Representations

Séminaire au LIP

Juin 2025

Paul Brunet
Université Paris-Est Créteil, EPISEN, & LACL

Disclaimer

- ☞ Work in progress
- ☞ Unpublished
- ☞ Largely a modelling/formalisation effort $\rightarrow$ many definitions, few results

Context: Program analysis

☞ Impossible for a single system to solve all program analysis tasks.

Context: Program analysis

- ☞ Impossible for a single system to solve all program analysis tasks.
- ☞ Need for a large and growing number of program analysis frameworks.

Context: Program analysis

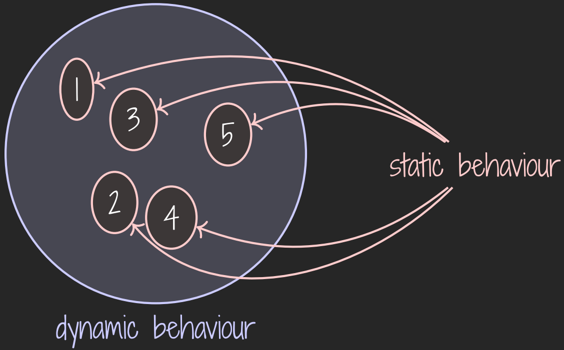
- ☞ Impossible for a single system to solve all program analysis tasks.
- ☞ Need for a large and growing number of program analysis frameworks.
- ☞ Need for tools to build new program analysis frameworks.

Context: Program analysis

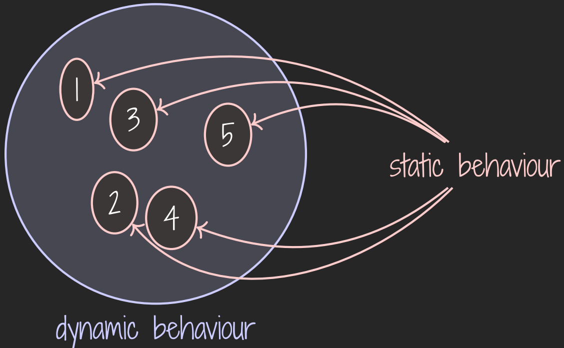
- ☛ Impossible for a single system to solve all program analysis tasks.
- ☛ Need for a large and growing number of program analysis frameworks.
- ☛ Need for tools to build new program analysis frameworks.

- ☛ This work: machinery to build new frameworks compositionally.

Two tiered models

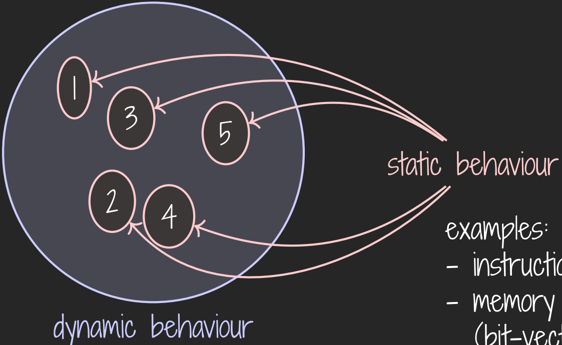


Two tiered models



examples:
- traces (i.e. words)
- concurrent traces (e.g. pomsets)
- graphs

Two tiered models



examples:
- traces (i.e. words)
- concurrent traces (e.g. pomsets)
- graphs

examples:
- instructions
- memory state (bit-vector)
- predicate/value record

Program analysis framework

Σ

signature

Program analysis framework

$$\begin{tikzcd}[column sep=small] & \underset{\text{signature}}{\Sigma} \arrow[dl] \arrow[dr] & \\ \underset{\text{logic}}{\vdash \varphi \to \psi} & & \underset{\text{semantics}}{\models \varphi \to \psi} \end{tikzcd}$$

Program analysis framework

$$\Sigma$$

signature

$$\vdash \varphi \to \psi$$

logic

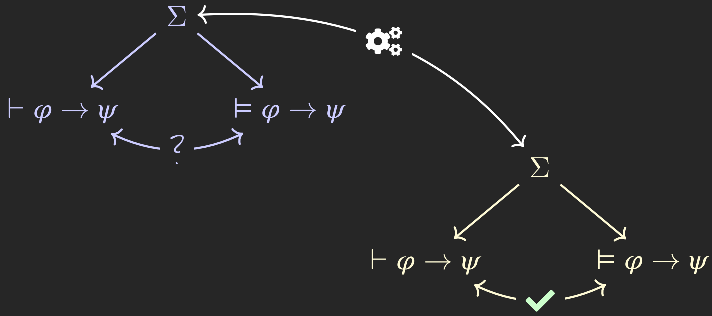
$$\models \varphi \to \psi$$

semantics

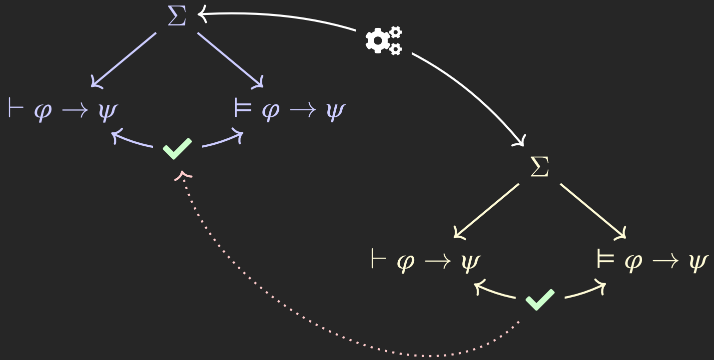
Reductions

$$\begin{tikzcd}
 & \Sigma \arrow[dl] \arrow[dr] & \\
\vdash \varphi \to \psi & & \vDash \varphi \to \psi \\
 & ? \arrow[ul, bend left] \arrow[ur, bend right] &
\end{tikzcd}$$

