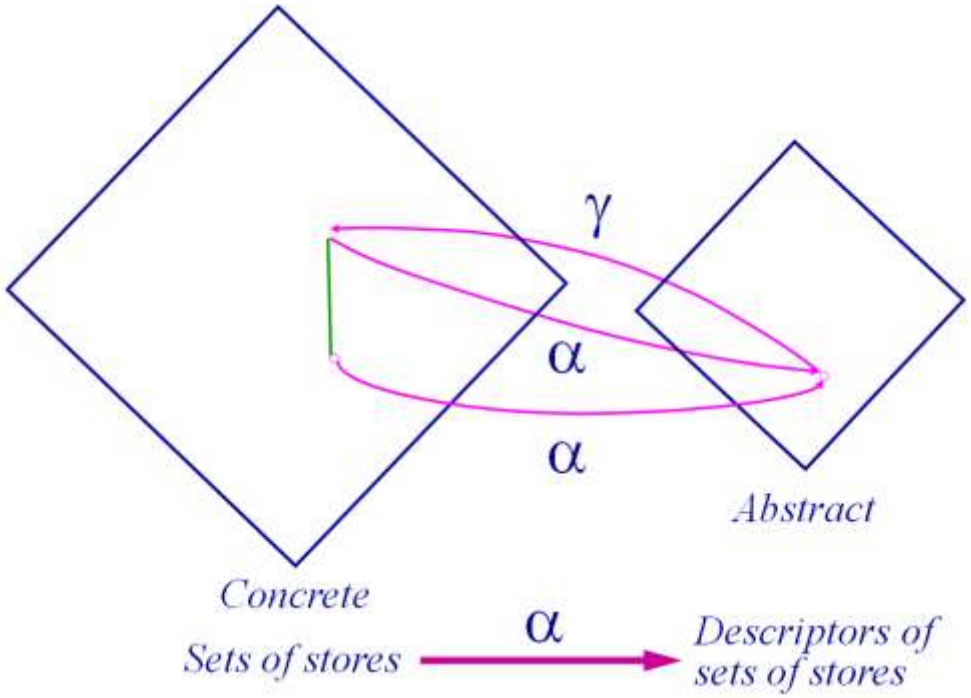
Program analysis - Prof. Mooly Sagiv

Lecture 4 – 14/11/12

Lital Biniashvili

Abstract interpretation

We will start from giving the definitions for the abstraction and concretization functions:



Abstraction function: (A special case of Constant Propagation)

The abstraction of an individual state:

$$\beta_{CP}: [\mathrm{Var} * \rightarrow Z] \rightarrow [\mathrm{Var} * \rightarrow Z \cup \{\bot, \top\}]$$
$$\beta_{CP}(\sigma) = \sigma$$

I.e. the abstraction of individual state is the same state, without any changes.

The abstraction of set of states:

$$\alpha_{CP}: P([\mathrm{Var} * \rightarrow Z]) \rightarrow [\mathrm{Var} * \rightarrow Z \cup \{\bot, \top\}]$$

$$\alpha_{CP}(CS) = \bigsqcup\{\beta_{CP}(\sigma) \mid \sigma \in CS\} = \bigsqcup\{\sigma \mid \sigma \in CS\}$$

I.e. the abstraction of set of states is the join of abstractions of each state in the set. In other words, abstraction of set of states gives us the lowest constant that contains all the states.

It's immediate from the definition that the function is monotonic (for a bigger set of states, the constant that includes all the states doesn't become smaller).

The abstraction function result is sound:

$$\alpha_{CP}(\text{Reach}(v)) \sqsubseteq df(v)$$

*Reach (v) = all the possible variables' values that are accessible from the state v.

*notice that the result we get from chaotic iterations is exactly df(v).

For example:

$$\alpha_{CP}([X \to 5, Y \to 7], [X \to 6, Y \to 7]) = [X \to ?, Y \to 7]$$

We can see in that example that we lose information when we use abstraction, because the result includes $\{x \to 22, y \to 7\}$, which is not in the original set of states.

