ABSTRACT INTERPRETATION FOR DATA-FLOW ANALYSIS

- most interesting properties of logic programs are properties of computed answers
  - groundness
    - essential component of several other analyses
      - independence
      - sharing
      - occur-check, ...
    - if the result of the analysis tells us that
      X is ground
      the construction guarantees that every possible execution will bind X to a ground term
  - freeness
  - linearity
  - sharing

    $$\{ \{ \}, \ldots, \{X, Y, Z\}, \ldots \}$$

    X, Y, Z may be bound to terms containing the same variable.
  - types
    - un esempio parlando di verifica
- we will look at "groundness" abstract domain

ABSTRACTING COMPUTED ANSWERS

- rather than defining a semi-perfect or semi-denotational observable modeling a suitable groundness domain (from SLD-derivations) we take as concrete semantics a semantics modeling computed answers
    - a denotational observable
    - we will reason about the denotational semantics only
- just to make the presentation simpler

- the denotational ~~[illegible]~~ definition of an "equational version" of the S-semantics
- we will abstract the concrete domain only

$$\mathcal{P}(Eqns)$$

- it is equivalent to defining a (semi-denotational) observable which is the composition of ~~[illegible]~~

$$\begin{tikzcd}
\text{SLD-derivations} \arrow[d] \arrow[dd, bend right] \\
\text{computed answers} \arrow[d] \\
\text{abstract (groundness) computed answers}
\end{tikzcd}$$

EQUATIONAL s-SEMANTICS

$$\mathcal{C}[\![p(\tilde{t}) :- e, q_1(\tilde{t}_1), \dots, q_n(\tilde{t}_n)]\!]_I =$$

$$= \Big\{ \langle p(\tilde{x}), E \rangle \;\Big|\; \begin{array}{l} \exists \langle q_i(\tilde{x}_i), E_i \rangle \in I, \\ e_i \in E_i, \\ e' = (\tilde{x} = \tilde{t} \wedge e \wedge \tilde{x}_1 = \tilde{t}_1 \wedge \dots \wedge \tilde{x}_n = \tilde{t}_n \wedge e_1 \wedge \dots \wedge e_n)_{\setminus \tilde{x}}, \\ e' \text{ is satisfiable}, \\ e' \in E \end{array} \Big\}$$

$$\mathcal{P}[\![\text{emptyproc}]\!]_I = \{ \langle p(\tilde{x}), \phi \rangle \mid p \in P \}$$

$$\mathcal{P}[\![c;\ P]\!]_I = \Big\{ \langle p(\tilde{x}), E \rangle \;\Big|\; \begin{array}{l} \langle p(\tilde{x}), E_1 \rangle \in \mathcal{C}[\![c]\!]_I, \\ \langle p(\tilde{x}), E_2 \rangle \in \mathcal{P}[\![P]\!]_I, \\ E = E_1 \cup E_2 \end{array} \Big\}$$

- concrete domain

$$\wp(Eqns)$$

- $e \in Eqns$, $E \in \wp(Eqns)$
- the partial order is essentially $\subseteq$

FROM CONCRETE TO ABSTRACT SEMANTICS

- assume α is an abstraction function from $\wp(Eqns)$ to a suitable abstract domain

- the abstract (version of the) "semantic" interpretation

$$\alpha_S(I) = \{ \langle p(\tilde{x}), F \rangle \mid \langle p(\tilde{x}), E \rangle \in I, F = \alpha(E) \}$$