Reductions



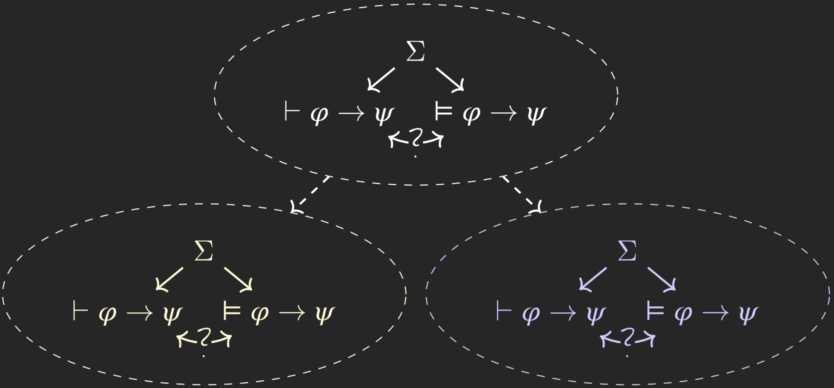
Reductions


Σ
$\vdash \varphi \to \psi$
$\vDash \varphi \to \psi$
Σ
$\vdash \varphi \to \psi$
$\vDash \varphi \to \psi$

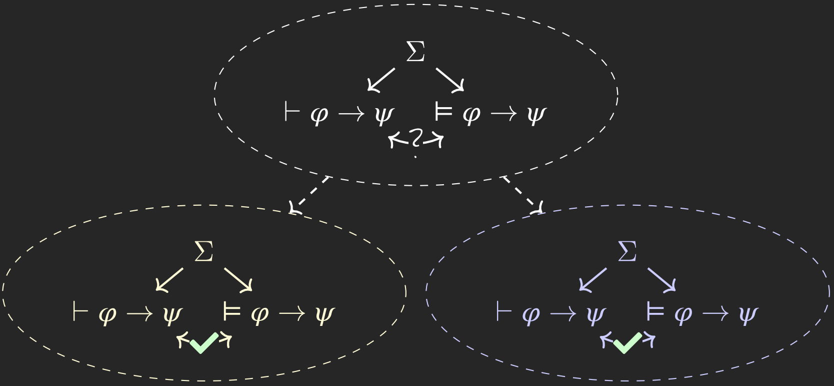
Decompositions

$$
\begin{tikzcd}
 & \Sigma \arrow[dl] \arrow[dr] & \\
\vdash \varphi \to \psi \arrow[rr, leftrightarrow, "?"'] & & \vDash \varphi \to \psi
\end{tikzcd}
$$

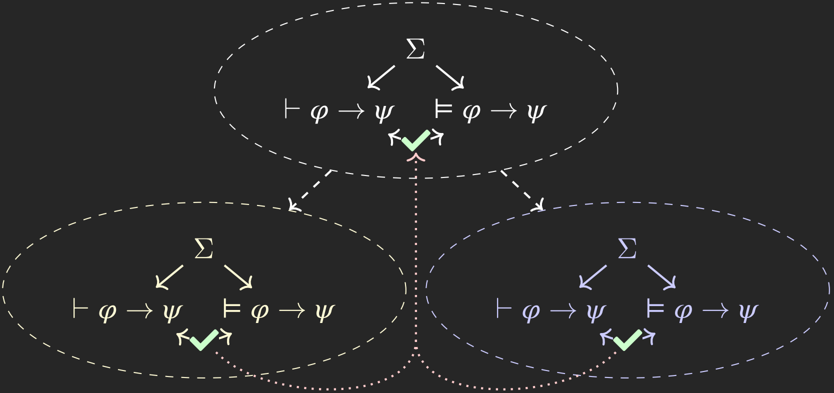
Decompositions



Decompositions



Decompositions



A meta-framework
Features

- ☞ Syntax as a parameter:

A meta-framework
Features

- ☞ Syntax as a parameter:
  - morphisms between structures with different signatures;

A meta-framework
Features

- ☞ Syntax as a parameter:
  - morphisms between structures with different signatures;
  - proof techniques for completeness, e.g. reductions.

A meta-framework
Features

- ☞ Syntax as a parameter:
  - morphisms between structures with different signatures;
  - proof techniques for completeness, e.g. reductions.
- ☞ Non-equal things should be different:

A meta-framework
Features

- ☞ Syntax as a parameter:
  - morphisms between structures with different signatures;
  - proof techniques for completeness, e.g. reductions.
- ☞ Non-equal things should be different:
  - work with explicit equivalence relations rather than quotients;

A meta-framework
Features

- ☞ Syntax as a parameter:
  - morphisms between structures with different signatures;
  - proof techniques for completeness, e.g. reductions.
- ☞ Non-equal things should be different:
  - work with explicit equivalence relations rather than quotients;
  - no identification of isomorphic structures.

A meta-framework
Features

- ☞ Syntax as a parameter:
  - morphisms between structures with different signatures;
  - proof techniques for completeness, e.g. reductions.
- ☞ Non-equal things should be different:
  - work with explicit equivalence relations rather than quotients;
  - no identification of isomorphic structures.
- ☞ Keep it simple:

A meta-framework
Features

- ☞ Syntax as a parameter:
  - morphisms between structures with different signatures;
  - proof techniques for completeness, e.g. reductions.
- ☞ Non-equal things should be different:
  - work with explicit equivalence relations rather than quotients;
  - no identification of isomorphic structures.
- ☞ Keep it simple:
  - as few hypotheses as possible;

A meta-framework

Features

- ☞ Syntax as a parameter:
  - morphisms between structures with different signatures;
  - proof techniques for completeness, e.g. reductions.
- ☞ Non-equal things should be different:
  - work with explicit equivalence relations rather than quotients;
  - no identification of isomorphic structures.
- ☞ Keep it simple:
  - as few hypotheses as possible;
  - diagrammatic or (in)equationnal reasonning.

