Lecture notes – Correctness of abstract interpretation

September 1, 2022

Slide 3/19. Collecting semantics stated as an abstract interpretation. This slide points out that if we use the 2^{State} lattice, with set-union as join, and the *nstate'* functions as the transfer functions of statements, then the abstract JOP at each point using this 3-tuple is nothing but the collecting semantics at that point. In other words, the collecting semantics is nothing but the abstract JOP using the concrete lattice and concrete transfer functions.

Slide 4. This slide defines the notion of one abstract interpretation (D, F_D, γ_D) being a *consistent abstraction* of another abstract interpretation (C, F_C, γ_C). Here, F_D is shorthand for the set of *all* transfer functions in the first A.I., while F_C is shorthand for the set all transfer functions in the second framework. The definition is based on a given pair of monotonic functions $(\alpha_{CD}, \gamma_{DC})$ such that this pair of functions form a *Galois connection* (to be defined later). Note that γ_{DC} is different from γ_D and from γ_C.

Consider a program P with N program points. Given lattice D, $\overline{D}$ represents a lattice of *vectors*, with each vector having N elements from D. The $\leq$ relation on $\overline{D}$ is nothing but pointwise ordering; i.e., if $\overline{d_1}, \overline{d_2} \in \overline{D}$, then $\overline{d_1} \leq \overline{d_2}$ iff for all i in $[1 \ldots N]$, $\overline{d_1}[i] \leq \overline{d_2}[i]$. Similarly, $\overline{C}$ represents the vectorized version of C. $\overline{\gamma_{DC}}(\overline{d_1})$ is nothing but γ_{DC} applied individually to each element in the vector $\overline{d_1}$ to result in a vector belonging to $\overline{C}$. We also define $\overline{\alpha_{CD}}(\overline{c_1})$ analogously.

Let $c_0 \in C$ and $d_0 \in D$ be some two elements. Let $\text{JOP}^{c_0}_{\overline{C}}$ denote the abstract JOP at all points in P using the C-framework and using c_0 as the initial abstract state. Let $\text{JOP}^{d_0}_{\overline{D}}$ denote the abstract JOP at all points in P using the D-framework and using d_0 as the initial abstract state. The definition is that the D-framework is a consistent abstraction of the C-framework under the given pair $(\alpha_{CD}, \gamma_{DC})$ if $\text{JOP}^{c_0}_{\overline{C}} \leq \gamma_{DC}(\text{JOP}^{d_0}_{\overline{D}})$ for any c_0 and d_0 chosen such that $\gamma_{DC}(d_0) \geq c_0$. Intuitively, what this means is that at any program point the γ_{DC} image of the JOP computed using the D-framework dominates the JOP computed at the same point using the C-framework.

The C-framework is said to be *more precise* than the D-framework under $(\alpha_{CD}, \gamma_{DC})$ iff the D-framework is a consistent abstraction of the C-framework under γ_{DC}.

(From here on, we use γ to mean γ_{DC} and α to mean α_{CD}.)

Slide 6. The notion of two functions (α, γ) forming a Galois Connection is defined here.

Generally C is a larger (more precise) lattice than D. The basic property that the Galois Connection enforces is that each element of D *represents* one or more elements of C. Each element c_1 of C that is in the *range* of γ_{DC} is *represented precisely* by the element $\alpha(c_1)$, as $\gamma(\alpha(c_1)) = c_1$. On the other hand, each element c_2 of C that is *not* in the range γ_{DC} is represented imprecisely (i.e., over-approximated) by $\alpha(c_2)$, as $\gamma(\alpha(c_2)) > c_2$.

Slide 7.

It can be shown that the given (α, γ) for this illustration form a Galois connection.

Note, to assert this consistent abstraction property, we need to also define the transfer functions for both domains. The f^{L_1} transfer functions are as defined in Slide 5 of the "Abstract interpretation" slides. For any statement n, the transfer function $f^{L_2}_n$ can be defined as follows:

$f^{L_2}_n(S) = \{(x, y) \mid (x', y') \in S, (x, y) \in$

$\gamma(f_n^{L_1}(x', y'))\}$

Note, in the "Introduction" slides we showed a program where this L_2 interpretation produces more precise results than the L_1 interpretation.

Slide 9. The first bullet brings out the relationship between the definitions of *consistent abstraction* and *correctness*. (Correctness of an abstract interpretation framework was defined earlier in the "Introduction to abstract interpretation" slides.)

The second bullet points out that "consistent abstraction of" is a transitive relation if the γ's are monotonic.

Slide 10. This slide gives the sufficient condition under which one A.I. is a consistent abstraction of another A.I. This is the main theorem of abstract interpretation.

Say a designer has proposed a new abstract lattice D, and a set of F_D transfer functions based on D. In order to prove that this proposed abstract interpretation is a consistent abstraction of some existing abstraction interpretation based on a lattice C, the designer of the D-abstract interpretation should prove that the F_D transfer functions that they have provided are abstractions of the corresponding F_C transfer functions. Note, if the designer wants to prove that the D-abstract interpretation is *correct*, then they need to show that the F_D transfer functions are abstractions of the corresponding *nstate'* transfer functions.

Slide 11. This slide gives the definition of a function $f_{MN,D}$ being an *abstraction* of a function $f_{MN,C}$.

The basic requirement, intuitively, is that for any pair of adjacent program points M, N, $f_{MN,D}$ over-approximates $f_{MN,C}$. This can be stated in two ways, that are provably equivalent to each other:

- for any $c \in C$, $\alpha(f_{MN,C}(c)) \leq f_{MN,D}(\alpha(c))$
- for any $d \in D$, $f_{MN,C}(\gamma(d)) \leq \gamma(f_{MN,D}(d))$

The two definitions above are shown pictorially in the slide.

We now discuss the sources of imprecision in abstract interpretation. Given an initial abstract state d_0, it corresponds to initial set of concrete states $\gamma(d_0)$. If MN is a pair of adjacent points (or is a straight-line path), the ideal collecting semantics that the user would like to obtain is $f_{MN,C}(\gamma(d_0))$. However, the abstract JOP reported would be $f_{MN,D}(d_0)$. Hence, the over-approximation of the collecting semantics that would be reported is $\gamma(f_{MN,D}(d_0))$. Note, even if a "most precise" abstract transfer function $f_{MN,D}$ is used, $\gamma(f_{MN,D}(d_0))$ would often be a strict over-approximation of $f_{MN,C}(\gamma(d_0))$. For e.g., this happens when d_0 is the CP fact $\{(x, 0)\}$ and MN is the true or false branch of the condition $x > 5$. This is the first major source of imprecision.

The second source of imprecision occurs when in a program there are multiple paths that reach a point N. If one ignores the first source of imprecision mentioned above, then the ideal solution one would expect from an approach would be:

$$\sqcup\{\gamma(d_p) \,|\, p \text{ is a path from } I \text{ to } N, d_p = f_{p,D}(d_0)\}$$

However, by using the abstract interpretation approach, one obtains the following result:

$$\gamma(\sqcup\{d_p \,|\, p \text{ is a path from } I \text{ to } N, d_p = f_{p,D}(d_0)\})$$

The result above is in general a strict over-approximation of the ideal solution.