The Concretization function: (A special case of Constant Propagation)

$$\gamma_{CP}: [Var * \to Z \cup \{\bot, \top\}] \to P([Var * \to Z])$$

$$\gamma_{CP}(df) = \{\sigma \mid \beta_{CP}(\sigma) \sqsubseteq df\} = \{\sigma \mid \sigma \sqsubseteq df\}$$

I.e. the Concretization function on a state s gives us all the concrete states that s can represent.

The concretization function is monotonic too – trivial.

The concretization function result is sound:

$$\text{Reach}(v) \subseteq \gamma_{CP}(df(v))$$

For instance:

$$\gamma_{\mathrm{CP}}([x \to 5, y \to ?]) = \{[x \to 5, y \to 1], [x \to 5, y \to 2], [x \to 5, y \to 3] \ldots.\}$$

In this example we can see the concretization function gives us more information, meaning that the result states are substantial.

Galois connections

Definition: Given lattices C and A, and functions α: C $\to$A and γ: A $\to$C, The pair of functions (α,γ) form Galois connection if:

- α and γ are monotone
- $\forall$ a $\in$ A: $\alpha(\gamma(a)) \sqsubseteq a$
- $\forall$ c $\in$ C: c $\sqsubseteq \gamma(\alpha(C))$

Alternatively if:

$\forall$ c $\in$ C $\forall$ a $\in$ A :

$\alpha(c) \sqsubseteq a$ iff c $\sqsubseteq \gamma(a)$

* The second definition meaning is that comparing element by α is equivalent to comparing element by γ.

* α and γ uniquely determine each other

Proof:

- Suppose that α and γ_1 forms a Galois connection and α and γ_2 forms a Galois connection.
- From the second definition we get $\alpha(c) \sqsubseteq a$ iff c $\sqsubseteq \gamma_1(a)$ iff c $\sqsubseteq \gamma_2(a)$.
- It holds that $\forall a \in A : \gamma_1(a) \sqsubseteq \gamma_1(a)$, so from the previous sentence we derive $\forall a \in A : \gamma_1(a) \sqsubseteq \gamma_2(a)$.
- It holds that $\forall a \in A : \gamma_2(a) \sqsubseteq \gamma_2(a)$, so from the previous sentence we derive $\forall a \in A : \gamma_2(a) \sqsubseteq \gamma_1(a)$.
- From $\forall a \in A : \gamma_2(a) \sqsubseteq \gamma_1(a)$ and $\forall a \in A : \gamma_1(a) \sqsubseteq \gamma_2(a)$ we get $\gamma_1 = \gamma_2$. i.e. γ determined uniquely by a.

The proof to "α determined uniquely by γ" is the same.

The abstraction and concretization functions form a Galois connection:

The abstraction and concretization functions in the constant propagation lattice fulfill the following conditions:

- Both functions are monotonic.
- $\forall$ df $\in$ [Var $*$ $\rightarrow$Z$\cup${$\perp$,$\top$}]

$$\alpha_{\mathrm{CP}}\ (\gamma_{\mathrm{CP}}\ (\mathrm{df})) \sqsubseteq \mathrm{df}$$

- $\forall$ c $\in$ P([Var$_*$$\rightarrow$Z])

$$c_{CP} \sqsubseteq \gamma_{\mathrm{CP}}\ (\alpha_{\mathrm{CP}}(\mathrm{C}))$$

We will illustrate the conditions with example:

Let $s = \{[x \rightarrow 5, y \rightarrow 7], [x \rightarrow 5, y \rightarrow 8]\}$

$$\gamma_{\mathrm{CP}}\left(\alpha_{\mathrm{CP}}(\mathrm{s})\right) = \gamma_{\mathrm{CP}}\ ([\mathrm{x} \rightarrow 5, \mathrm{y} \rightarrow \perp]) = \{[\mathrm{x} \rightarrow 5, \mathrm{y} \rightarrow 1], [\mathrm{x} \rightarrow 5, \mathrm{y} \rightarrow 2], \dots\}$$

And it holds:

$$\{[x \rightarrow 5, y \rightarrow 7], [x \rightarrow 5, y \rightarrow 8]\} \sqsubseteq \{[\mathrm{x} \rightarrow 5, \mathrm{y} \rightarrow 1], [\mathrm{x} \rightarrow 5, \mathrm{y} \rightarrow 2], \dots\}$$

And the third condition is fulfilled.