Representations
Plan

I. Background

II. Representations

III. Reductions

IV. Higher order representations

V. Conclusion

Background

- Category theory (a little)
- Relation algebra (a lot)

Binary relations

$$x : A \multimap B \Leftrightarrow x \subseteq A \times B$$

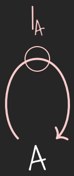
i.e. $x \subseteq y$

- ☞ entailment: $x \twoheadrightarrow y$ means $(a,b) \in x \Rightarrow (a,b) \in y$.
- ☞ equivalence: $x \leftrightarrow y$ means $(a,b) \in x \Leftrightarrow (a,b) \in y$.

i.e. $x=y$

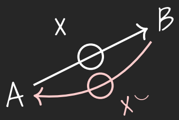
Relation algebra
A multiplicative fragment

- ☞ identity $\mathsf{I}_A : A \multimap A := \{(a, a') \mid a = a' \in A\}$.



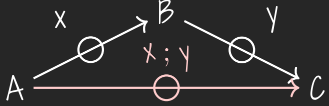
Relation algebra
A multiplicative fragment

- ☞ identity $1_A : A \twoheadrightarrow A := \{(a,a') \mid a = a' \in A\}$.
- ☞ converse $x^\smile : B \twoheadrightarrow A := \{(b,a) \mid (a,b) \in x\}$.



Relation algebra
A multiplicative fragment

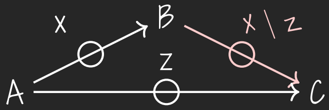
- identity $1_A : A \multimap A := \{(a, a') \mid a = a' \in A\}$.
- converse $x^\smile : B \multimap A := \{(b, a) \mid (a, b) \in x\}$.
- composition
  $x ; y : A \multimap C := \{(a, c) \mid \exists b \in B : (a, b) \in x \ \& \ (b, c) \in y\}$.



Relation algebra
A multiplicative fragment

- identity $1_A : A \multimap A := \{(a,a') \mid a = a' \in A\}$.
- converse $x^{\smile} : B \multimap A := \left\{(b,a) \;\middle|\; (a,b) \in x\right\}$.
- composition
  $x \,;\, y : A \multimap C := \left\{(a,c) \;\middle|\; \exists b \in B :\ (a,b) \in x \ \&\ (b,c) \in y\right\}$.
- left-residual
  $x \setminus z : B \multimap C := \left\{(b,c) \;\middle|\; \forall a \in A,\ (a,b) \in x \Rightarrow (a,c) \in z\right\}$.

$$\boxed{x\,;\,y \rightarrow z \quad\Leftrightarrow\quad y \rightarrow x \setminus z}$$



Functions and relations

☞ Functions can be represented as relations:
Let $f : A \to B$ be a function

$$f_* : A \looparrowright B := \{(x, f(x)) \mid x \in A\} \quad \text{i.e. } (a,b) \in f_* \Leftrightarrow f(a) = b.$$
$$f^* : B \looparrowright A := \{(f(x), x) \mid x \in A\} \quad \text{i.e. } (b,a) \in f^* \Leftrightarrow b = f(a).$$

Functions and relations

- Functions can be represented as relations:
  Let $f : A \to B$ be a function

$$f_* : A \multimap B := \{(x, f(x)) \mid x \in A\} \quad \text{i.e. } (a,b) \in f_* \Leftrightarrow f(a) = b.$$
$$f^* : B \multimap A := \{(f(x), x) \mid x \in A\} \quad \text{i.e. } (b,a) \in f^* \Leftrightarrow b = f(a).$$

- This is a monoid homomorphism:

$$(f \circ g)^* \Leftrightarrow f^* ; g^* \qquad (id_A)^* \Leftrightarrow 1_A$$

Functions and relations
Characterisation of functions

☞ Functions are <u>univalent</u> and <u>total</u>.

Functions and relations
Characterisation of functions

☞ Functions are <u>univalent</u> and <u>total</u>.

- univalence: $x^{\smile} ; x \rightarrow 1_B$
  (means that if $(a, b_1), (a, b_2) \in x$, then $b_1 = b_2$.)

Functions and relations
Characterisation of functions

☞ Functions are <u>univalent</u> and <u>total</u>.

- univalence: $x^{\smile} ; x \rightarrow 1_B$
  (means that if $(a, b_1), (a, b_2) \in x$, then $b_1 = b_2$.)
- totality: $1_A \rightarrow x ; x^{\smile}$
  (means that for any $a \in A$, there is $b \in B$ such that $(a, b) \in x$.)

Functions and relations
Characterisation of functions

- ☞ Functions are univalent and total.
  - univalence: $x^{\smile} \,;\, x \;\;\multimap\;\; 1_B$
    (means that if $(a, b_1), (a, b_2) \in x$, then $b_1 = b_2$.)
  - totality: $1_A \;\;\multimap\;\; x \,;\, x^{\smile}$
    (means that for any $a \in A$, there is $b \in B$ such that $(a, b) \in x$.)
- ☞ If a relation x is univalent and total there is a unique function f such that $x = f_*$.

