ABSTRACT INTERPRETATION: THEORY AND APPLICATIONS

P. Cousot

Patrick.Cousot@ens.fr http://www.di.ens.fr/~cousot

Second International Summer School in Computational Logic, ISCL 2002

25th—30th August 2002, Acquafredda di Maratea (Basilicata, Italy)

LESSON 2 (2/2)

Elements of Abstract Interpretation

- P. Cousot. *Méthodes itératives de construction et d'approximation de points fixes d'opérateurs monotones sur un treillis, analyse sémantique de programmes*. Thèse d'État ès sciences mathématiques. Grenoble, 21 Mar. 1978.

Galois Connections[12]

$$\langle P, \leq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle Q, \sqsubseteq \rangle$$

$\stackrel{\text{def}}{=}$

- $\langle P, \leq \rangle$ is a poset
- $\langle Q, \sqsubseteq \rangle$ is a poset
- $\forall x \in P : \forall y \in Q : \alpha(x) \sqsubseteq y \iff x \leq \gamma(y)$

12 The original Galois correspondence is semi-dual ($\sqsupseteq$ instead of $\sqsubseteq$).

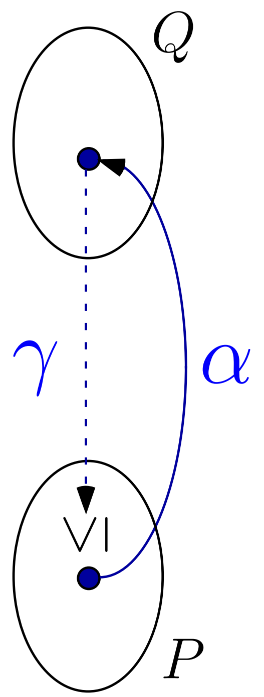
Composing Galois Connections

- If $\langle P, \leq\rangle \xleftarrow[\alpha_1]{\gamma_1}\!\!\xrightarrow{} \langle Q, \sqsubseteq\rangle$ and $\langle Q, \sqsubseteq\rangle \xleftarrow[\alpha_2]{\gamma_2}\!\!\xrightarrow{} \langle R, \preceq\rangle$ then

$$\langle P, \leq\rangle \xleftarrow[\alpha_2 \circ \alpha_1]{\gamma_1 \circ \gamma_2}\!\!\xrightarrow{} \langle R, \preceq\rangle$$ [13]

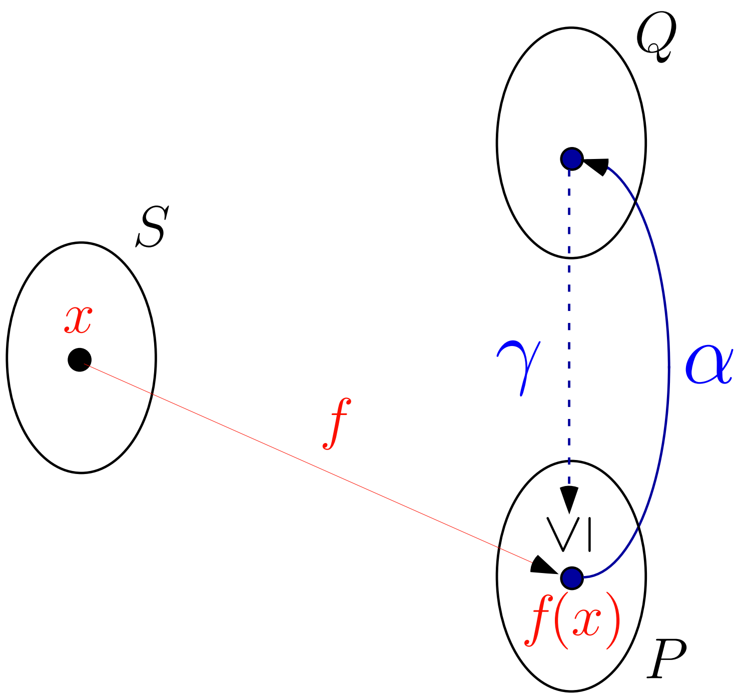
13 This would not be true with the original definition of Galois correspondences.