- the abstract semantic equations are obtained from the concrete one by replacing $\wedge$ and $\cup$ by glb and lub respectively

$$\mathcal{C}^\alpha [\![p(\tilde{t}) \text{:-} e, q_1(\tilde{t}_1), \dots, q_n(\tilde{t}_n)]\!]_{I^\alpha} =$$

$$\{ \langle p(\tilde{x}), F \rangle \mid \exists \langle q_i(\tilde{x}_i), F_i \rangle \in I^\alpha,$$
$$F = glb(\{ \alpha(\tilde{x} = \tilde{t}), \alpha(e), \alpha(\tilde{x}_1 = \tilde{t}_1), \dots, \alpha(\tilde{x}_n = \tilde{t}_n), F_1, \dots, F_n \})_{|\tilde{x}} \}$$

$$\mathcal{P}^\alpha [\![\text{empty prog}]\!]_{I^\alpha} = \{ \langle p(\tilde{x}), \perp \rangle \mid p \in P \}$$

$$\mathcal{P}^\alpha [\![c; P]\!]_{I^\alpha} = \{ \langle p(\tilde{x}), F \rangle \mid \langle p(\tilde{x}), F_1 \rangle \in \mathcal{C}^\alpha [\![c]\!]_{I^\alpha},$$
$$\langle p(\tilde{x}), F_2 \rangle \in \mathcal{P}^\alpha [\![P]\!]_{I^\alpha},$$
$$F = lub(\{ F_1, F_2 \}) \}$$

$$\mathcal{F}^\alpha [\![P]\!] = \mathcal{P}^\alpha [\![P]\!] \uparrow \omega$$

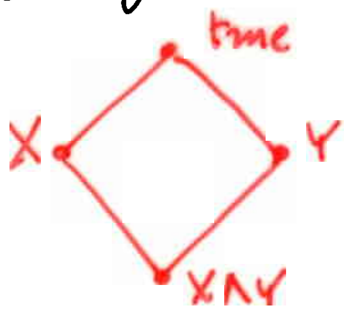
THE FIRST GROUNDNESS ABSTRACT DOMAIN

- concrete domain $(\wp(Eqns), \subseteq)$

  sets of sets of equations in solved form

- abstract domain $(Ground, \leq)$

  shown for 2 variables



- concretization function

$$\gamma(x) = \begin{cases} Eqns, & \text{if } x = true \\ \{e \mid x_i = t_i \in e,\ t_i \text{ ground}\}, & \text{if } x = x_1 \wedge \dots \wedge x_n \end{cases}$$

X represents the set of all sets of equations where X is bound to a ground term

- abstraction function

$$\alpha(y) = glb_{\leq} \begin{cases} true, & \text{if } y \subseteq Eqns \\ x_1 \wedge \dots \wedge x_k, & \text{if for every } e_i \in y, \\ & x_1 = t_1 \in e_i,\ t_1 \text{ ground} \\ & \vdots \\ & x_k = t_k \in e_i,\ t_k \text{ ground} \end{cases}$$

FROM $\wp(E_{qns})$ TO Ground

$$E_1 = \{\{X = f(Y, Z), W = a\}\}$$

$$\alpha_{Ground}(E_1) = W$$

$$E_2 = \{\{X = a\}, \{X = f(Y)\}\}$$

$$\alpha_{Ground}(E_2) = true$$

$$E_3 = \{\{X = a\}, \{Y = a\}\}$$

$$\alpha_{Ground}(E_3) = true$$

AN ABSTRACT SEMANTICS

P:

```
p(a,Y).
p(X,b).
q(X,X).
r(X,Y) :- p(X,Y), q(X,Y)
```

- the (concrete) s-semantics

$$F[\![P]\!] = \{ \langle p(X,Y), \{\{X=a\},\{Y=b\}\}\rangle, \langle q(X,Y), \{\{X=Y\}\}\rangle, \langle r(X,Y), \{\{X=a,Y=a\},\{X=b,Y=b\}\}\rangle \}$$

- observing groundness on the concrete semantics

$$\alpha_{Ground}(F[\![P]\!]) = \{ \langle p(X,Y), true\rangle, \langle q(X,Y), true\rangle, \langle r(X,Y), X \wedge Y\rangle \}$$

- computing the abstract semantics

$$\mathcal{P}^{\alpha_{Ground}}[\![P]\!] \uparrow 1 = \{ \langle p(X,Y), \underset{\text{lub}(\{X,Y\})}{true}\rangle, \langle q(X,Y), true\rangle \}$$