Plan

I. Background

II. Representations

III. Reductions

IV. Higher order representations

V. Conclusion

Representations

$$R = \langle T, E, \models, \leq \rangle.$$

Representations

Set of traces

$$R = \langle T, E, \models, \leq \rangle.$$

Representations

Set of traces

$$R = \langle T, E, \models, \leq \rangle.$$

Set of expressions

Representations

Set of traces

Satisfaction relation

$$R = \langle T, E, \models, \leq \rangle.$$

Set of expressions

Representations

Set of traces

Satisfaction relation

$$R = \langle T, E, \models, \leq \rangle.$$

Set of expressions

(Pre)order relation

Representations

Set of traces

Satisfaction relation

$$R = \langle T, E, \models, \leq \rangle.$$

Set of expressions

(Pre)order relation

$$\boxed{\models ; \leq \;\rightharpoonup\; \models}$$

Representations

Set of traces

Satisfaction relation

$$R = \langle T, E, \models, \leq \rangle.$$

Set of expressions

(Pre)order relation

$$\boxed{\models\ ;\ \leq\ \rightharpoonup\ \models}$$

$$\forall t \in T,\ f \in E,\ \left(\exists e \in E : t \models e \ \&\ e \leq f\right) \Rightarrow t \models f$$

Representations

Set of traces

Satisfaction relation

$$R = \langle T, E, \models, \leq \rangle.$$

Set of expressions

(Pre)order relation

$$\boxed{\models ; \leq \ \rightarrow \ \models}$$

$$\forall t \in T,\ f \in E,\ \left(\exists e \in E : t \models e \ \&\ e \leq f\right) \Rightarrow t \models f$$

$$\boxed{\Leftrightarrow \forall e, f \in E,\ e \leq f \Rightarrow \forall t,\ t \models e \Rightarrow t \models f}$$

Representations

Set of traces

Satisfaction relation

$$R = \langle T, E, \models, \leq \rangle.$$

$$\boxed{\begin{array}{ccc} 1 & \rightarrow & \leq \\ \leq ; \leq & \rightarrow & \leq \end{array}}$$

Set of expressions

(Pre)order relation

$$\boxed{\models ; \leq \;\rightarrow\; \models}$$

$$\forall t \in T,\ f \in E,\ \left(\exists e \in E : t \models e \;\&\; e \leq f\right) \Rightarrow t \models f$$

$$\boxed{\Leftrightarrow \forall e, f \in E,\ e \leq f \Rightarrow \forall t,\ t \models e \Rightarrow t \models f}$$

Interpretations

$$R = \langle T, E, \vDash, \leq \rangle.$$

Equivalently, we can replace $\vDash$ with an interpretation function $I : E \to \mathcal{P}(T)$:

$$I(e) := \{t \in T \mid t \vDash e\}$$

Interpretations

$$R = \langle T, E, \vDash, \leq \rangle.$$

Equivalently, we can replace $\vDash$ with an interpretation function $I : E \to \mathcal{P}(T)$:

$$I(e) := \{t \in T \mid t \vDash e\}$$

To get back $\vDash$, we use the fact that $t \vDash e \Leftrightarrow t \in I(e)$ i.e. $\vDash \, := \, \in ; I^*$.

Interpretations

$$R = \langle T, E, \vDash, \leq \rangle.$$

Equivalently, we can replace $\vDash$ with an interpretation function $I : E \to \mathcal{P}(T)$:

$$I(e) := \{t \in T \mid t \vDash e\}$$

To get back $\vDash$, we use the fact that $t \vDash e \Leftrightarrow t \in I(e)$ i.e. $\vDash := \in ; I^*$.

The axiom of representations means that $\leq$ is sound with respect to I:

$$\boxed{(\vDash ; \leq \;\rightharpoonup\; \vDash) \Leftrightarrow (\forall e, f \in E, e \leq f \Rightarrow I(e) \subseteq I(f))}$$

Exact representations

A representation is <u>exact</u> if the following entailment holds:

$$\boxed{\models \setminus \models \ \rightharpoonup\ \leq .}$$

I.e. $\forall e, f,\ (\forall t,\ t \models e \Rightarrow t \models f) \Rightarrow e \leq f$.

Exact representations

A representation is exact if the following entailment holds:

$$\boxed{\vDash \backslash \vDash \;\rightarrow\; \leq\,.}$$

I.e. $\forall e, f,\ (\forall t,\ t \vDash e \Rightarrow t \vDash f) \Rightarrow e \leq f.$

lemma

A representation is exact if and only if $\leq$ is sound and complete for I, i.e.:

$$\forall e, f \in E,\ I(e) \subseteq I(f) \Leftrightarrow e \leq f.$$

Representations

Set of traces

Satisfaction relation

$$R = \langle T, E, \models, \leq \rangle.$$

Set of expressions

(Pre)order relation

$$\boxed{I \rightarrow \leq}$$

$$\boxed{\leq ; \leq \rightarrow \leq}$$

$$\boxed{\models ; \leq \rightarrow \models}$$

exact:

$$\boxed{\models \backslash \models \rightarrow \leq}$$

Examples

$$R = \langle T, E, \models, \leq \rangle.$$

- ☞ Algebra + axiomatisation
- ☞ Logic + semantics
- ☞ Programs + behavioural equivalence
- ☞ Programs + evaluation (exactness less relevant)

Trivial representations

Fact: for every binary relation x, $x \backslash x$ is a preorder.

Trivial representations

Fact: for every binary relation x, $x \setminus x$ is a preorder.

lemma

Let $x : A \multimap B$, then $R^x = \langle A, B, x, x \setminus x \rangle$ is an exact representation.

Trivial representations

Fact: for every binary relation x, $x \setminus x$ is a preorder.

lemma

Let $x : A \multimap B$, then $R^x = \langle A, B, x, x \setminus x \rangle$ is an exact representation.

Example:
Since $\in \setminus \in \;\Leftrightarrow\; \subseteq$ (easy exercise) $R^\in = \langle A, \mathcal{P}(A), \in, \subseteq \rangle$ is exact.

