Correctness of Abstract Interpretation

Scribe: Ross Fleming, Frances Spalding

Lecturer: Radu Rugina

1 Correctness of Analysis

An analysis such as sign analysis ($L_a = Var \to \mathcal{P}(sign)$) is an approximation of a concrete domain ($L_c = Var \to \mathcal{P}(\mathbb{Z})$). The abstraction and concretization functions, α and γ, are conservative approximations. Correctness is maintained by insuring changes in the abstract information correspond to changes in the Operational/Denotational semantics.

$$
\begin{tikzcd}
i_a \arrow[r, "{[\![c]\!]^l}"] \arrow[d, dash, dashed] & i'_a \arrow[d, dash, dashed] \\
\sigma \arrow[r, "{\mathcal{C}[\![c]\!]}"] & \sigma'
\end{tikzcd}
$$

abstract information $i_a \overset{?}{\longleftrightarrow}$ state σ

$$\sigma \in \Sigma = Var \to \mathbb{Z}$$

- Σ and L_c are both about concrete values, but Σ characterizes execution points (eg. the head of a loop after completion of one iteration) while L_c characterizes program points (eg. the head of a loop). L_c characterizes all iterations so we can consider it to be a set of states.
- We take $\sigma \in \Sigma$ and construct $\sim \{\sigma\}$. To do this we construct a beta function: $\beta(\sigma) = \alpha(\lambda x \in Var.\{\sigma(x)\}) \in L_a$
- $i_a \in L_a$ safely (conservatively) approximates σ if $\beta(\sigma) \sqsubseteq i_a$. It is sufficient to construct $\beta(\sigma) = i_a$.

$$
\begin{tikzcd}
i_a \arrow[r] & i'_a \\
 & \rotatebox{90}{$\sqsubseteq$} \\
\sigma \arrow[uu, "\beta"] \arrow[r, "{\mathcal{C}[\![c]\!]}"] & \sigma' \arrow[u, "\beta"']
\end{tikzcd}
$$

2 Soundness (Correctness)

$$\forall c, \sigma, i_a \quad \beta(\sigma) \sqsubseteq i_a \Rightarrow \beta(\mathcal{C}[\![c]\!]\sigma) \sqsubseteq [\![c]\!]^l i_a$$

An alternative definition that is sufficient since the transfer functions are monotonic:

$$\beta(\mathcal{C}[\![c]\!]\sigma) \sqsubseteq [\![c]\!]^l(\beta(\sigma))$$

$$
\begin{tikzcd}
i_a \arrow[r] & i'_a \\
 & \rotatebox{90}{$\sqsubseteq$} \\
\sigma \arrow[uu, "\beta"] \arrow[r, "{\mathcal{C}[\![c]\!]}"] \arrow[uur, bend left=60] \arrow[uur, bend right=60] & \sigma' \arrow[u, "\beta"']
\end{tikzcd}
$$

3 Example: Soundness of Sign Analysis

Recall Sign Analysis from lecture on 12/2/2002. We used sign analysis to determine the possible signs of each variable at each point in the program.

The lattice L_a for the abstract domain is:

$$L_a = Var \to \mathcal{P}(Sign)$$

β is similar to the abstraction function α, except that its domain is the set of states, Σ, instead of concrete information, L_c.

$$\begin{aligned} \beta &: \Sigma \to L_a \\ \beta(\sigma) &= \lambda x \in Var \;.\; \{sign'(\sigma(x))\} \end{aligned}$$

Recall the following definitions from sign analysis:

$$\begin{aligned} Sign &= \{+, 0, -\} \\ sign &: Aexp \times L_a \to \mathcal{P}(Sign) \\ sign' &: \mathbb{Z} \to Sign \\ [\![x := a]\!]^\ell &: L_a \to L_a \\ [\![x := a]\!]^\ell s &= s[x \to sign(a, s)] \end{aligned}$$

where $[\![s]\!]$ is the transfer function, and $s : Var \to Sign$ is a sign mapping.