Function Abstraction (1)



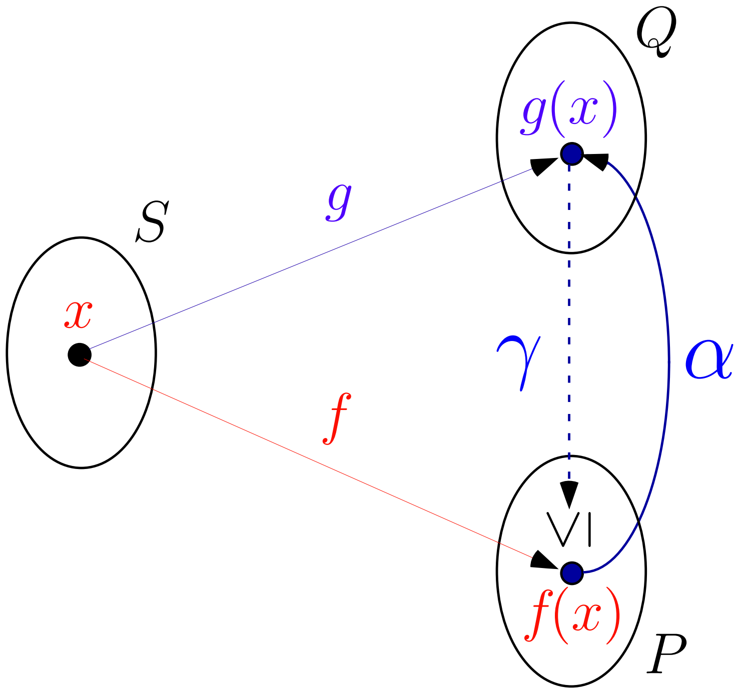
$$\langle P, \leq \rangle \xleftarrow[\alpha]{\gamma} \langle Q, \sqsubseteq \rangle$$

Function Abstraction (1)



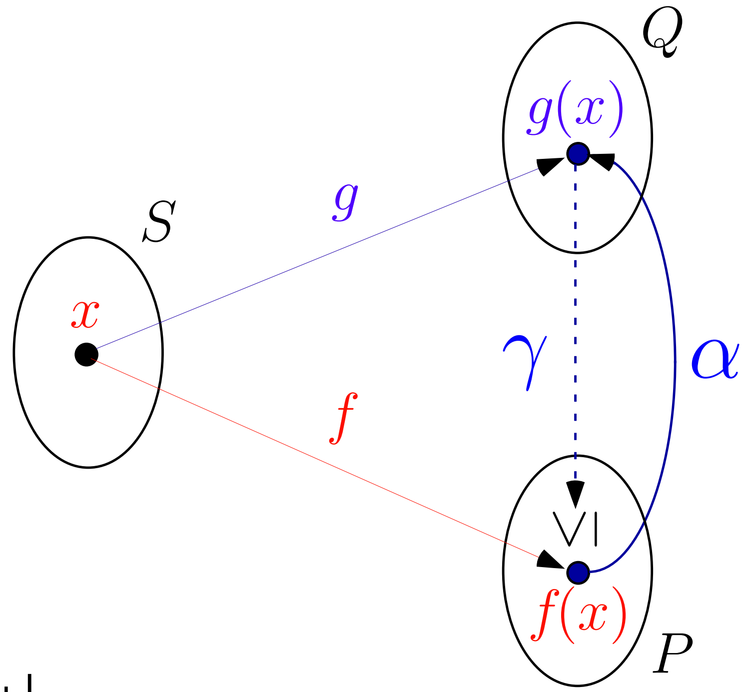
$$\langle P, \leq \rangle \xrightleftharpoons[\alpha]{\gamma} \langle Q, \sqsubseteq \rangle$$