Representations
Plan

I. Background

II. Representations

III. Reductions

IV. Higher order representations

V. Conclusion

Reductions
A proof technique

☛ I want to prove that a representation is exact.

Reductions
A proof technique

- I want to prove that a representation is exact.
- To do that, I want to leverage the fact that I know some exact representations.

Reductions
A proof technique

- I want to prove that a representation is exact.
- To do that, I want to leverage the fact that I know some exact representations.

- I want to transfer the problem of checking exactness in R to the problem of proving exactness in R'.

Reductions
A proof technique

- ☞ I want to prove that a representation is exact.
- ☞ To do that, I want to leverage the fact that I know some exact representations.

- ☞ I want to transfer the problem of checking exactness in R to the problem of proving exactness in R'.
- ☞ I need some transformations between R and R'.

Reductions

A proof technique

$$R_1 = \langle T_1, E_1, \models_1, \leq_1 \rangle \qquad R_2 = \langle T_2, E_2, \models_2, \leq_2 \rangle$$

Reductions
A proof technique

$$R_1 = \langle T_1, E_1, \models_1, \leq_1 \rangle \qquad\qquad R_2 = \langle T_2, E_2, \models_2, \leq_2 \rangle$$

A reduction from R_1 to R_2 is given by:

- ☞ a pair of function $\varphi : E_1 \to E_2$ and $\psi : E_2 \to E_1$
- ☞ a binary relation $\rho : T_2 \leftrightarrow T_1$;

Reductions

A proof technique

$$R_1 = \langle T_1, E_1, \models_1, \leq_1 \rangle \qquad\qquad R_2 = \langle T_2, E_2, \models_2, \leq_2 \rangle$$

A reduction from R_1 to R_2 is given by:

- ☞ a pair of function $\varphi : E_1 \to E_2$ and $\psi : E_2 \to E_1$
- ☞ a binary relation $\rho : T_2 \leftrightarrow T_1$;

such that

$$\equiv_1 \;\Leftrightarrow\; \leq_1 \cap \geq_1$$

$$\boxed{\psi^* ; \leq_2 \;\Rightarrow\; \leq_1 ; \psi^* \qquad \psi^* ; \varphi^* \;\Rightarrow\; \equiv_1 \qquad \models_2 ; \varphi^* \;\Leftrightarrow\; \rho ; \models_1}$$

Reductions
A proof technique

$$R_1 = \langle T_1, E_1, \models_1, \leq_1 \rangle \qquad\qquad R_2 = \langle T_2, E_2, \models_2, \leq_2 \rangle$$

A reduction from R_1 to R_2 is given by:

- ☞ a pair of function $\varphi : E_1 \to E_2$ and $\psi : E_2 \to E_1$
- ☞ a binary relation $\rho : T_2 \multimap T_1$;

such that

$$\equiv_1 \Leftrightarrow \leq_1 \cap \geq_1$$

$$\boxed{\psi^*; \leq_2 \ \rightarrow\ \leq_1 ;\psi^* \qquad \psi^* ; \varphi^* \ \rightarrow\ \equiv_1 \qquad \models_2 ;\varphi^* \ \Leftrightarrow\ \rho; \models_1}$$

I.e. : $\qquad e \leq_2 e' \Rightarrow \psi(e) \leq_1 \psi(e') \qquad e \leq_1 \psi(\varphi(e)) \leq_1 e$

$$I_2(\varphi(e)) = \{t \in T_2 \mid \exists s, (t,s) \in \rho \ \&\ s \in I_1(e)\}$$

Reductions
A proof technique

lemma

If R_2 is exact and R_1 reduces to R_2, then R_1 is exact.

Reductions

A proof technique

lemma

If R_2 is exact and R_1 reduces to R_2, then R_1 is exact.

lemma

The reduction relation is a preorder.

Reductions
A proof technique

lemma

If R_2 is exact and R_1 reduces to R_2, then R_1 is exact.

lemma

The reduction relation is a preorder.

lemma

A representation R is exact iff R reduces to $\langle T, E, \models, \models \setminus \models \rangle$.

Proof

If R_2 is exact and R_1 reduces to R_2, then R_1 is exact.

1) Show that $\vDash_1 \setminus \vDash_1 \;\rightarrow\; \varphi_*;\vDash_2 \setminus \vDash_2;\varphi^*$.

Proof

If R_2 is exact and R_1 reduces to R_2, then R_1 is exact.

1) Show that $\vDash_1 \setminus \vDash_1 \rightarrow \varphi_* ; \vDash_2 \setminus \vDash_2 ; \varphi^*$.
   - Since $f_* ; x \setminus y ; g^* \leftrightarrow (x ; f^*) \setminus (y ; g^*)$

Proof

If R_2 is exact and R_1 reduces to R_2, then R_1 is exact.

1) Show that $\vDash_1 \setminus \vDash_1 \;\to\; \varphi_*; \vDash_2 \setminus \vDash_2 ;\varphi^*$.
   - Since $f_* ; x \setminus y ; g^* \;\leftrightarrow\; (x ; f^*) \setminus (y ; g^*)$
   - this is equivalent to showing $\vDash_2 ;\varphi^* ; (\vDash_1 \setminus \vDash_1) \;\to\; \vDash_2 ;\varphi^*$.

Proof

If R_2 is exact and R_1 reduces to R_2, then R_1 is exact.

1) Show that $\vDash_1 \backslash \vDash_1 \;\rightarrow\; \varphi_*; \vDash_2 \backslash \vDash_2 ; \varphi^*$.
   - Since $f_* ; x \backslash y ; g^* \;\leftrightarrow\; (x ; f^*) \backslash (y ; g^*)$
   - this is equivalent to showing $\vDash_2 ; \varphi^* ; (\vDash_1 \backslash \vDash_1) \;\rightarrow\; \vDash_2 ; \varphi^*$.
   - which we can derive like so:

$$\vDash_2 ; \varphi^* ; (\vDash_1 \backslash \vDash_1) \;\leftrightarrow\; \rho ; \vDash_1 ; (\vDash_1 \backslash \vDash_1) \;\rightarrow\; \rho ; \vDash_1 \;\leftrightarrow\; \vDash_2 ; \varphi^*$$

Proof

If R_2 is exact and R_1 reduces to R_2, then R_1 is exact.

