A Few Remarks on the Abstraction and Equivalence of Semantics

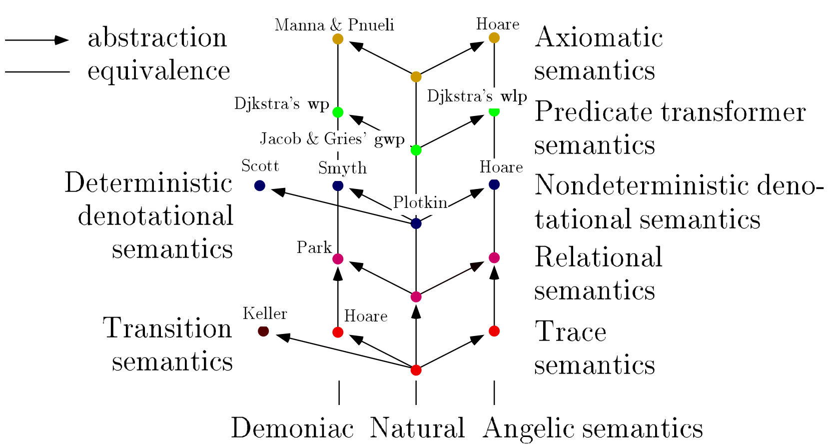
P. Cousot

DMI - École Normale Supérieure
45 rue d'Ulm, 75230 Paris cedex 05, France

cousot@dmi.ens.fr

http://www.dmi.ens.fr/~cousot

The Hierarchy of Semantics



Objective

- Assume that we are given any transition system:

$$\langle S,\, t\rangle$$

state space ↵ ↳ transition relation

- We first define all semantics of this given transition system in the hierarchy of semantics as abstractions of the natural trace semantics;
- We then constructively derive fixpoint characterizations of all semantics in the hierarchy by abstraction of a fixpoint characterization of the natural trace semantics of the transition system.

Description of the hierarchy of semantics as abstractions of the natural trace semantics

Natural trace semantics

- The system/program we are interested in is assumed to be specified by a transition system:

$$\underbrace{\langle S,}_{\text{state space}}\ \underbrace{t\rangle}_{\text{transition relation}}$$

- Its natural trace semantics is:

$$\begin{array}{llll}
\mathcal{T}^\natural = & \{\bullet \longrightarrow \bullet \longrightarrow \bullet \ldots \bullet \longrightarrow \bullet \longrightarrow \bullet\ \text{blocking state}\} & \leftarrow & \text{finite traces} \\
& \quad\downarrow \qquad\qquad\qquad\quad \downarrow & & \\
& \cup\ \{\bullet \longrightarrow \bullet \longrightarrow \bullet \ldots \bullet \longrightarrow \bullet \longrightarrow \bullet \ldots\} & \leftarrow & \text{infinite traces} \\
& \quad\downarrow \qquad\qquad\qquad\quad \downarrow & & \\
& \quad\text{any state} \qquad\quad \text{transition} & &
\end{array}$$

Natural, Demoniac & Angelic Semantics

- Natural trace semantics: $\mathcal{T}^\natural$;
- Angelic abstraction [1]:

$$\alpha(\mathcal{T}^\natural) = \{\bullet \to \bullet \to \ldots \to \bullet \to \bullet \mid \bullet \to \bullet \to \ldots \to \bullet \to \bullet \in \mathcal{T}^\natural\};$$

- Demoniac abstraction [2]:

$$\alpha(\mathcal{T}^\natural) = \mathcal{T}^\natural \cup \{\bullet \to \bullet \to \ldots \to \bullet \to \bullet \mid \bullet \to \bullet \to \ldots \to \bullet \to \bullet \to \ldots \in \mathcal{T}^\natural\}.$$

The α's are Galois connections.

[1] Eliminate all infinite traces.
[2] Introduce all arbitrary finite traces for states possibly starting an infinite trace.

Relational Semantics

$$\alpha \in \text{Traces} \longmapsto \wp(\mathcal{S} \times \mathcal{S}_\bot), \qquad \mathcal{S}_\bot = \mathcal{S} \cup \{\bot\}$$

$$\begin{aligned}
\mathcal{R} &= \alpha(\mathcal{T}) \\
&= \{\langle \overset{a}{\bullet}, \overset{b}{\bullet}\rangle \mid \overset{a}{\bullet} \to \bullet \to \ldots \to \bullet \to \overset{b}{\bullet} \in \mathcal{T}\} \\
&\quad \cup \{\langle \overset{a}{\bullet}, \bot\rangle \mid \overset{a}{\bullet} \to \bullet \to \ldots \to \bullet \to \bullet \to \ldots \in \mathcal{T}\}
\end{aligned}$$

α is a Galois connection.