Let $df = [x \rightarrow 7, y \rightarrow ?\,]$

$$\alpha_{\mathrm{CP}}\ \left(\gamma_{\mathrm{CP}}\ (\mathrm{df})\right) = \alpha_{\mathrm{CP}}\ (\{\dots, [\mathrm{x} \rightarrow 7, \mathrm{y} \rightarrow 1], [\mathrm{x} \rightarrow 7, \mathrm{y} \rightarrow 2], \dots\}) = [\mathrm{x} \rightarrow 7, \mathrm{y} \rightarrow ?\,]$$

And it holds:

$$[\mathrm{x} \rightarrow 7, \mathrm{y} \rightarrow ?\,] \sqsubseteq [\mathrm{x} \rightarrow 7, \mathrm{y} \rightarrow ?\,]$$

And the second condition is fulfilled.

Upper closures

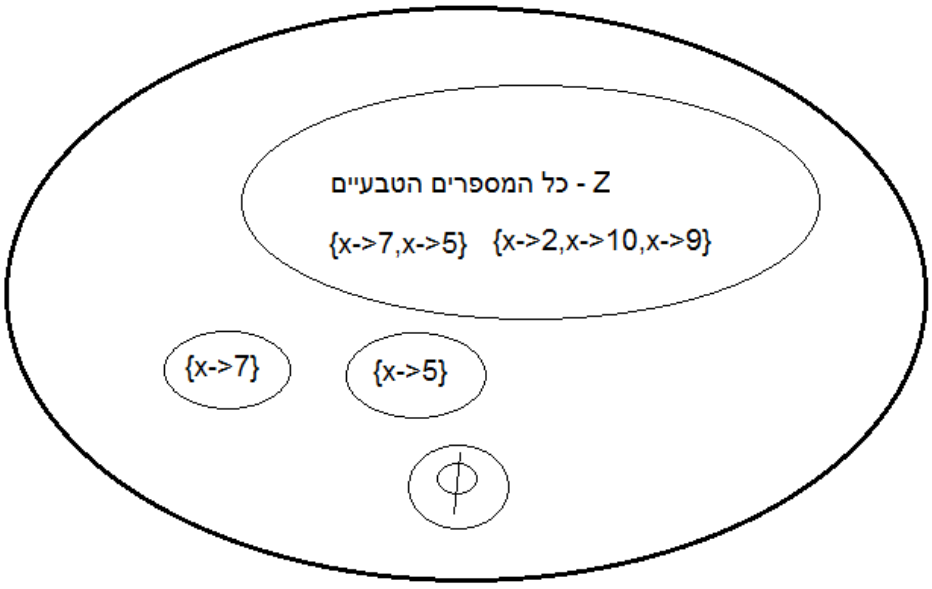
Upper closure is a function$\uparrow$: P(Σ) $\rightarrow$P(Σ) such that

- $\uparrow$ is monotone, i.e., $X \subseteq Y \rightarrow \uparrow X \subseteq \uparrow Y$
- $\uparrow$ is extensive, i.e., $\uparrow X \supseteq X$
- $\uparrow$ is closure, i.e., $\uparrow(\uparrow X) = \uparrow X$

*Every Galois connection defines an upper closure, which is the composing of γ with α (this function is monotonic, because γ and α are monotonic, and we can see that the second and third condition fulfilled immediately from Galois Connection definitions).

*The upper closure of the Galois connection we saw before is the composing of the concretization function with the abstraction function.

In the following example, each circle represents a group of states that have the same closures.



For instance, the closure of {x->7} is $\gamma\big(\alpha(\{x \rightarrow 7\})\big) = \gamma(\{x \rightarrow 7\}) = \{x \rightarrow 7\}$.