Function Abstraction (1)



$$\langle P, \leq \rangle \xleftarrow[\alpha]{\gamma} \langle Q, \sqsubseteq \rangle$$

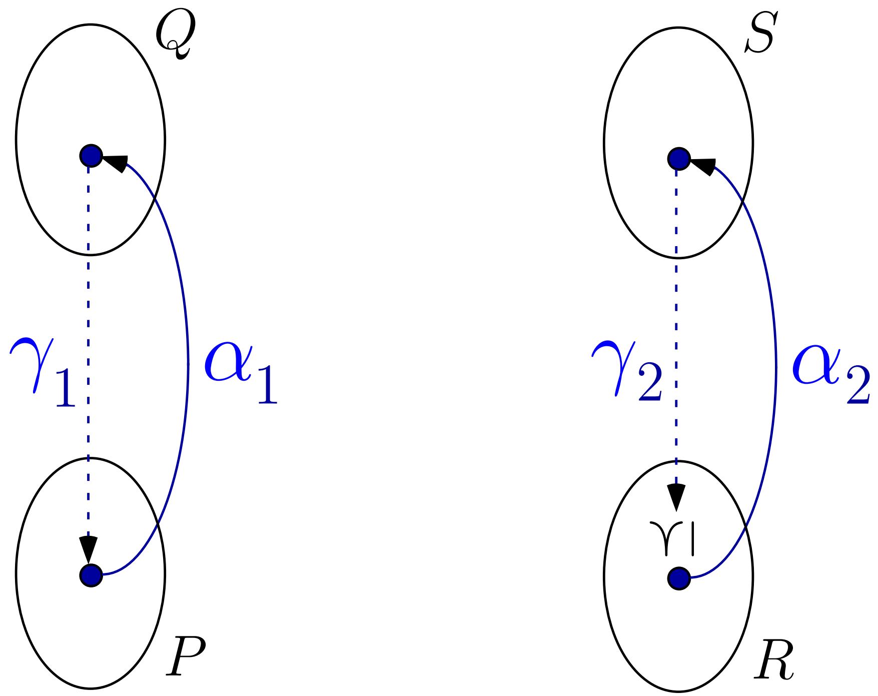
Function Abstraction (1)



- If $\langle P, \leq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle Q, \sqsubseteq \rangle$ then

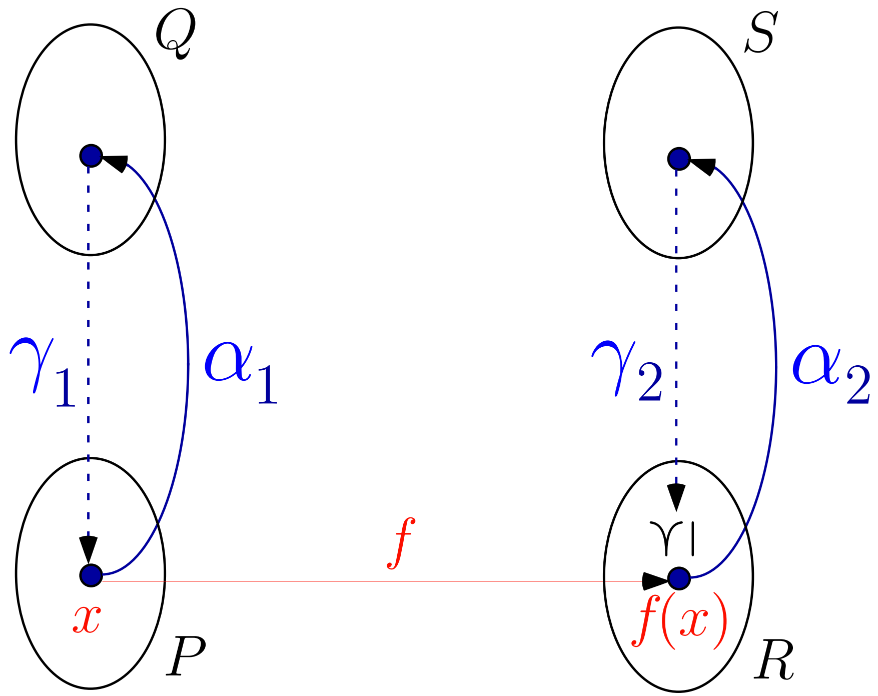
$$\langle S \mapsto P, \dot{\leq} \rangle \underset{\boldsymbol{\lambda} f \cdot \boldsymbol{\lambda} x \cdot \alpha(f(x))}{\overset{\boldsymbol{\lambda} g \cdot \boldsymbol{\lambda} x \cdot \gamma(g(x))}{\leftrightarrows}} \langle S \mapsto Q, \dot{\sqsubseteq} \rangle$$