Non-deterministic Denotational Semantics

$$\alpha \in \wp(\mathcal{S} \times \mathcal{S}_\bot) \longmapsto (\mathcal{S} \longmapsto \wp(\mathcal{S}_\bot))$$

$$\begin{aligned}
\mathcal{D} &= \alpha(\mathcal{R}) \\
&= \lambda s \cdot \{s' \in \mathcal{S}_\bot \mid \langle s, s'\rangle \in \mathcal{R}\} \qquad \text{right image}
\end{aligned}$$

α is a Galois isomorphism.

Predicate Transformer Semantics

$$\alpha \in (\mathcal{S} \longmapsto \wp(\mathcal{S}_\perp)) \longmapsto (\wp(\mathcal{S}_\perp) \longmapsto \wp(\mathcal{S}))$$

$$\begin{aligned}\mathcal{W} &= \alpha(\mathcal{D}) \\ &= \lambda Q \cdot \{s \in \mathcal{S} \mid \forall s' \in \mathcal{S}_\perp : s' \in \mathcal{D}(s) \Rightarrow s' \in Q\}\end{aligned}$$

α is a Galois injection.

Axiomatic Semantics

$$\alpha \in (\wp(\mathcal{S}) \longmapsto \wp(\mathcal{S}_\perp)) \longmapsto \wp(\wp(\mathcal{S}) \times \wp(\mathcal{S}_\perp))$$

$$\begin{aligned}\mathcal{H} &= \alpha(\mathcal{W}) \\ &= \{\langle P, Q\rangle \mid P \subseteq \mathcal{W}(Q)\}\end{aligned}$$

α is a Galois injection.

Fixpoint presentation of the semantics in the hierarchy

Fixpoint presentation of a semantics

- Fixpoint presentations of a semantic:

partial ordering

$\uparrow$

$$\mathcal{T} = \mathrm{lfp}^{\sqsubseteq}\, \mathcal{F}$$

$\uparrow$ $\downarrow$

semantics monotonic transformer

- Problem: find a fixpoint characterization of all semantics in the hierarchy.

- The known fixpoint characterizations look similar[3];
- So there should be a simple way of transferring/lifting fixpoint definitions through abstractions α (as we do in abstract interpretation [CC77]);
- I failed for some time and will explain some of the crucial steps to have this idea work properly.

Reference

[CC77] P. Cousot and R. Cousot. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *4th POPL*, pages 238–252, Los Angeles, Calif., 1977. ACM Press.

[3] although not completely identical.

Difficulty 1: Orderings

- Because "natural" semantics describe both finite and infinite behaviors simultaneously, we cannot use lfp for $\subseteq$. But we could use $\mathrm{gfp}^{\subseteq}$;
- Unfortunately the abstraction of the $\mathrm{gfp}^{\subseteq}$ fixpoint semantics for natural traces does not lead to Scott's denotational semantics;
- So we resort to two orderings[4]:
  1. $\subseteq$ (approximation, refinement, logical implication, ...) for Galois connections α;
  2. $\sqsubseteq$ (computational ordering) for fixpoints.

[4] They are generally different but may happen to coincide by further abstractions.

Natural trace fixpoint semantics

Let X and Y be sets of complete traces:

- $X \subseteq Y$, refinement
- $X \sqsubseteq Y$, computational ordering
  $\stackrel{\Delta}{=} X^+ \subseteq Y^+ \wedge X^\omega \supseteq Y^\omega$
  $X^+ =$ the finite traces of X
  $X^\omega =$ the infinite traces of X
- $$\mathcal{T}^\natural = \mathrm{lfp}^{\sqsubseteq}\, \mathcal{F}$$
  $$\mathcal{F} \stackrel{\Delta}{=} \underset{\substack{\uparrow\\ \text{traces of length 1 ending}\\ \text{in blocking states}}}{\bar{t}} \cup \underset{\substack{\uparrow\\ \text{traces of } X \text{ prefixed}\\ \text{by an initial transition}}}{t \,;\, X}$$