1) Show that $\vDash_1 \backslash \vDash_1 \;\to\; \varphi_*; \vDash_2 \backslash \vDash_2 ;\varphi^*$.
   - Since $f_* ; x \backslash y ; g^* \;\leftrightarrow\; (x ; f^*) \backslash (y ; g^*)$
   - this is equivalent to showing $\vDash_2 ;\varphi^* ; (\vDash_1 \backslash \vDash_1) \;\to\; \vDash_2 ;\varphi^*$.
   - which we can derive like so:

$$\vDash_2 ;\varphi^* ; (\vDash_1 \backslash \vDash_1) \;\leftrightarrow\; \rho; \vDash_1 ;(\vDash_1 \backslash \vDash_1) \;\to\; \rho; \vDash_1 \;\leftrightarrow\; \vDash_2 ;\varphi^*$$

2) Therefore, we have :

$$\begin{aligned}\vDash_1 \backslash \vDash_1 \;\to\; \varphi_*; \vDash_2 \backslash \vDash_2 ;\varphi^* \;&\to\; \varphi_*; \leq_2 ;\varphi^* \\ &\to\; \varphi_* ; \psi_* ; \psi^*; \leq_2 ;\varphi^* \\ &\to\; \varphi_* ; \psi_*; \leq_1 ;\psi^* ; \varphi^* \\ &\to\; \equiv_1 ; \leq_1 ; \equiv_1 \;\to\; \leq_1 \qquad \square\end{aligned}$$

Reduction: example

We are interested in the following representation:

$$\boxed{\mathsf{FinLat}(A) = \langle A, E(A), \in ; \mathsf{l}^*, \leq \rangle}$$

$$e, f \in E(A) ::= a \mid \bot \mid e \vee f$$

$$\mathsf{l}(a) := \{a\} \qquad \mathsf{l}(\bot) = \emptyset \qquad \mathsf{l}(e \vee f) = \mathsf{l}(e) \cup \mathsf{l}(f)$$

$$e \leq e \qquad 0 \leq e \qquad e \leq e \vee f \qquad f \leq e \vee f$$

$$\frac{e \leq f \leq g}{e \leq g} \qquad \frac{e \leq g \quad f \leq g}{e \vee f \leq g}$$

Reduction: example

We are interested in the following representation:

$$\boxed{\mathrm{FinLat}(A) = \langle A, E(A), \in ;\mathrm{l}^*, \leq \rangle}$$

$$e, f \in E(A) ::= a \mid \bot \mid e \vee f$$

$$\mathrm{l}(a) := \{a\} \qquad \mathrm{l}(\bot) = \emptyset \qquad \mathrm{l}(e \vee f) = \mathrm{l}(e) \cup \mathrm{l}(f)$$

$$e \leq e \qquad 0 \leq e \qquad e \leq e \vee f \qquad f \leq e \vee f$$

$$\frac{e \leq f \leq g}{e \leq g} \qquad \frac{e \leq g \quad f \leq g}{e \vee f \leq g}$$

Its soundness is obvious, we now want to prove it exact.

Reduction: example

We start with a trivial representation

$$\boxed{\mathsf{FinSet}(A) = \langle A, A^\star, \ln, \ln \setminus \ln \rangle}$$

with $\ln$ defined by $(a, b \cdot w) \in \ln \Leftrightarrow a = b \vee (a, w) \in \ln$.

Reduction: example

We start with a trivial representation

$$\boxed{\mathsf{FinSet}(A) = \langle A, A^\star, \mathsf{In}, \mathsf{In} \setminus \mathsf{In} \rangle}$$

with In defined by $(a, b \cdot w) \in \mathsf{In} \Leftrightarrow a = b \vee (a, w) \in \mathsf{In}$.

$$\boxed{\text{FinLat reduces to FinSet.}}$$

$$\varphi(a) = a \qquad \varphi(\bot) = \varepsilon \qquad \varphi(e \vee f) = \varphi(e) \cdot \varphi(v)$$

$$\psi(\varepsilon) = \bot \qquad \psi(a \cdot w) = a \vee \psi(w) \qquad (\rho \quad \Leftrightarrow \quad I_A)$$

FinLat reduces to FinSet

1) $\psi^*; \leq_2 \; \rightharpoonup \; \leq_1 ; \psi^*$

FinLat reduces to FinSet

1) $\psi^*; \leq_2 \rightharpoonup \leq_1 ; \psi^*$

$u\ (\mathrm{In} \setminus \mathrm{In})\ v \Leftrightarrow \psi(u) \leq \psi(v)$

by induction on u:

FinLat reduces to FinSet

1) $\psi^*; \leq_2 \rightharpoonup \leq_1 ; \psi^*$

$u\ (\ln \setminus \ln)\ v \Leftrightarrow \psi(u) \leq \psi(v)$

by induction on u:

☞ $\varepsilon\ (\ln \setminus \ln)\ v \Rightarrow \psi(\varepsilon) = \bot \leq \psi(v)$

FinLat reduces to FinSet

1) $\psi^*; \leq_2 \;\rightharpoonup\; \leq_1 ; \psi^*$

$u \;(\ln \setminus \ln)\; v \Leftrightarrow \psi(u) \leq \psi(v)$

by induction on u:

- ☞ $\varepsilon \;(\ln \setminus \ln)\; v \Rightarrow \psi(\varepsilon) = \bot \leq \psi(v)$
- ☞ $a \cdot u \;(\ln \setminus \ln)\; v \Rightarrow a \ln v$ and $u \;(\ln \setminus \ln)\; v$
  by induction on v $a \ln v \Rightarrow a \leq \psi(v)$
  by induction hypothesis $\psi(u) \leq \psi(v)$
  $\Rightarrow \psi(a \cdot u) = a \vee \psi(u) \leq \psi(v)$

FinLat reduces to FinSet

2) $\psi^* \, ; \varphi^* \quad \rightharpoondown \; \equiv_1$

FinLat reduces to FinSet

2) $\psi^* ; \varphi^* \;\rightarrow\; \equiv_l$

by induction on terms

$$\psi(\varphi(a)) = a \vee \bot \equiv a \qquad\qquad \psi(\varphi(\bot)) = \psi(\varepsilon) = \bot$$

$$\cfrac{\cfrac{\text{by induction on } u}{\psi(u \cdot v) \equiv \psi(u) \vee \psi(v)}}{\psi(\varphi(e \vee f)) = \psi(\varphi(e) \cdot \varphi(f)) \equiv \psi(\varphi(e)) \vee \psi(\varphi(f))}$$

FinLat reduces to FinSet

3) $\models_2 ; \varphi^* \Leftrightarrow \rho ; \models_1$

FinLat reduces to FinSet

3) $\models_2 ; \varphi^* \iff \rho ; \models_1$

$a \ln \varphi(e) \Leftrightarrow a \in l(e)$
by induction on e

FinLat reduces to FinSet

3) $\models_2 ; \varphi^* \quad \Leftrightarrow \quad \rho ; \models_1$

$a \ln \varphi(e) \Leftrightarrow a \in \mathrm{l}(e)$

by induction on e

$$a \ln \varphi(b) = b \cdot \varepsilon \Leftrightarrow a = b \Leftrightarrow a \in \{b\} = \mathrm{l}(b)$$

$$\neg\,(a \ln \varphi(\bot) = \varepsilon) \qquad\qquad \neg(a \in \emptyset = \mathrm{l}(\bot))$$

$$\begin{aligned} a \ln \varphi(e \vee f) = \varphi(e) \cdot \varphi(f) &\Leftrightarrow a \ln \varphi(e) \text{ or } a \ln \varphi(f) \\ &\Leftrightarrow a \in \mathrm{l}(e) \text{ or } a \in \mathrm{l}(f) \\ &\Leftrightarrow a \in \mathrm{l}(e) \cup \mathrm{l}(f) = \mathrm{l}(e \vee f) \end{aligned}$$

FinLat reduces to FinSet

3) $\models_2 ; \varphi^* \quad \Leftrightarrow \quad \rho ; \models_1$

$a \ln \varphi(e) \Leftrightarrow a \in l(e)$
by induction on e

$$a \ln \varphi(b) = b \cdot \varepsilon \Leftrightarrow a = b \Leftrightarrow a \in \{b\} = l(b)$$

$$\neg\,(a \ln \varphi(\bot) = \varepsilon) \qquad\qquad \neg(a \in \emptyset = l(\bot))$$

$$\begin{aligned} a \ln \varphi(e \vee f) = \varphi(e) \cdot \varphi(f) &\Leftrightarrow a \ln \varphi(e) \text{ or } a \ln \varphi(f) \\ &\Leftrightarrow a \in l(e) \text{ or } a \in l(f) \\ &\Leftrightarrow a \in l(e) \cup l(f) = l(e \vee f) \end{aligned}$$

$\square$

Representations
Plan

I. Background

II. Representations

III. Reductions

IV. Higher order representations

V. Conclusion

Higher order representations

☞ Parametric representations over sets

Higher order representations

- ☞ Parametric representations over sets
- ☞ e.g. $\mathsf{FinLat}(A)$, $\mathsf{FinSet}(A)$, $\mathsf{Lang}(A)$, $\mathsf{Mon}(A)$

Higher order representations

- ☞ Parametric representations over sets
- ☞ e.g. FinLat(A), FinSet(A), Lang(A), Mon(A)
- ☞ i.e. structures $\mathcal{R}(A) := \langle \mathcal{T}(A), \mathcal{E}(A), \models_A, \leqq_A \rangle$

Higher order representations

- Parametric representations over sets
- e.g. $\mathsf{FinLat}(A)$, $\mathsf{FinSet}(A)$, $\mathsf{Lang}(A)$, $\mathsf{Mon}(A)$
- i.e. structures $\mathcal{R}(A) := \langle \mathcal{T}(A), \mathcal{E}(A), \models_A, \leqq_A \rangle$
- $\rightarrow$ functors from Set to Repr

Higher order representations

- Parametric representations over sets
- e.g. $\mathsf{FinLat}(A)$, $\mathsf{FinSet}(A)$, $\mathsf{Lang}(A)$, $\mathsf{Mon}(A)$
- i.e. structures $\mathcal{R}(A) := \langle \mathcal{T}(A), \mathcal{E}(A), \models_A, \leqq_A \rangle$
- $\rightarrow$ functors from Set to Repr

- Parametric representations over representations

Higher order representations

- ☞ Parametric representations over sets
- ☞ e.g. FinLat(A), FinSet(A), Lang(A), Mon(A)
- ☞ i.e. structures $\mathcal{R}(A) := \langle \mathcal{T}(A), \mathcal{E}(A), \models_A, \leqq_A \rangle$
- ☞ $\to$ functors from Set to Repr

- ☞ Parametric representations over representations
- ☞ $\to$ functors from Repr to Repr ?