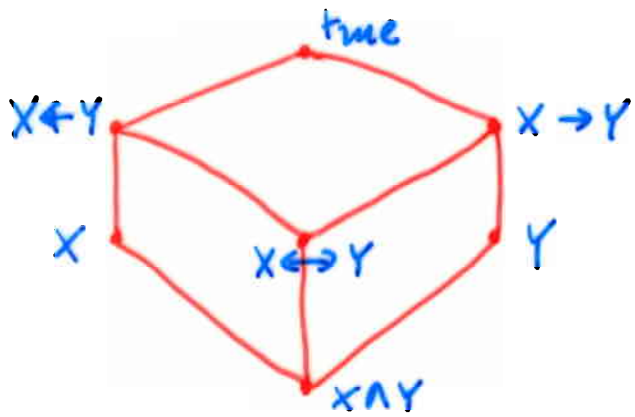
$$F^{\alpha}[\![P]\!] = \mathcal{P}^{\alpha_{Ground}}[\![P]\!] \uparrow 2 = \{ \langle p(X,Y), true\rangle, \langle q(X,Y), true\rangle, \langle r(X,Y), true\rangle \}$$

- $F^{\alpha}[\![P]\!]$ is less precise than $\alpha(F[\![P]\!])$

A BETTER DOMAIN FOR GROUNDNESS ANALYSIS: GROUND DEPENDENCIES

- concrete domain $(\mathcal{P}(Eqns), \subseteq)$
- abstract domain $(Def, \leq)$

shown for 2 variables



- concretization function

$$\gamma(x) = \begin{cases} Eqns, & \text{if } x = true \\ \{e \mid x_i = t_i \in e, t_i \text{ ground}\}, & \text{if } x = x_1 \wedge \ldots \wedge x_n \\ \{e \mid x_1 = t_1 \in e, y \in var(t_1)\}, & \text{if } x = x_1 \rightarrow y \\ \{e \mid x_1 = t_1 \in e, \{y_1, \ldots, y_k\} = var(t_1)\}, & \text{if } x = x_1 \leftrightarrow y_1 \wedge \ldots \wedge y_k \end{cases}$$

$X \wedge Y$ all the sets of equations where X and Y are bound to ground terms

$X \rightarrow Y$ all the sets of equations which contain the equation $X = t$, such that Y occurs in t (if X is ground then Y is ground too)

$X \leftrightarrow Y_1 \wedge Y_2$ all the sets of equations which contain $X = t$, such that Y_1 and Y_2 are the only variables occurring in t (if X is ground then Y_1 and Y_2 are ground and viceversa)

FROM $\wp(Eqns)$ TO Ground / Def

$$E_1 = \{\{X = f(Y, Z),\ W = a\}\}$$

$$\alpha_{Ground}(E_1) = W$$

$$\alpha_{Def}(E_1) = W \wedge (X \leftrightarrow Y \wedge Z) \wedge (X \to Y) \wedge (X \to Z)$$

$$E_2 = \{\{X = a\},\ \{X = f(Y)\}\}$$

$$\alpha_{Ground}(E_2) = true$$

$$\alpha_{Def}(E_2) = X \leftarrow Y$$

$$\mathrm{lub}(\{X,\ X \leftrightarrow Y\}) = X \leftarrow Y$$

$$E_3 = \{\{X = a\},\ \{Y = a\}\}$$

$$\alpha_{Ground}(E_3) = true$$

$$\alpha_{Def}(E_3) = true$$

THE ABSTRACT SEMANTICS ON DEF

P:

```
p(a, Y).
p(X, b).
q(X, X).
r(X, Y) :- p(X, Y), q(X, Y).
```

- the (concrete) s-semantics

$$F[\![P]\!] = \{ \langle p(X,Y), \{\{X=a\}, \{Y=b\}\}\rangle, \langle q(X,Y), \{\{X=Y\}\}\rangle, \langle r(X,Y), \{\{X=a, Y=a\}, \{X=b, Y=b\}\}\rangle \}$$

- observing DEF-groundness on the concrete semantics

$$\alpha_{Def}(F[\![P]\!]) = \{ \langle p(X,Y), true\rangle, \langle q(X,Y), X \Leftrightarrow Y\rangle, \langle r(X,Y), X \wedge Y\rangle \}$$