Difficulty 2: THE computational ordering

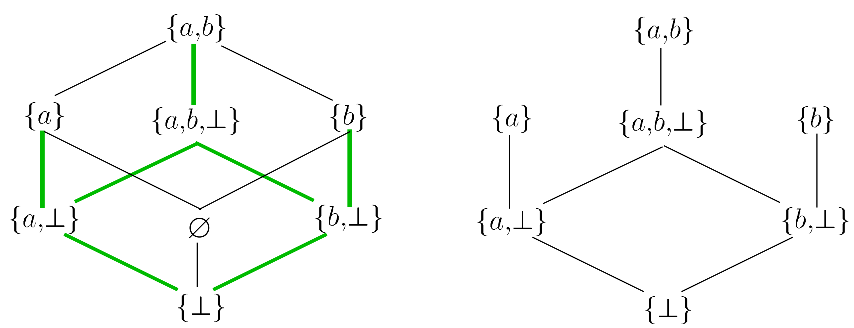
- There is only one approximation ordering;
- There are many possible computational orderings;
- <u>Theorem</u> (very rough sketch) $\mathrm{lfp}^{\sqsubseteq}\, \mathcal{F} = \mathrm{lfp}^{\sqsubseteq'}\, \mathcal{F}$ *iff when ordering the transfinite iterates of $\mathcal{F}$ from $\perp$ by $\sqsubseteq$ and $\sqsubseteq'$, the respective lubs will lead to the same limit.*

More precisely ...

Fixpoint Iterates Reordering

- Let $\langle\langle D,\ \sqsubseteq,\ \bot,\ \sqcup\rangle,\ F\rangle$ be a fixpoint semantic specification;
- let E be a set and $\preceq$ be a binary relation on E, such that:
  1. $\preceq$ is a pre-order on E;
  2. all iterates F^δ, $\delta \in \mathbb{O}$ of F belong to E;
  3. $\bot$ is the $\preceq$-infimum of E;
  4. the restriction $F|_E$ of F to E is $\preceq$-monotone;
  5. for all $x \in E$, if λ is a limit ordinal and $\forall \delta < \lambda : F^\delta \preceq x$ then $\bigsqcup_{\delta<\lambda} F^\delta \preceq x$.
- Then $\mathrm{lfp}_{\bot}^{\sqsubseteq} F = \mathrm{lfp}_{\bot}^{\preceq} F|_E \in E$.

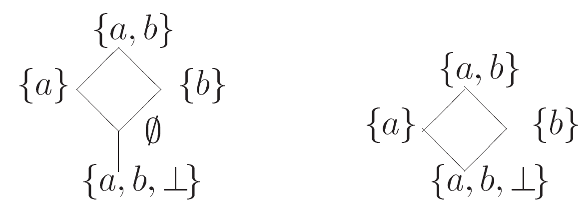
Orderings for the Nondeterministic Denotational Semantics, $S = \{a, b\}$



Computational ordering $\sqsubseteq$ Egli-Milner ordering $\sqsubseteq^{\text{EM}}$

: possible iterates of $\mathcal{F}$

Possible Demoniac Iterate Orderings



Demoniac ordering $\sqsubseteq^\sharp$ Demoniac ordering $\sqsubseteq^\lozenge$

Smyth ordering $\sqsubseteq^{\text{s}}$ Flat ordering $\sqsubseteq^{=}$

: possible iterates of $\mathcal{F}$

Difficulty 3: Fixpoint transfer

- Fixpoint transfer/lifting theorems based upon:
  - Kleene def. of fixpoints
  - Tarski ”

  may not be applicable;
- However, fixpoint transfer/lifting may work by parts.

Kleene Fixpoint Transfer Theorem

If $\langle \mathcal{D},\ F\rangle$ and $\langle \mathcal{D}^\sharp,\ F^\sharp\rangle$ are semantic specifications and

$$\alpha(\bot)\ =\ \bot^\sharp$$

$$F^\sharp \circ \alpha\ =\ \alpha \circ F$$

$\forall$ $\sqsubseteq$-increasing chains $X_\kappa, \kappa \in \Delta : \alpha(\bigsqcup^\sharp_{\kappa\in\Delta} X_\kappa)\ =\ \bigsqcup^\sharp_{\kappa\in\Delta} \alpha(X_\kappa)$
then

$$\alpha(\mathrm{lfp}^{\sqsubseteq} F) = \mathrm{lfp}^{\sqsubseteq^\sharp} F^\sharp$$

Note 1: The condition $F^\sharp \circ \alpha = \alpha \circ F$ provides guidelines for designing $F^\sharp$ when knowing F and α;

Note 2: $F^\sharp$ convergence is faster than that of F.