Function Abstraction (2)



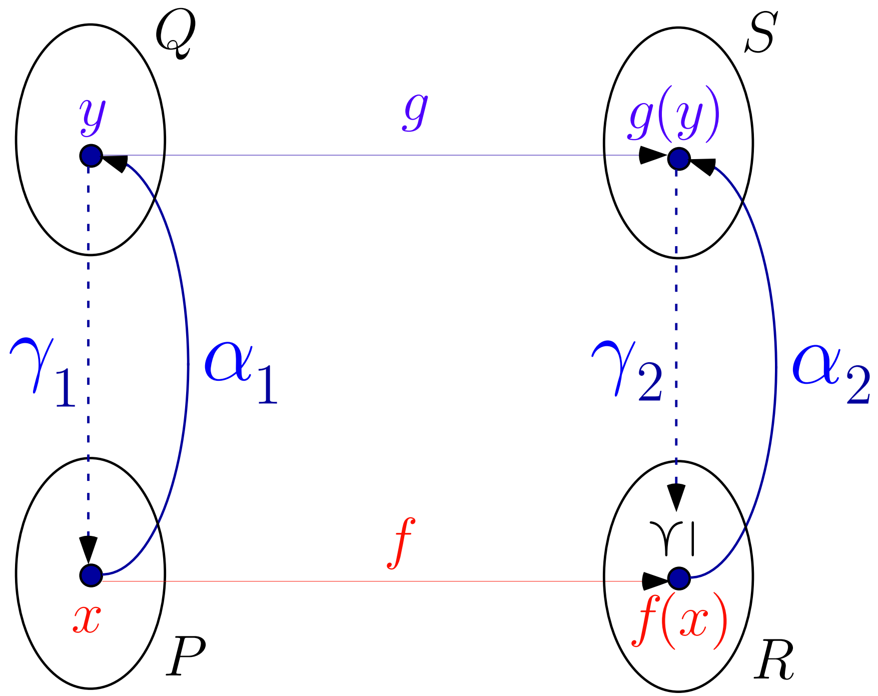
$$\langle P, \leq \rangle \underset{\alpha_1}{\overset{\gamma_1}{\leftrightarrows}} \langle Q, \subseteq \rangle \qquad \langle R, \preceq \rangle \underset{\alpha_2}{\overset{\gamma_2}{\leftrightarrows}} \langle S, \sqsubseteq \rangle$$

Function Abstraction (2)