We will see that the closure property is holds for this set: $\{x \rightarrow 7, x \rightarrow 5\}$:

$$\gamma\big(\alpha(\{x \rightarrow 7, x \rightarrow 5\})\big) = \gamma(\{x \rightarrow ?\}) = \{\dots, x \rightarrow -1, x \rightarrow 0, x \rightarrow +1, \dots\}$$

$$\gamma\left(\alpha\left(\gamma\left(\alpha(\{x \to 7, x \to 5\})\right)\right)\right) = \gamma\left(\alpha(\{\dots, x \to -1, x \to 0, x \to +1, \dots\})\right)$$
$$= \gamma(\{x \to ?\}) = \{\dots, x \to -1, x \to 0, x \to +1, \dots\}$$

So it holds that $\uparrow\left(\uparrow(\{x \to 7, x \to 5\})\right) = \uparrow(\{x \to 7, x \to 5\})$

How to proof soundness?

- Define an "appropriate" operational semantics
- Define "collecting" structural operational semantics
- Establish a Galois connection between collecting states and abstract states
- (Local correctness) Show that the abstract interpretation of every atomic statement is sound
  w.r.t. the collecting semantics
- (Global correctness) Conclude that the analysis is sound

Collecting semantics:

"Collect" all the states for all possible inputs to the program- i.e. execute the program on each possible input, and add the states we get in this execution to the group of states.

In this approach, there is no reachable state, which is not in this "collection".

Another definition: (iterative one)

- Generate a system of monotone equations (the equations are monotic, because if we generate the collecting method on a bigger set of inputs, we will get a bigger group of states).
- The least solution is well-defined
- The least solution is the collecting interpretation
- But may not be computable

Equations Generated for Collecting Interpretation:

Equations for elementary statements

- [skip]

$$\text{CSexit(l)} = \text{CSentry(l)}$$

- [b]

$$\text{CSexit(l)} = \{\sigma: \sigma \in \text{CSentry(l)}, [\![b]\!]\sigma = tt\}$$

- [x := a]

$$\text{CSexit(l)} = \{(s[x \to [\![\mathbf{A}]\!]s]) \mid s \in \text{CSentry(l)}\}$$

Equations for control flow constructs:

- CSentry(l) = $\cup$ CSexit(l') ,when l' immediately precedes l in the control flow graph. Meaning, the collecting semantic in before executing the line l, is the union of all the possible collecting semantics we get at the end of the previous line (regarding to the execution).

An equation for the entry:

- $CS_{entry}(1) = \{\sigma \mid \sigma \in Var_* \to Z\}$ (For the first line in the execution).

Example:

```
x=0;

If(x>7)

    y=10;

else y=x+3;

y--;
```

The value of $CS_{entry}(1)$, for this program is:

$$CS_{entry}(1) = \{\dots, \{x \to -1, y \to 12\}, \{x \to -2, y \to 0), \dots\}$$

Specialized Chaotic Iterations System of Equations:

$$S = \begin{cases} CS_{entry}[s] = \{\sigma_0\} \\ CS_{entry}[v] = \cup\{f(e)(CS_{entry}[u]) \mid (u, v) \in E\} \\ \text{where } f(e) = \lambda X.\ \{\leftarrow st(e) \rightarrow \sigma \mid \sigma \in X\} \text{ for atomic statements} \\ \qquad f(e) = \lambda X.\{\sigma \mid \leftarrow b(e) \rightarrow \sigma = tt\} \end{cases}$$

The solution of this system of equations is the reachable states in the program.

Example:

For the program from the previous example, we get:

$$CS_{entry}(1) = \{\dots, \{x \to -1, y \to 12\}, \{x \to -2, y \to 0), \dots\}$$

$$CS_{exit}(1) = \{\dots, \{x \to 0, y \to -1\}, \{x \to 0, y \to 0), \{x \to 0, y \to 1\}, \dots\}$$

$$CS_{entry}(2) = \{\dots, \{x \to 0, y \to -1\}, \{x \to 0, y \to 0), \{x \to 0, y \to 1\}, \dots\}$$

$$CS_{exit}(2) = \{\dots, \{x \to 0, y \to -1\}, \{x \to 0, y \to 0), \{x \to 0, y \to 1\}, \dots\}$$