- computing the abstract semantics

$$T_P^{\alpha_{Def}}[\![P]\!] \uparrow 1 = \{ \langle p(X,Y), true\rangle, \langle q(X,Y), X \Leftrightarrow Y\rangle \}$$

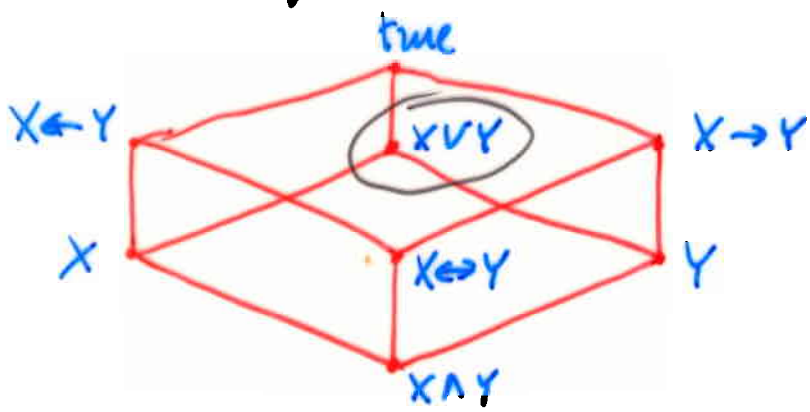
$$F^{\alpha_{Def}}[\![P]\!] = T_P^{\alpha_{Def}}[\![P]\!] \uparrow 2 = \{ \langle p(X,Y), true\rangle, \langle q(X,Y), X \Leftrightarrow Y\rangle, \langle r(X,Y), X \Leftrightarrow Y\rangle \}$$

- the abstract semantics is still less precise than the abstraction of the concrete semantics, yet it is better than $F^{\alpha_{Ground}}$

IMPROVING THE ABSTRACT DOMAIN FOR GROUNDNESS ANALYSIS: SOME DISJUNCTIVE INFORMATION

- concrete domain $(\wp(Eqns), \leq)$
- abstract domain $(Pos, \leq)$

shown for two variables



- concretization function
  - the one defined for Def plus the new case

$\{e \mid x_1 = t \in e, t \text{ ground or } y = t' \in e, t' \text{ ground}\}$, if $x = x_1 \vee y$

$x \vee y$ represents all the sets of equations where either x or y are bound to a ground term

FROM $\wp(\mathrm{Eqns})$ TO Ground / Def / Pos

$$E_1 = \{\{X = f(Y, Z),\ W = a\}\}$$

$$\alpha_{Ground}(E_1) = W$$

$$\alpha_{Def}(E_1) = W \wedge (X \leftrightarrow Y \wedge Z) \wedge (X \to Y) \wedge (X \to Z)$$

$$\alpha_{Pos}(E_1) = W \wedge (X \leftrightarrow Y \wedge Z) \wedge (X \to Y) \wedge (X \to Z)$$

$$E_2 = \{\{X = a\},\ \{X = f(Y)\}\}$$

$$\alpha_{Ground}(E_2) = true$$

$$\alpha_{Def}(E_2) = X \leftarrow Y$$

$$\alpha_{Pos}(E_2) = X \vee (X \leftrightarrow Y)$$

$lub(\{X, X \leftrightarrow Y\}) = X \leftarrow Y$

$$E_3 = \{\{X = a\},\ \{Y = a\}\}$$

$$\alpha_{Ground}(E_3) = true$$

$$\alpha_{Def}(E_3) = true$$

$$\alpha_{Pos}(E_3) = X \vee Y$$

THE ABSTRACT SEMANTICS ON POS

P:

```
p(a,Y).
p(X,b).
q(X,X).
r(X,Y) :- p(X,Y), q(X,Y).
```

the (concrete) s-semantics

$$F[\![P]\!] = \{ \langle p(X,Y), \{\{X=a\},\{Y=b\}\} \rangle, \langle q(X,Y), \{\{X=Y\}\} \rangle, \langle r(X,Y), \{\{X=a, Y=a\}, \{X=b, Y=b\}\} \rangle \}$$

observing Pos-groundness on the concrete semantics

$$\alpha_{Pos}(F[\![P]\!]) = \{ \langle p(X,Y), X \vee Y \rangle, \langle q(X,Y), X \Leftrightarrow Y \rangle, \langle r(X,Y), X \wedge Y \rangle \}$$