And as we know, the denotational semantics for assignment state that:

$$\mathcal{C}[\![x := a]\!]\sigma = \sigma[x \to \mathcal{C}[\![a]\!]\sigma]$$

In order to prove the soundness of sign analysis, we will use a combination of the above facts. Formally, we want to prove:

$$\beta(\mathcal{C}[\![x := a]\!]\sigma) \sqsubseteq [\![c]\!]^\ell(\beta(\sigma))$$

Using the denotation of assignment $\mathcal{C}[\![x := a]\!]\sigma = \sigma[x \to \mathcal{C}[\![a]\!]\sigma]$ and $[\![x := a]\!]^\ell s = s[x \to sign(a, s)]$, this becomes:

$$\beta(\sigma[x \to \mathcal{C}[\![a]\!]\sigma]) \sqsubseteq \beta(\sigma)[x \to sign(a, B(\sigma))]$$

However, we are only interested in the value of x, since the assignment of a to x clearly won't affect any values other than x. Therefore, it is enough to prove:

$$\{sign'(\mathcal{C}[\![a]\!]\sigma)\} \sqsubseteq sign(a, \beta(\sigma))$$

We will prove this using induction on the structure of a.

- $a = n$:
  $\{sign'(\mathcal{C}[\![n]\!]\sigma)\} \sqsubseteq sign(n, \beta(\sigma)) \Rightarrow \{sign'(n)\} \sqsubseteq sign(n, \beta(\sigma))$. The sign of n is not dependent on $\beta(\sigma)$, and will return the singleton set containing the sign of n. $\{sign'(n)\}$ will be the same set.

- $a = x$:
  $\{sign'(\mathcal{C}[\![x]\!]\sigma)\} = \{sign'(\sigma(x))\}$.
  $sign(x, \beta(\sigma)) = \beta(\sigma)(x) = \{sign'(\sigma(x))\}$
  So therefore, $\{sign'(\mathcal{C}[\![x]\!]\sigma)\} = sign(x, \beta(\sigma))$

- $a = a_0 + a_1$:
  $\{sign'(\mathcal{C}[\![a_0 + a_1]\!]\sigma)\} \sqsubseteq sign(a_0 + a_1, \mathcal{B}(\sigma))$
  $\{sign'(\mathcal{C}[\![a_0]\!]\sigma + \mathcal{C}[\![a_1]\!]\sigma)\} \sqsubseteq sign(a_0, \mathbb{B}(\sigma) +_s sign(a_1, \beta(\sigma))$
  Recall the definition of $+_s$ from two lectures ago:

  $$s_1 +_s s_2 = \cup_{x \in s_1, y \in s_2} \quad x \oplus y$$

$\oplus$	$-$	0	$+$
$-$	$\{-\}$	$\{-\}$	$\{-,0,+\}$
0	$\{-\}$	$\{0\}$	$\{+\}$
$+$	$\{-,0,+\}$	$\{+\}$	$\{+\}$

  By the induction hypothesis, we know that:

  $$\{sign'(\mathcal{C}[\![a_0]\!]\sigma)\} \sqsubseteq sign(a_0, \beta(\sigma))$$

  $$\{sign'(\mathcal{C}[\![a_1]\!]\sigma)\} \sqsubseteq sign(a_1, \beta(\sigma))$$

  $$\{sign'(n_0)\} \sqsubseteq i_0$$

  $$\{sign'(n_1)\} \sqsubseteq i_1$$

  Therefore, we need to show that $\{sign'(n_0 + n_1)\} \sqsubseteq i_0 +_s i_1$. We will break this down depending on the 9 possible relationships between n_0 and n_1.

  - $n_0 > 0, n_1 = 0$:
    $n_0 > 0 \;\Rightarrow\; \{sign'(n_0)\} = \{+\} \sqsubseteq i_0$
    $n_1 = 0 \;\Rightarrow\; \{sign'(n_1)\} = \{0\} \sqsubseteq i_1$
    According to the above definition, $+ \oplus 0 \sqsubseteq i_0 + i_1 \;\Rightarrow\; \{+\} \sqsubseteq i_0 + i_1$.
    Also, $n_0 > 0 \wedge n_1 = 0 \;\Rightarrow\; n_0 + n_1 > 0 \;\Rightarrow\; \{sign'(n_0 + n_1)\} = \{+\}$.
    Therefore, we have $\{+\} \sqsubseteq \{+\}$, and so $\{sign'(n_0 + n_1)\} \sqsubseteq i_0 +_s i_1$.
  - $n_0 > 0, n_1 < 0$:
    We get $+ \oplus - \sqsubseteq i_0 + i_1 \;\Rightarrow\; \{+,0,-\} \sqsubseteq i_0 + i_1$, and because $\{+,0,-\} = \top$, we get $\{sign'(n_0 + n_1)\} \sqsubseteq \{+,0,-\} = i_0 + i_1$.
  - The remaining seven cases are left as an exercise.

- The remaining cases are left as an exercise.