Tarski Fixpoint Transfer Theorem

If $\langle \mathcal{D},\ \sqsubseteq,\ \bot,\ \sqcup\rangle$ and $\langle \mathcal{D}^\sharp,\ \sqsubseteq^\sharp,\ \bot^\sharp,\ \sqcup^\sharp\rangle$ are complete lattices, $F \in \mathcal{D} \stackrel{\mathrm{m}}{\longmapsto} \mathcal{D}$, $F^\sharp \in \mathcal{D}^\sharp \stackrel{\mathrm{m}}{\longmapsto} \mathcal{D}^\sharp$ are monotonic and

- α is a complete $\sqcap$-morphism
- $F^\sharp \circ \alpha \sqsubseteq^\sharp \alpha \circ F$
- $\forall y \in \mathcal{D}^\sharp : F^\sharp(y) \sqsubseteq^\sharp y \Rightarrow \exists x \in \mathcal{D} : \alpha(x) = y \wedge F(x) \sqsubseteq^\natural x$

then

$$\alpha(\mathrm{lfp}^{\sqsubseteq} F) = \mathrm{lfp}^{\sqsubseteq^\sharp} F^\sharp$$

Example: Traces to relation abstraction

- Problem for $\alpha \in$ Traces $\longmapsto$ Relation:
  - α is continuous for $\subseteq$,
  - α is <u>not</u> continuous for $\sqsubseteq$:
    $\Rightarrow$ Kleene fixpoint transfer not applicable,
    $\Rightarrow$ But applicable to finite traces;
  - α is <u>not</u> a complete $\sqcap$-morphism (because not complete $\cap$-morphism):
    $\Rightarrow$ Tarski fixpoint transfer not applicable,
    $\Rightarrow$ But applicable to infinite traces (since α is a a complete $\cup$-morphism) ,
- Solution: split, transfer by parts, combine.

Difficulty 4: Predicate transformer transformer

- For the predicate transformer semantics, the fixpoint characterization has the form:

$$\underset{\substack{\uparrow\\ \text{predicate transformer}}}{\mathcal{W}} = \mathrm{lfp}^{\dot{\sqsubseteq}}\ \underset{\substack{\uparrow\\ \text{predicate transformer}\\ \text{transformer}}}{\mathcal{F}}$$

- Use the further abstraction:

$$\begin{array}{rcl}\alpha_Q & \in & (\wp(S_\bot) \longmapsto \wp(S)) \longmapsto \wp(S) \\ \alpha_Q(\mathcal{W}) & = & \mathcal{W}(Q)\end{array}$$

which consists in fixing the postcondition $Q \subseteq S_\bot$ to get Dijkstra's fixpoint:

$$\mathcal{W} = \lambda Q \cdot \mathrm{lfp}^{\sqsubseteq}\, \mathcal{F}(Q)$$

$\uparrow$ predicate transformer

$\uparrow$ predicate transformer

Difficulty 5: Healthiness conditions

- Healthiness conditions follow from the requirement that the abstraction α should be surjective;
- More precisely by characterizing the image of the semantic domain by the abstraction function α;
- Galois injections now become Galois isomorphisms.

Example 1: Predicate transformers

Denotational | Predicate transformer

$$(S \longmapsto \wp(S_\bot)) \xrightarrow{\alpha} (\wp(S_\bot) \longmapsto \wp(S))$$

$+\ \alpha$ surjective

$\Rightarrow$

$$(S \longmapsto \wp(S_\bot)) \xrightarrow{\alpha} (\wp(S_\bot) \overset{\cap}{\longmapsto} \wp(S))$$

Example 2: Hoare logics

Predicate transformer | Hoare logic

$$(\wp(S_\bot) \overset{\cup}{\longmapsto} \wp(S)) \xrightarrow{\alpha} \wp((\wp(S) \times \wp(S_\bot)))$$

$+\ \alpha$ surjective

$\Rightarrow$

$$(\wp(S_\bot) \overset{\cup}{\longmapsto} \wp(S)) \xrightarrow{\alpha} \wp(S) \otimes \wp(S_\bot)$$

Exercice: what is $\otimes$?

- Tensor product:

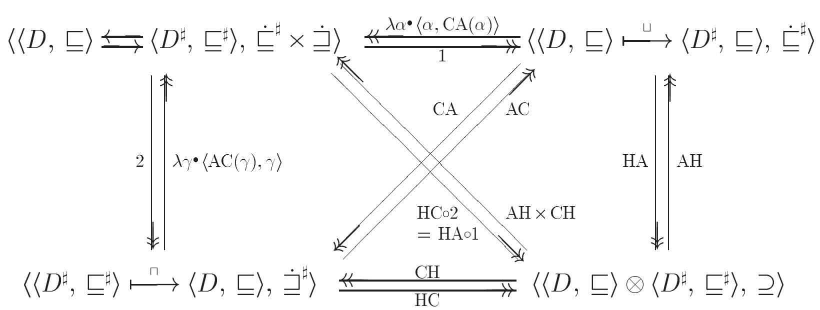
$$\langle D,\ \sqsubseteq\rangle \otimes \langle D^\sharp,\ \sqsubseteq^\sharp\rangle \ \stackrel{\Delta}{=}\ \{H \in \wp(D \times D^\sharp) \mid (1) \wedge (2) \wedge (3)\}$$

where the conditions are:

1. $(X \sqsubseteq X' \wedge \langle X',\ Y'\rangle \in H \wedge Y' \sqsubseteq^\sharp Y) \Rightarrow (\langle X,\ Y\rangle \in H)$;
   (consequence rule of Hoare logic)
2. $(\forall i \in \Delta : \langle X_i,\ Y\rangle \in H) \Rightarrow (\langle \bigsqcup_{i\in\Delta} X_i,\ Y\rangle \in H)$;
3. $(\forall i \in \Delta : \langle X,\ Y_i\rangle \in H) \Rightarrow (\langle X,\ \bigsqcap_{i\in\Delta} Y_i\rangle \in H)$
   (by induction on the program structure, 2 and 3 follow from Hoare logic rules).

Galois Connection Commutative Diagram

$$
\begin{array}{llll}
1(\langle\alpha,\ \gamma\rangle) \stackrel{\Delta}{=} \alpha & & \mathrm{HA}(\alpha) \stackrel{\Delta}{=} \{\langle x,\ y\rangle \in D \times D^\sharp \mid \alpha(x) \sqsubseteq^\sharp y\} \\
2(\langle\alpha,\ \gamma\rangle) \stackrel{\Delta}{=} \gamma & & \mathrm{HC}(\gamma) \stackrel{\Delta}{=} \{\langle x,\ y\rangle \in D \times D^\sharp \mid x \sqsubseteq \gamma(y)\} \\
\mathrm{AC}(\gamma) \stackrel{\Delta}{=} \lambda x\bullet \sqcap^\sharp\{y \mid x \sqsubseteq \gamma(y)\} & & \mathrm{AH}(H) \stackrel{\Delta}{=} \lambda x\bullet \sqcap^\sharp\{y \mid \langle x,\ y\rangle \in H\} \\
\mathrm{CA}(\alpha) \stackrel{\Delta}{=} \lambda y\bullet \sqcup\{x \mid \alpha(x) \sqsubseteq^\sharp y\} & & \mathrm{CH}(H) \stackrel{\Delta}{=} \lambda y\bullet \sqcup\{x \mid \langle x,\ y\rangle \in H\}
\end{array}
$$



Difficulty 6: From fixpoint to proof rule semantics

1) For safety/invariance, use Park induction (F monotonic on complete lattice) :

$$
\begin{array}{ll}
& \mathrm{lfp}_\perp^\sqsubseteq F \sqsubseteq P \\
\Longleftrightarrow & \exists I : F(I) \sqsubseteq I \wedge I \sqsubseteq P
\end{array}
$$

2) For inevitability/liveness, use Scott induction ? No (F monotonic on cpo):

$$
\begin{array}{ll}
& P \sqsubseteq \mathrm{lfp}_\perp^\sqsubseteq F \\
\Longleftrightarrow & \exists \epsilon \in \mathbb{O} : \\
& \quad \exists I \in (\epsilon + 1) \longmapsto \wp(\Sigma) : \\
& \quad\quad I^0 \sqsubseteq \perp \\
& \quad \wedge\ \forall \delta : 0 < \delta \leq \epsilon : I^\delta \sqsubseteq F(\bigsqcup_{\beta<\delta} I^\beta) \\
& \quad \wedge\ P \subseteq I^\epsilon
\end{array}
$$

Conclusion

- Synthetic and uniformizing (although somewhat contemplative) work;
- Shows that abstract interpretation formalizes semantics abstraction nicely;
- Help to compare abstract interpretation based program analysis methods;
- Help to understand their limitations (e.g. denotational semantics + $\subseteq = \sqsubseteq \Rightarrow$ failure for binding time analysis + strictness analysis);

Research work

- Extend the hierarchy to other semantics of transition systems;
- Extend to a programming calculus with interpretations at all levels in the hierarchy;
- Extend at higher-order to the λ-calculus[5].

[5] This should work, but is it really worth a long effort?

Reference

For technical details and references, see:

– P. Cousot. Constructive design of a hierarchy of semantics of a transition system by abstract interpretation. *Electronic Notes in Theoretical Computer Science*, 6, 1997, 25 pages.
URL: `http://www.elsevier.nl/locate/entcs/volume6.html`.