- computing the abstract semantics

$$P^{\alpha_{Pos}}_{Pos}[\![P]\!] \uparrow 1 = \{ \langle p(X,Y), X \vee Y \rangle, \langle q(X,Y), X \Leftrightarrow Y \rangle \}$$

$$F^{\alpha_{Pos}}[\![P]\!] = P^{\alpha_{Pos}}[\![P]\!] \uparrow 2 = \{ \langle p(X,Y), X \vee Y \rangle, \langle q(X,Y), X \Leftrightarrow Y \rangle, \langle r(X,Y), X \wedge Y \rangle \}$$

$glb(\{X \Leftrightarrow Y, X \vee Y\})$

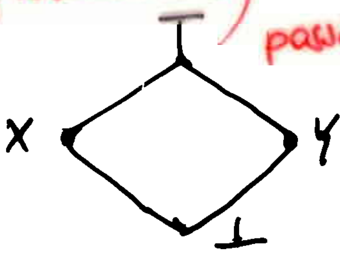
- the abstract semantics is more precise than $F^{\alpha_{Def}}$ and, on this example, as precise as the abstraction of the concrete semantics

REFINEMENT OPERATORS

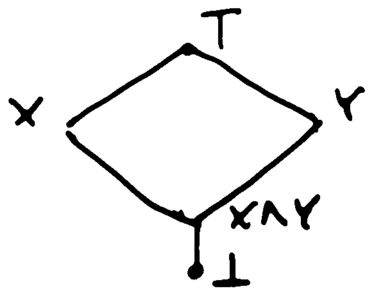
- the property (groundness)

possibly un-ground

x is definitely ground →



- conjunctive completion (to get a Moore family)



Ground

- Heyting completion (functional dependencies)

$$Def = Ground \rightarrow Ground$$

- another Heyting completion (looks like a disjunction)

$$Pos = Def \rightarrow Def$$

- $$Pos = Pos \rightarrow Pos$$
  - improvement in precision
  - some compositionality properties may be verified
- the same construction can be applied to other domains
  - types

RELATION TO THE SEMANTICS

- the abstract domain is indeed an abstraction of $\wp(Eqns)$ and the abstract semantics is an abstraction of the S-semantics

  - OK if one wants to analyze groundness of computed answers

  - if we want to know the groundness information for procedure calls (to optimize the compiled code) we have to combine
    - call patterns → semidenotational observable?
    - groundness abstraction for $\wp(Subst)$ or $\wp(Eqns)$

  - if we want a modular groundness analysis, we need to combine
    - an OR-compositional semantics (SLD-derivations, resultants
    - groundness abstraction → semiperfect observable?

  - if we want a top-down groundness analysis, we need to combine
    - a perfect observable
    - groundness abstraction → semiperfect observable?

TOP-DOWN VS. BOTTOM-UP

- top-down ([redacted] operational) analysis
  - we abstract the operational semantics (transition system)
    - composition of an abstraction on substitutions and an abstraction on the structure of SLD-derivations, corresponding to a perfect observable
      - SLD derivations
      - resultants
        - more abstract observables, such as computed answers, lead to imprecise computations.

- bottom-up (denotational) analysis
  - we abstract the denotational semantics (in particular, the suitable T_P-operator)
    - composition of an abstraction on substitutions and an abstraction on SLD-derivations, corresponding at least to a denotational observable
      - computed answers
      - correct answers
      - ground instances of correct answers
      - call patterns
      - partial answers
    - the bottom-up analysis is always performed by first analyzing the program (without the goal) and then possibly deriving the behaviour of the goal from the abstract semantics of the program.

