpha(\mathbb{S}) &= \text{neg} \quad \text{if} \quad \forall n \in \mathbb{S} \cdot n < 0 \\
\alpha(\mathbb{S}) &= \text{pos} \quad \text{if} \quad \forall n \in \mathbb{S} \cdot n > 0 \\
\alpha(\mathbb{S}) &= \text{?} \quad \text{otherwise ??}
\end{align*} \]
Abstraction function

\[ \alpha : \Sigma \rightarrow \Sigma_A \]

maps concrete objects into abstract objects

- \( \alpha(\{0\}) = \text{zero} \)
- \( \alpha(S) = \text{neg} \) if \( \forall n \in S . n < 0 \)
- \( \alpha(S) = \text{pos} \) if \( \forall n \in S . n > 0 \)
- \( \alpha(S) = T \) otherwise

Abstraction function

\[ \alpha : \Sigma \rightarrow \Sigma_A \]

maps concrete objects into abstract objects

- \( \alpha(\{0\}) = \text{zero} \)
- \( \alpha(S) = \text{neg} \) if \( \forall n \in S . n < 0 \)
- \( \alpha(S) = \text{pos} \) if \( \forall n \in S . n > 0 \)
- \( \alpha(S) = T \) otherwise

Concretization function

\[ \gamma : \Sigma_A \rightarrow \Sigma \]

maps abstract objects to concrete objects

gives meaning to the abstract objects

- \( \gamma(T) = \mathbb{Z} \)
- \( \gamma(\text{zero}) = \{ n \in \mathbb{Z} \mid n < 0 \} \)
- \( \gamma(\text{pos}) = \{ n \in \mathbb{Z} \mid n > 0 \} \)
- \( \gamma(\emptyset) = \emptyset \)
Abstraction and Concretization function

\[ \alpha : \Sigma \rightarrow \Sigma_A \]

abstraction

\[ \gamma : \Sigma_A \rightarrow \Sigma \]

concretization

\[
\begin{align*}
\alpha(\emptyset) &= \bot \quad \text{if } \emptyset \subseteq \emptyset \\
\alpha(S) &= \text{neg} \quad \text{if } \forall n \in S . n < 0 \\
\alpha(\{0\}) &= \text{zero} \\
\alpha(S) &= \text{pos} \quad \text{if } \forall n \in S . n > 0 \\
\alpha(S) &= \top \quad \text{otherwise}
\end{align*}
\]

\[
\gamma(\bot) = \emptyset \\
\gamma(\text{neg}) = \{ n \in \mathbb{Z} | n < 0 \} = \mathbb{Z}^- \\
\gamma(\text{zero}) = \{ 0 \} \\
\gamma(\text{pos}) = \{ n \in \mathbb{Z} | n > 0 \} = \mathbb{Z}^+ \\
\gamma(\top) = \mathbb{Z}
\]

Abstract version of multiplication

Concrete multiplication: \( x_C : \Sigma \times \Sigma \rightarrow \Sigma \)

Example: \( \{1, 2\} \cdot \{3, 4\} = \{3, 4, 6, 8\} \)

Abstract multiplication: \( x_A : \Sigma_A \times \Sigma_A \rightarrow \Sigma_A \)

Abstraction function

\[
\Sigma = 2^\mathbb{Z} \\
\{3, 4\} \\
\{7, -12, -42\} \\
\emptyset \\
\mathbb{Z}^- \\
\mathbb{Z}^+ \\
\{0\} \\
\{2, 0\} \\
\{2, 0, 14\} \\
\{2, 0, 14, 27\} \\
\{0\}
\]

size: 5

size: uncountable

Concretization function

\[
\begin{align*}
\Sigma &= \{3, 4\} \\
\{7, -12, -42\} \\
\emptyset \\
\mathbb{Z}^- \\
\mathbb{Z}^+ \\
\{0\} \\
\{2, 0\} \\
\{2, 0, 14\} \\
\{2, 0, 14, 27\} \\
\{0\}
\end{align*}
\]

size: uncountable

size: 5
Abstract version of multiplication

Abstract multiplication: \( x_A : \Sigma_A \times \Sigma_A \rightarrow \Sigma_A \)

<table>
<thead>
<tr>
<th>( x_A )</th>
<th>( \bot )</th>
<th>neg</th>
<th>zero</th>
<th>pos</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \bot )</td>
<td>( \bot )</td>
<td>( \bot )</td>
<td>( \bot )</td>
<td>( \bot )</td>
<td>( \bot )</td>
</tr>
<tr>
<td>neg</td>
<td>( \bot )</td>
<td>pos</td>
<td>zero</td>
<td>neg</td>
<td>T</td>
</tr>
<tr>
<td>zero</td>
<td>( \bot )</td>
<td>zero</td>
<td>zero</td>
<td>zero</td>
<td></td>
</tr>
<tr>
<td>pos</td>
<td>( \bot )</td>
<td>pos</td>
<td>zero</td>
<td>zero</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>( \bot )</td>
<td>T</td>
<td>zero</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