$$CS_{entry}(3) = \{\{x \to 0, y \to 7\}, \{x \to 0, y \to 8), \{x \to 0, y \to 9\}, \dots\}$$

$$CS_{exit}(3) = \{\{x \to 0, y \to 10\}$$

$$CS_{entry}(4) = \{\dots \{x \to 0, y \to 5\}, \{x \to 0, y \to 6), \{x \to 0, y \to 7\}\}$$

$$CS_{exit}(4) = \{\{x \to 0, y \to 3\}\}$$

$$CS_{entry}(5) = CS_{exit}(3) \cup CS_{exit}(4) = \{\{x \to 0, y \to 10\}, \{x \to 0, y \to 3\}\}$$

$$CS_{exit}(5) = \{\{x \to 0, y \to 9\}, \{x \to 0, y \to 2\}\}$$

Translation of the equations system into a function:

$F_S : L^n \to L^n$

$F_s(X)[v] = \cup\{f(e)[u] \mid (u, v) \in E\}$

It holds: lfp(S) = lfp(FS).

The least solution:

- 2n sets of equations
  CSentry(1), …, CSentry (n), CSexit(1), …, CSexit (n)
- Can be written in vectorial form $\overrightarrow{CS} = F_{cs}(\overrightarrow{CS})$
- The least solution lfp(Fcs) is well-defined
- Every component is minimal
- Since Fcs is monotone such a solution always exists
- CSentry(v) = {s|∃s0| <P, s0 > ⇒* (S', s)),
  init(S')=v}
- Simplify the soundness criteria

Now we will see use of the concretization function, in finding a fix point of a function.

Let $f^{\#}$ be a proximity function of f, which holds:

We compute the least fix point of the function $f^{\#}$. if we gives the concretization function this least fix point as input, we get a point that is bigger than the least fix point of f.

We will show now that this is true: let b be the lfp of $f^{\#}$, so it's also holds $f^{\#}(b)=b$. and we get

$$f(\gamma(b)) \sqsubseteq \gamma\left(f^{\#}(b)\right) \sqsubseteq \gamma(b) => \gamma(b) \in Red(f) => \gamma(b) \sqsupseteq lfp(f)$$

Using this fact, we can get a sound approximation to the lfp of f.

Soundness Theorem

There are 3 versions that are different only in their 4th condition:

Version 1:

1. Let (α, γ) form Galois connection from C to A
2. f: C $\rightarrow$ C be a monotone function
3. $f^{\#}$: A $\rightarrow$ A be a monotone function
4. $\forall a \in A: f(\gamma(a)) \sqsubseteq \gamma(f^{\#}(a))$

$$lfp(f) \sqsubseteq \gamma(lfp(f^{\#}))$$

$$\alpha(lfp(f)) \sqsubseteq lfp(f^{\#})$$

Version 2:

1. Let (α, γ) form Galois connection from C to A
2. f: C $\rightarrow$ C be a monotone function
3. $f^{\#}$: A $\rightarrow$ A be a monotone function
4. $\forall c \in C: \alpha(f(c)) \sqsubseteq f^{\#}(\alpha(c))$

$$\alpha(lfp(f)) \sqsubseteq lfp(f^{\#})$$

$$lfp(f) \sqsubseteq \gamma(lfp(f^{\#}))$$

Version 3:

1. Let (α, γ) form Galois connection from C to A
2. f: C $\rightarrow$ C be a monotone function
3. $f^{\#}$: A $\rightarrow$ A be a monotone function
4. $\forall a \in A: \alpha(f(\gamma(a))) \sqsubseteq f^{\#}(a)$

$$\alpha(lfp(f)) \sqsubseteq lfp(f^{\#})$$

$$lfp(f) \sqsubseteq \gamma\left(lfp(f^{\#})\right)$$

*we mentioned in class that the 3th version provide a way to define the best transformer.

Completeness

$$\alpha(lfp(f)) = lfp(f^{\#})$$

$$lfp(f) = \gamma(lfp(f^{\#}))$$

Completeness is harder to achieve than soundness, because here we require equality compared to the inclusion we had before. This means that completeness requires the accurate values (and not a group that includes them).