GOAL-DEPENDENCE VS GOAL-INDEPENDENCE

- goal-independence
  - the analysis is the abstract semantics of the program

    the least fixpoint of the (denotational) abstract T_P operator
    - the collection of (operational) (abstract) behaviours for most general atomic goals
- goal-dependence
  - we only analyze the abstract behaviour of a specific goal
    - we apply the abstract transition system to the (abstract) goal
    - using denotational bottom-up methods the behaviour of a goal is always derived from the abstract semantics of the program (goal compositionality by construction)
- goal-dependent (top-down) analyses may sometimes give more precise results than goal-independent (top-down or bottom-up) analyses?
  - goal-independent computations are as precise as goal-dependent ones, if the abstraction is condensing, i.e. if the abstract behaviour derived from the goal independent abstract semantics is the same as the one that would be (top-down) computed for the specific goal

RELATION AMONG ABSTRACT SEMANTICS

- top-down (goal-independent) program denotation

$$\mathcal{O}_\alpha [\![P]\!]$$

- bottom-up (goal-independent) program denotation

$$\mathcal{F}_\alpha [\![P]\!] = P_\alpha [\![P]\!] \uparrow \omega$$

- goal-dependent top-down abstract semantics

$$\mathcal{B}_\alpha [\![G \text{ in } P]\!]$$

- denotational semantics of a goal

$$\mathcal{Q}_\alpha [\![G \text{ in } P]\!] = \mathcal{G}_\alpha [\![G]\!]_{\mathcal{F}_\alpha [\![P]\!]}$$

- in the concrete semantics

$$\mathcal{O} [\![P]\!] = \mathcal{F} [\![P]\!] = P_{\alpha} [\![P]\!] \uparrow \omega$$ (equivalence of top-down and bottom-up denotations)

$$\mathcal{B} [\![G \text{ in } P]\!] = \mathcal{Q} [\![G \text{ in } P]\!] = \mathcal{G}_\alpha [\![G]\!]_{\mathcal{F}_\alpha [\![P]\!]}$$ (condensing or goal-unfoldinality: goal-independent is equivalent to goal-dependent)

- perfect observables
  - equivalence top-down / bottom-up
  - equivalence goal-dependent / goal-independent
  - completeness of the abstract semantics
- denotational observables
  - completeness of the abstract denotational semantics
- semi perfect observables
  - top-down = bottom-up
  - goal-dependent = goal-independent

semi denotational observables

?

TOP-DOWN COMPUTATIONS WITH (SEMI) DENOTATIONAL OBSERVABLES

- we want to compute "operationally" computed answers (a denotational observable)
- we want to compute "operationally" computed answers abstracted to elements of Pos (a semidenotational observable)
    - the abstract transition systems are too imprecise
        - we cannot perform the abstraction at every computation step
    - we can choose a (more concrete) perfect (semi-perfect) observable, compute on that domain and then (at the end) perform the abstraction
        - how we compute answers by computing SLD-derivations or resultants
        - how we compute groundness information for answers by computing on "abstract" SLD-derivations or resultants
- in order to do something operationally we sometimes need to be more concrete than in the denotational case.

BACK TO THE GROUNDNESS DOMAINS

- how to do "top-down" operational abstract computations
  - SLD-derivations, where substitutions (equations) are replaced by formulas in Ground, Def or Pos
- we can check whether the various domains are "condensing", i.e. whether
  - the goal-dependent behaviour can be derived without losing precision from the goal-independent denotation
  - the abstract denotational semantics for a (usually more abstract domain) is also equivalent
- Ground and Def are not condensing, while Pos is.