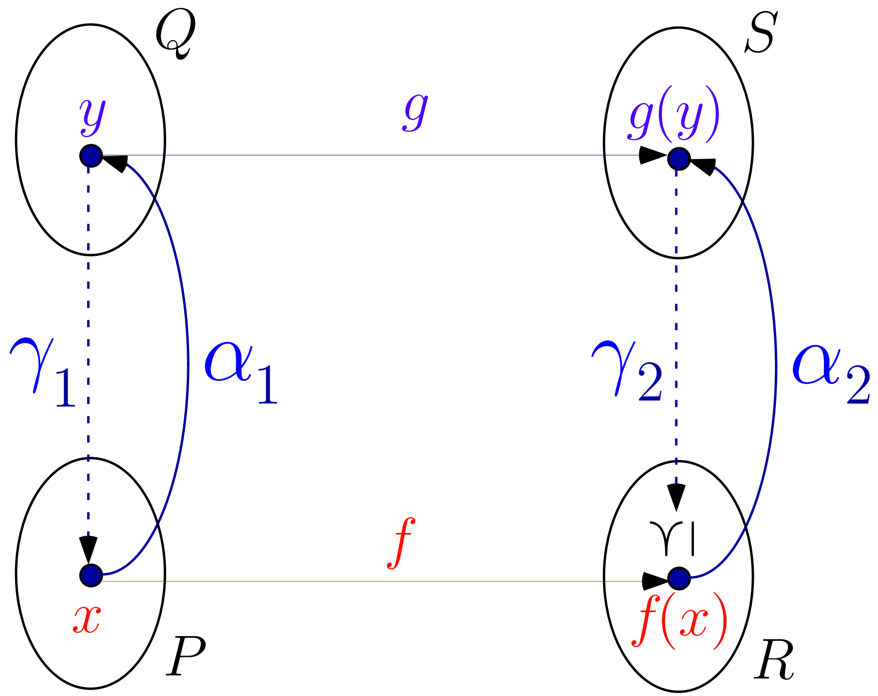
$$\langle P, \leq\rangle \underset{\alpha_1}{\overset{\gamma_1}{\leftrightarrows}} \langle Q, \subseteq\rangle \qquad \langle R, \preceq\rangle \underset{\alpha_2}{\overset{\gamma_2}{\leftrightarrows}} \langle S, \sqsubseteq\rangle$$

Function Abstraction (2)



$$\langle P, \leq \rangle \underset{\alpha_1}{\overset{\gamma_1}{\leftrightarrows}} \langle Q, \subseteq \rangle \qquad \langle R, \preceq \rangle \underset{\alpha_2}{\overset{\gamma_2}{\leftrightarrows}} \langle S, \sqsubseteq \rangle$$

Function Abstraction (2)



- If $\langle P, \leq\rangle \underset{\alpha_1}{\overset{\gamma_1}{\leftrightarrows}} \langle Q, \subseteq\rangle$ and $\langle R, \preceq\rangle \underset{\alpha_2}{\overset{\gamma_2}{\leftrightarrows}} \langle S, \sqsubseteq\rangle$ then

$$\langle P \overset{m}{\longmapsto} R, \dot{\subseteq}\rangle \underset{\boldsymbol{\lambda} f \cdot \alpha_2 \circ f \circ \gamma_1}{\overset{\boldsymbol{\lambda} g \cdot \gamma_2 \circ g \circ \alpha_1}{\leftrightarrows}} \langle Q \overset{m}{\longmapsto} S, \dot{\sqsubseteq}\rangle$$

Fixpoint Approximation

Let $F \in L \stackrel{m}{\longmapsto} L$ and $\overline{F} \in \overline{L} \stackrel{m}{\longmapsto} \overline{L}$ be respective monotone maps on the cpos $\langle L, \bot, \sqsubseteq \rangle$ and $\langle \overline{L}, \overline{\bot}, \overline{\sqsubseteq} \rangle$ and $\langle L, \sqsubseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \overline{L}, \overline{\sqsubseteq} \rangle$ such that $\alpha \circ F \circ \gamma \; \dot{\overline{\sqsubseteq}} \; \overline{F}$. Then [14]:

- $\forall \delta \in \mathbb{O}\colon\ \alpha(F^{\delta}) \mathrel{\overline{\sqsubseteq}} \overline{F}^{\delta}$ (iterates from the infimum);
- The iteration order of $\overline{F}$ is $\leq$ to that of F;
- $\alpha(\mathrm{lfp}^{\sqsubseteq} F) \mathrel{\overline{\sqsubseteq}} \mathrm{lfp}^{\overline{\sqsubseteq}} \overline{F}$;

14 P. Cousot & R. Cousot. *Systematic design of program analysis frameworks*. ACM POPL'79, pp. 269–282, 1979. Numerous variants!