Abstract analysis

Concrete question: \( n_1 \times n_2 \overset{?}{=} 0 \)

Procedure:

Abstract \( n_1 \) and \( n_2 \):
\( n_1^A = \alpha ( \{ n_1 \} ) \)
\( n_2^A = \alpha ( \{ n_2 \} ) \)

Perform abstract multiplication:
\( n^A = n_1^A \times n_2^A \)

Concretize \( n^A \):
\( S = Y ( n^A ) \)
- If \( S = Z^+ \) then \( n_1 \times n_2 > 0 \)
- If \( S = Z^- \) then \( n_1 \times n_2 < 0 \)
- If \( S = \{ 0 \} \) then \( n_1 \times n_2 = 0 \)
- If \( S = Z \) then we don’t know

Abstract analysis -- Example

\( n_1 = 783,422 \)
\( n_2 = 409,312 \)

\[ n_1 \times n_2 \overset{?}{=} 0 \]

Abstract \( n_1 \) and \( n_2 \):
\( n_1^A = \alpha ( \{ n_1 \} ) = \text{pos} \)
\( n_2^A = \alpha ( \{ n_2 \} ) = \text{pos} \)

Perform abstract multiplication:
\( n^A = n_1^A \times n_2^A \)
\( = \text{pos} \times \text{pos} = \text{pos} \)

Concretize \( n^A \):
\( S = Y ( n^A ) = Z^+ \)

- If \( S = Z^+ \) then \( n_1 \times n_2 > 0 \)

Conclusion: \( 783,422 \times 409,312 > 0 \)

Abstract analysis -- Observations

- The choice of abstract domain was governed by the question. If the question had been to determine whether the result was even or odd, we would have chosen a different abstract domain and abstract semantics.

- The concrete domain is a partially ordered set with the subset relation \( \subseteq \) as order.

- We can also impose an order \( \langle^A \) on the abstract domain:

\[ \bot \langle^A \neg \langle^A \neg \langle^A \bot \langle^A \top \]

\[ \bot \langle^A \neg \langle^A \neg \langle^A \bot \langle^A \top \]

\[ \bot \langle^A \neg \langle^A \neg \langle^A \bot \langle^A \top \]

\[ \bot \langle^A \neg \langle^A \neg \langle^A \bot \langle^A \top \]
Abstract analysis -- Observations

• $\alpha$ and $\gamma$ are both monotone:

\[
S_i \subseteq S_2 \rightarrow \alpha(S_i) \leq^A \alpha(S_2)
\]
\[
a_1 \leq^A a_2 \rightarrow \gamma(a_1) \subseteq \gamma(a_2)
\]

Example:

\{0\} \subseteq \{0, 1, 2\}
\alpha(\{0\}) = \text{zero}
\alpha(\{0, 1, 2\}) = \text{T}
\text{zero} \leq^A \text{T}
\alpha(S) = \perp \quad \text{if } S = \emptyset
\alpha(\{0\}) = \text{zero}
\alpha(S) = \text{neg} \quad \text{if } \forall n \in S \cdot n < 0
\alpha(S) = \text{pos} \quad \text{if } \forall n \in S \cdot n > 0
\alpha(S) = \text{T} \quad \text{otherwise}

Example:

$\gamma(\text{zero}) = \{0\}$
$\gamma(\text{T}) = \mathbb{Z}$
$\{0\} \subseteq \mathbb{Z}$
$\alpha(\gamma(a)) = a$

Example:

$a = \text{pos}$
$\gamma(a) = \mathbb{Z}^+$
$\alpha(\gamma(a)) = \text{pos}$
Abstract analysis -- Observations

- Abstract multiplication over-approximates

\[ \gamma( a_1 ) \times \gamma( a_2 ) \subseteq \gamma( a_1 \times^a a_2 ) \]