Def IS NOT CONDENSING

```
p(a,Y).
p(X,b).
q(X,X).
```

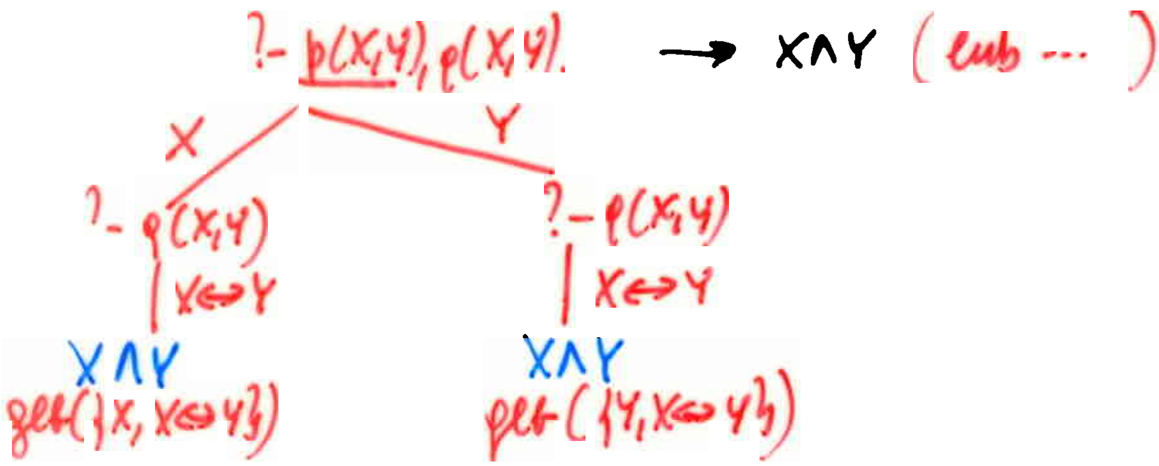
P

- goal-independent top-down abstract denotation

$$\mathcal{O}_\alpha[\![P]\!] = \{ \langle p(X,Y), \underbrace{true}_{lub(\{X,Y\})} \rangle, \langle q(X,Y), X \leftrightarrow Y \rangle \}$$

- goal-dependent top-down abstract semantics

$$\mathcal{B}_\alpha[\![?\text{-}\, p(X,Y), q(X,Y) \text{ in } P]\!] =$$



- derivation of the abstract goal semantics from $\mathcal{O}_\alpha[\![P]\!]$

$$glb(\{true, X \leftrightarrow Y\}) = X \leftrightarrow Y \geq X \wedge Y$$

(the same result would have been obtained by the denotational construction)

- Pos works correctly on this example and is indeed condensing

- the abstraction from $\wp(Eqns)$ to the abstract domain, takes place in

  the semantics of a clause

$$\mathcal{C}^\alpha[\![p(\tilde{t}) :- e, q(\tilde{t}_1), \dots, q_n(\tilde{t}_n)]\!]_{I^\alpha} =$$
$$\{ \langle p(\tilde{x}), F \rangle \mid \exists \langle q_i(\tilde{x}_i), F_i \rangle \in I^\alpha, F = glb(\{\alpha(\tilde{x}=\tilde{t}), \alpha(e), \alpha(\tilde{x}_1=\tilde{t}_1), \dots, \alpha(\tilde{x}_n=\tilde{t}_n), F_1, \dots, F_n\})|_{\alpha}\tilde{x} \}$$

  - the semantics of composition of clauses

$$\wp^\alpha[\![c ; p]\!]_{I^\alpha} = \{ \langle p(\tilde{x}), F \rangle \mid \exists \langle p(\tilde{x}), F_1 \rangle \in \mathcal{C}^\alpha[\![c]\!]_{I^\alpha}, \exists \langle p(\tilde{x}), F_2 \rangle \in \wp^\alpha[\![p]\!]_{I^\alpha}, F = lub(\{F_1, F_2\}) \}$$

- rather than taking the abstraction at any application of the semantic evaluation function

  we can abstract the set of (concrete) equations in the program text, obtaining an "abstract program"

  - the abstract semantics of the concrete program is the (regular) semantics (over a different domain) of the abstract program

  - more efficient, since we perform the abstraction once and for all at "compilation time" (translation time)