Fixpoint Approximation

Let $F \in L \stackrel{m}{\longmapsto} L$ and $\overline{F} \in \overline{L} \stackrel{m}{\longmapsto} \overline{L}$ be respective monotone maps on the cpos $\langle L, \bot, \sqsubseteq \rangle$ and $\langle \overline{L}, \overline{\bot}, \overline{\sqsubseteq} \rangle$ and $\langle L, \sqsubseteq \rangle \stackrel{\gamma}{\underset{\alpha}{\leftrightarrows}} \langle \overline{L}, \overline{\sqsubseteq} \rangle$ such that $\alpha \circ F \circ \gamma \; \dot{\overline{\sqsubseteq}} \; \overline{F}$. Then [14]:

- $\forall \delta \in \mathbb{O}$: $\alpha(F^\delta) \; \overline{\sqsubseteq} \; \overline{F}^\delta$ (iterates from the infimum);
- The iteration order of $\overline{F}$ is $\leq$ to that of F;
- $\alpha(\mathrm{lfp}^{\sqsubseteq} F) \; \overline{\sqsubseteq} \; \mathrm{lfp}^{\overline{\sqsubseteq}} \overline{F}$;

Soundness: $\mathrm{lfp}^{\overline{\sqsubseteq}} \overline{F} \; \overline{\sqsubseteq} \; \overline{P} \Rightarrow \mathrm{lfp}^{\sqsubseteq} F \sqsubseteq \gamma(\overline{P})$.

---

14 P. Cousot & R. Cousot. *Systematic design of program analysis frameworks.* ACM POPL'79, pp. 269–282, 1979. Numerous variants!

Fixpoint Abstraction

Moreover, the *commutation condition* $\overline{F} \circ \alpha = \alpha \circ F$ implies[15]:

- $\overline{F} = \alpha \circ F \circ \gamma$, and
- $\alpha(\mathrm{lfp}^{\sqsubseteq} F) = \mathrm{lfp}^{\overline{\sqsubseteq}} \overline{F}$;

[15] P. Cousot & R. Cousot. *Systematic design of program analysis frameworks*. ACM POPL'79, pp. 269–282, 1979. Numerous variants!

Fixpoint Abstraction

Moreover, the *commutation condition* $\overline{F} \circ \alpha = \alpha \circ F$ implies[15]:

- $\overline{F} = \alpha \circ F \circ \gamma$, and
- $\alpha(\mathrm{lfp}^{\sqsubseteq} F) = \mathrm{lfp}^{\overline{\sqsubseteq}} \overline{F}$;

Completeness: $\mathrm{lfp}^{\sqsubseteq} F \sqsubseteq \gamma(\overline{P}) \Rightarrow \mathrm{lfp}^{\overline{\sqsubseteq}} \overline{F} \mathrel{\overline{\sqsubseteq}} \overline{P}$.

15 P. Cousot & R. Cousot. *Systematic design of program analysis frameworks*. ACM POPL'79, pp. 269–282, 1979. Numerous variants!

Systematic Design of an Abstract Semantics

By structural induction on the language syntax, for each language construct:

- Define the concrete semantics $\operatorname{lfp}^{\sqsubseteq} F$;
- Choose the abstraction $\alpha = \kappa(\alpha_1, \ldots, \alpha_n)$ and check $\langle L, \sqsubseteq\rangle \xleftrightarrow[\alpha]{\gamma} \langle \overline{L}, \overline{\sqsubseteq}\rangle$;
- Calculate $\overline{F} \stackrel{\text{def}}{=} \alpha \circ F \circ \gamma$ and check that $\overline{F} \circ \alpha = \alpha \circ F$;
- It follows, by construction, that $\alpha(\operatorname{lfp}^{\sqsubseteq} F) = \operatorname{lfp}^{\overline{\sqsubseteq}} \overline{F}$.

(and similarly in case of approximation).