( in this case it is actually equal )

\[
\begin{array}{c|cccccc}
  x_A & \bot & \text{neg} & \text{zero} & \text{pos} & T \\
  \hline
  \bot & \bot & \bot & \bot & \bot & \bot \\
  \text{neg} & \bot & \text{pos} & \text{zero} & \text{neg} & T \\
  \text{zero} & \bot & \text{zero} & \text{zero} & \text{zero} & \text{zero} \\
  \text{pos} & \bot & \text{neg} & \text{zero} & \text{pos} & T \\
  T & \bot & T & \text{zero} & T & T \\
\end{array}
\]

we don't lose anything by doing abstract multiplications

Example:

\[ \gamma( \text{pos} ) \times \gamma( \text{pos} ) = Z^+ \times Z^+ = Z^+ \]

\[ \gamma( \text{pos} ) = Z^+ \]

\[
\begin{array}{c|cccccc}
  x_A & \bot & \text{neg} & \text{zero} & \text{pos} & T \\
  \hline
  \bot & \bot & \bot & \bot & \bot & \bot \\
  \text{neg} & \bot & \text{pos} & \text{zero} & \text{neg} & T \\
  \text{zero} & \bot & \text{zero} & \text{zero} & \text{zero} & \text{zero} \\
  \text{pos} & \bot & \text{neg} & \text{zero} & \text{pos} & T \\
  T & \bot & T & \text{zero} & T & T \\
\end{array}
\]

Abstract analysis -- Observations

\[ \gamma( a_1 ) \times \gamma( a_2 ) = \gamma( a_1 \times^a a_2 ) \]

Example:

\[ \gamma( \text{pos} ) \times \gamma( \text{pos} ) = Z^+ \times Z^+ = Z^+ \]

\[ \gamma( \text{pos} ) = Z^+ \]

\[
\begin{array}{c|cccccc}
  x_A & \bot & \text{neg} & \text{zero} & \text{pos} & T \\
  \hline
  \bot & \bot & \bot & \bot & \bot & \bot \\
  \text{neg} & \bot & \text{pos} & \text{zero} & \text{neg} & T \\
  \text{zero} & \bot & \text{zero} & \text{zero} & \text{zero} & \text{zero} \\
  \text{pos} & \bot & \text{neg} & \text{zero} & \text{pos} & T \\
  T & \bot & T & \text{zero} & T & T \\
\end{array}
\]

Galois connection

Let \(( \Sigma_A , \preceq^a )\) and \(( \Sigma , \subseteq )\) be partially ordered sets.

A pair \(( \alpha , \gamma )\) is a Galois connection if the following hold:

1. \( \alpha : \Sigma \rightarrow \Sigma_A \) and \( \gamma : \Sigma_A \rightarrow \Sigma \)
2. \( \alpha \) and \( \gamma \) are monotone
3. \( S \subseteq \gamma( \alpha( S ) ) \) for all \( S \in \Sigma \) and

\[\alpha( \gamma( a ) ) \preceq^a a\]

for all \( a \in \Sigma_A \)

Note: if \( \alpha( \gamma( a ) ) = a \) then \(( \alpha , \gamma )\) is called a Galois insertion
Galois connection

The functions $\alpha$ and $\gamma$ determine each other: if one is given, the other follows.

Given $\gamma$:

$\alpha( S )$ is the smallest object in $\Sigma_A$ that represents all of $S$:

$$\alpha( S ) = \inf \{ a \in \Sigma_A \mid S \subseteq \gamma( a ) \}$$

$$= \bigwedge^A \{ a \in \Sigma_A \mid S \subseteq \gamma( a ) \} \text{ (meet)}$$

Example: $S = \{ 3, 4 \}$

$S \subseteq \gamma( T ) \quad S \subseteq \gamma( \text{pos} )$

$\alpha( \{ 3, 4 \} ) = \inf \{ \text{pos} , T \} = \text{pos}$

Galois connection

Given $\gamma$:

$\alpha( S )$ is the smallest object in $\Sigma_A$ that represents all of $S$:

$$\alpha( S ) = \inf \{ a \in \Sigma_A \mid S \subseteq \gamma( a ) \}$$

$$= \bigwedge^A \{ a \in \Sigma_A \mid S \subseteq \gamma( a ) \} \text{ (meet)}$$

Given $\alpha$:

$\gamma( a )$ is the largest object in $\Sigma$ that is fully described by $a$:

$$\gamma( a ) = \sup \{ S \in \Sigma \mid \alpha( S ) \leq^A a \}$$

$$= \bigcup \{ S \in \Sigma \mid \alpha( S ) \leq^A a \} \text{ (join)}$$

Example:

$\alpha( \{ 3, 4 \} ) \leq^A \text{pos}$

$\alpha( \{ 17, 32, 42 \} ) \leq^A \text{pos}$

$\gamma( \text{pos} ) = \{ 3, 4 \} \cup \{ 17, 32, 42 \} \cup \ldots \quad = \mathbb{Z}^+$