FROM THE LOGIC PROGRAM TO THE ABSTRACT PROGRAM

- 2 transformations

1. from logic program to equational CLP program

```
p(a,Y).
p(X,b).
q(X,X).
r(X,Y) :- p(X,Y), q(X,Y).
```

```
p(X,Y) :- X=a
p(X,Y) :- Y=b
q(X,Y) :- X=Y
r(X,Y) :- p(X,Y), q(X,Y)
```

2. from equational CLP program to abstract program

```
p(X,Y) :- α({X=a})
p(X,Y) :- α({Y=b})
q(X,Y) :- α({X=Y})
r(X,Y) :- p(X,Y), q(X,Y)
```

THE SEMANTICS OF THE ABSTRACT PROGRAM

$$\mathcal{C}[\![p(\tilde{x}_m) :- c, q_1(\tilde{x}_1), \dots, q_n(\tilde{x}_n)]\!]_{\mathcal{I}^\alpha} =$$

$$\{ \langle p(\tilde{x}), F \rangle \mid \langle q_i(\tilde{x}_i), F_i \rangle \in \mathcal{I}^\alpha \text{ (suitably renamed)} \quad F = glb(\{c, F_1, \dots, F_n\}|_{\bar{x}}) \}$$

- no more abstractions of concrete values
  glb's (or lub's) of abstract values coming from the (abstract) program and the current approximation of the semantics

AN EXAMPLE ON DEF

$$
\begin{aligned}
&p(X,Y) \mathrel{:-} \alpha(\{X=a\})\\
&p(X,Y) \mathrel{:-} \alpha(\{Y=b\})\\
&q(X,Y) \mathrel{:-} \alpha(\{X=Y\})\\
&r(X,Y) \mathrel{:-} p(X,Y), q(X,Y)
\end{aligned}
$$

$$
\begin{aligned}
&p(X,Y) \mathrel{:-} X\\
&p(X,Y) \mathrel{:-} Y\\
&q(X,Y) \mathrel{:-} X \leftrightarrow Y\\
&r(X,Y) \mathrel{:-} p(X,Y), q(X,Y)
\end{aligned}
$$

$$\mathit{true}$$

$$\Uparrow$$

$$\langle p(X,Y), \mathrm{lub}\{X,Y\} \rangle$$

$$\langle q(X,Y), X \leftrightarrow Y \rangle$$

$$\langle r(X,Y), \mathrm{glb}\{\mathit{true}, X \leftrightarrow Y\}$$

$$\Downarrow$$

$$X \leftrightarrow Y$$

AN EXAMPLE ON DEF

$$a(X,Y,Z) :- X=[\,],\ Y=Z$$
$$a(X,Y,Z) :- X=[X1|X2],\ Z=[X1|Z2],\ a(X2,Y,Z2)$$

$$a(X,Y,Z) :- X \wedge Y \leftrightarrow Z$$
$$a(X,Y,Z) :- X \leftrightarrow X1 \wedge X2,\ Z \leftrightarrow X1 \wedge Z2,\ a(X2,Y,Z2)$$

$$P^{\alpha}[\![P]\!]\uparrow 0 = \{\langle a(X,Y,Z), \bot \rangle\}$$

$$P^{\alpha}[\![P]\!]\uparrow 1 = \{\langle a(X,Y,Z), \underbrace{lub(\{X \wedge Y \leftrightarrow Z, \underbrace{glb(\{\bot, X \leftrightarrow X1 \wedge X2 \wedge Z \leftrightarrow X1 \wedge Z2\})}_{\bot}\})}_{X \wedge Y \leftrightarrow Z} \rangle\}$$

$$P^{\alpha}[\![P]\!]\uparrow 2 = \{\langle a(X,Y,Z), \underbrace{lub(\{X \wedge Y \leftrightarrow Z, \underbrace{glb(\{X2 \wedge Y \leftrightarrow Z2, X \leftrightarrow X1 \wedge X2 \wedge Z \leftrightarrow X1 \wedge Z2\})}_{Z \leftrightarrow (X \wedge Y)}\})}_{Z \leftrightarrow (X \wedge Y)} \rangle\}$$

- if we make another iteration we generate the same "interpretation" which is therefore the fixpoint