Abstract Domains

An abstraction α is a specification of an abstract domain, including:

- the representation of the abstract properties;
- the approximation ordering lattice structure $(\leq, 0, 1, \vee, \wedge, \ldots)$;
- the computational ordering cpo structure $(\sqsubseteq, \bot, \sqcup, \ldots)$;
- the abstract operators, e.g. *non-relational abstract multiplication*:
  - $P \otimes Q \stackrel{\text{def}}{=} \alpha(\{x \times y \mid x \in \gamma(P) \wedge y \in \gamma(Q)\})$ *postcondition*
  - $\otimes^{-1}(R) \stackrel{\text{def}}{=} \alpha(\{\langle x, y\rangle \mid x \times y \in \gamma(R)\})$ *precondition*

Combinations of Abstract Domains[16]

Operation	$\kappa(\alpha_1, \ldots, \alpha_n)$	*Intuition*
Composition	$\alpha_n \circ \ldots \circ \alpha_1$	*Successive abstractions*
Duality	$\neg\kappa(\neg\alpha_1, \ldots, \neg\alpha_n)$	*Contraposition*[17]
Reduced product	$\alpha_1 \sqcap \ldots \sqcap \alpha_n$	*Conjunction*
Reduced power	$\alpha_1 \mapsto \ldots \mapsto \alpha_n$	*Case analysis*

16 P. Cousot & R. Cousot. *Systematic design of program analysis frameworks*. ACM POPL'79, pp. 269–282, 1979.

17 P. Cousot. *Semantic Foundations of Program Analysis*. In *Program Flow Analysis: Theory and Applications*, Prentice-Hall, pp. 303–342, 1981.

A Potpourri of Applications of Abstract Interpretation

Content of the Potpourri of Applications of Abstract Interpretation

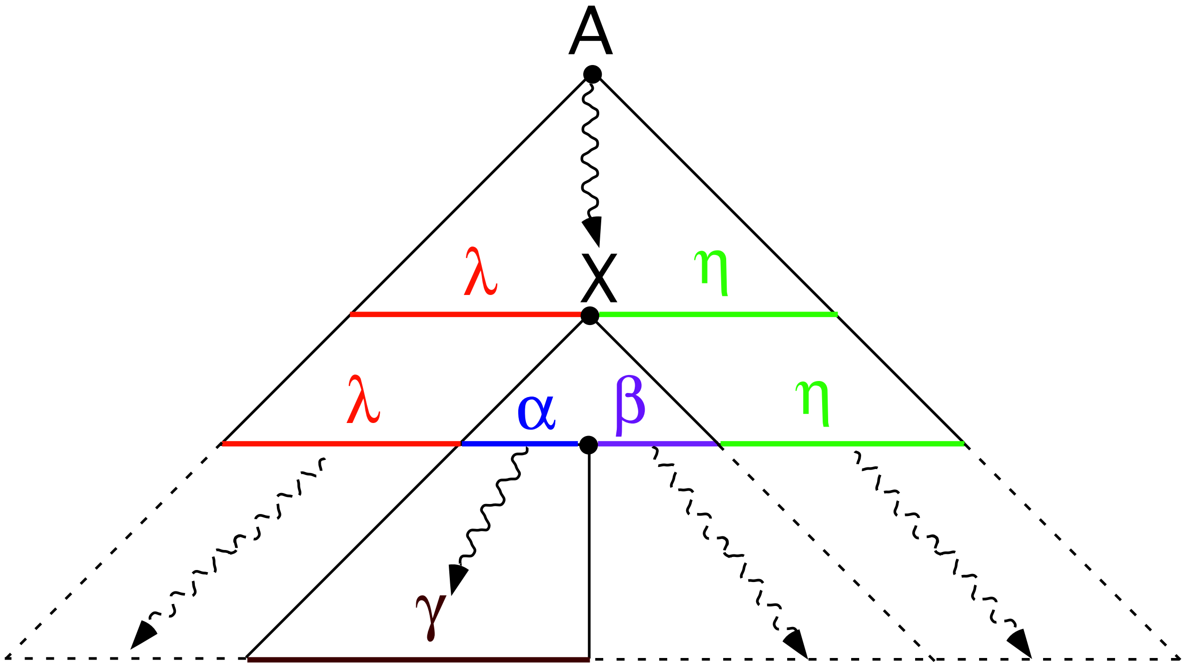
1. Syntax 47
2. Semantics 51
3. Typing 62
4. Model Checking 76
5. Program Transformations 89
6. Static Program Analysis 93

Application to Syntax

• P. Cousot & R. Cousot. *Parsing as Abstract Interpretation of Grammar Semantics*, TCS, 2002, in press.

The Semantics of Syntax

- The semantics of a grammar $G = \langle N, T, P, A \rangle$ is the set of items $[\lambda, X := \alpha/\gamma \bullet \beta]$ such that $\exists \eta : \exists X := \alpha\beta \in P$:



The Fixpoint Semantics of Syntax

$$S = \mathrm{lfp}^{\subseteq} F$$

$$\begin{aligned}
F(I) \stackrel{\text{def}}{=} & \{[\epsilon, A := \epsilon/\epsilon \bullet \beta] \mid A := \beta \in P\} \\
& \cup \{[\lambda, X := \alpha Y/\gamma\delta \bullet \beta] \mid [\lambda, X := \alpha/\gamma \bullet Y\beta] \in I \wedge \\
& \qquad Y := \delta \in P\} \\
& \cup \{[\lambda, X := \alpha Y/\gamma\xi \bullet \beta] \mid [\lambda, X := \alpha/\gamma \bullet Y\beta] \in I \wedge \\
& \qquad [\lambda\gamma, Y := \delta/\xi \bullet \epsilon] \in I\} \\
& \cup \{[\lambda, X := \alpha a/\gamma a \bullet \beta] \mid [\lambda, X := \alpha/\gamma \bullet a\beta] \in I\} .
\end{aligned}$$

Syntactic Abstractions

- $\alpha_\ell(I) \stackrel{\text{def}}{=} \{\gamma \in T^\star \mid [\epsilon, A := \alpha/\gamma \bullet \epsilon] \in I\}$

  Language of the grammar $G = \langle N, T, P, A \rangle$

Syntactic Abstractions

- $\alpha_\ell(I) \stackrel{\text{def}}{=} \{\gamma \in T^\star \mid [\epsilon, A := \alpha/\gamma \bullet \epsilon] \in I\}$

  Language of the grammar $G = \langle N, T, P, A\rangle$

- $\omega = \omega_1 \ldots \omega_i \omega_{i+1} \ldots \omega_j \ldots \omega_n$ input string

  $$\alpha_\omega(I) \stackrel{\text{def}}{=} \{\langle X := \alpha \bullet \beta, i, j\rangle \mid 0 \leq i \leq j \leq n \wedge [\omega_1 \ldots \omega_i, X := \alpha/\omega_{i+1} \ldots \omega_j \bullet \beta] \in I\}$$

  Earley's algorithm

Syntactic Abstractions

- $\alpha_\ell(I) \stackrel{\text{def}}{=} \{\gamma \in T^\star \mid [\epsilon, A := \alpha/\gamma \bullet \epsilon] \in I\}$

  Language of the grammar $G = \langle N, T, P, A\rangle$

- $\omega = \omega_1 \ldots \omega_i \omega_{i+1} \ldots \omega_j \ldots \omega_n$ input string

  $$\alpha_\omega(I) \stackrel{\text{def}}{=} \{\langle X := \alpha \bullet \beta, i, j\rangle \mid 0 \leq i \leq j \leq n \land [\omega_1 \ldots \omega_i, X := \alpha/\omega_{i+1} \ldots \omega_j \bullet \beta] \in I\}$$

  Earley's algorithm

- $$\alpha_f(I) \stackrel{\text{def}}{=} \{a \in T \mid [\lambda, X := \alpha/a\gamma \bullet \beta] \in I\} \cup \{\epsilon \mid [\lambda, X := \alpha\beta/\epsilon \bullet \epsilon] \in I\}$$

  FIRST algorithm