The category Repr
Representation morphisms

$$\langle T_1, E_1, \models_1, \leq_1 \rangle \xrightarrow{\langle \varphi, \psi \rangle} \langle T_2, E_2, \models_2, \leq_2 \rangle.$$

with $\varphi : T_1 \to T_2$ and $\psi : E_1 \to E_2$, and such that:

The category Repr
Representation morphisms

$$\langle T_1, E_1, \models_1, \leq_1 \rangle \xrightarrow{\langle \varphi, \psi \rangle} \langle T_2, E_2, \models_2, \leq_2 \rangle.$$

with $\varphi : T_1 \to T_2$ and $\psi : E_1 \to E_2$, and such that:

$$\boxed{\psi^*; \leq_1 \ \rightharpoonup \ \leq_2 ; \psi^*}$$

$$\forall e, f \in E_1,\ e \leq_1 f \Rightarrow \psi(e) \leq_2 \psi(f).$$

The category Repr
Representation morphisms

$$\langle T_1, E_1, \vDash_1, \leq_1 \rangle \xrightarrow{\langle \varphi, \psi \rangle} \langle T_2, E_2, \vDash_2, \leq_2 \rangle.$$

with $\varphi : T_1 \to T_2$ and $\psi : E_1 \to E_2$, and such that:

$$\boxed{\psi^*; \leq_1 \ \rightharpoonup \ \leq_2 ; \psi^*}$$

$$\forall e, f \in E_1,\ e \leq_1 f \Rightarrow \psi(e) \leq_2 \psi(f).$$

$$\boxed{\varphi^*; \vDash_1 \ \leftrightharpoons \ \vDash_2 ; \psi^*}$$

$$\forall e \in E_1,\ \{\varphi(t) \mid t \vDash_1 e\} = \{t \in T_2 \mid t \vDash_2 \psi(e)\}$$

$$\text{i.e. } I_2(\psi(e)) = \{\varphi(t) \mid t \in I_1(e)\}$$

The category Repr

- ☞ Objects: representations
- ☞ Arrows: morphisms

Remark: exact representations form a full sub-category of Repr.

Higher order representations

- ☞ Parametric representations over sets
- ☞ e.g. FinLat(A), FinSet(A), Lang(A), Mon(A)
- ☞ i.e. structures $\mathcal{R}(A) := \langle \mathcal{T}(A), \mathcal{E}(A), \equiv_A, \leq_A \rangle$
- ☞ $\to$ functors from Set to Repr functions from sets to relations
  $\to$ "higher order relations"
- ☞ Parametric representations over representations
- ☞ $\to$ functors from Repr to Repr ?

Higher order relations

Let F, G be Set endofunctors.

☞ Set-indexed familly of relations $\rho_A : FA \rightarrowtriangle GA$.

$$\begin{tikzcd} \mathrm{Set} \arrow[r, bend left, "F"] \arrow[r, bend right, "G"'] & \mathrm{Set} \end{tikzcd}$$

Higher order relations

Let F, G be Set endofunctors.

- ☞ Set-indexed familly of relations $\rho_A : FA \mathrel{-\!\!\ominus\!\!\!\rightarrow} GA$.
- ☞ notation : $\rho : F \mathrel{=\!\!\ominus\!\!\!\Rightarrow} G$ if for all $f : A \to B$ <u>"natural relation"</u>

$$(x, y) \in \rho_A \Rightarrow (Ff(x), Gf(y)) \in \rho_B \qquad \text{i.e. } \rho \,;\, Gf_* \;\; -\!\!\triangleright \;\; Ff_* \,;\, \rho$$

$$\begin{tikzcd} \mathrm{Set} \arrow[r, bend left, "F"] \arrow[r, bend right, "G"'] & \mathrm{Set} \end{tikzcd}$$

Higher order relations

Let F, G be Set endofunctors.

☞ Set-indexed familly of relations $\rho_A : FA \mathrel{-\!\!\ominus\!\!\rightarrow} GA$.

☞ notation : $\rho : F \mathrel{=\!\!\ominus\!\!\Rightarrow} G$ if for all $f : A \to B$ <u>"natural relation"</u>

$$(x, y) \in \rho_A \Rightarrow (Ff(x), Gf(y)) \in \rho_B \qquad \text{i.e. } \rho \,;\, Gf_* \;\rightarrowtriangle\; Ff_* \,;\, \rho$$

☞ $\approx$ 2-cell in Rel

$$\begin{tikzcd}
\mathrm{Set} \arrow[r, bend left, "F"] \arrow[r, bend right, "G"'] \arrow[r, phantom, "\rho \Downarrow"] & \mathrm{Set}
\end{tikzcd}$$

Higher order relations

Let F, G be Set endofunctors.

- ☞ Set-indexed familly of relations $\rho_A : FA \not\rightarrow GA$.
- ☞ notation : $\rho : F \Rrightarrow G$ if for all $f : A \to B$ <u>"natural relation"</u>

$$(x, y) \in \rho_A \Rightarrow (Ff(x), Gf(y)) \in \rho_B \qquad \text{i.e. } \rho ; Gf_* \;\rightarrow\; Ff_* ; \rho$$

- ☞ $\approx$ 2-cell in Rel

- ☞ but not quite: F, G need to be $\mathrm{Set} \to \mathrm{Rel}$ functors!

$$\begin{tikzcd} \mathrm{Set} \arrow[r, bend left, "F_*"] \arrow[r, bend right, "G_*"'] \arrow[r, phantom, "\rho \Downarrow"] & \mathrm{Rel} \end{tikzcd}$$

- ☞ $F_* : \mathrm{Set} \to \mathrm{Rel}$ does as F on objects, and maps f to $(Ff)_*$.

Relation lifting

Let $F : \mathsf{Set} \to \mathsf{Set}$ be a functor preserving weak pullbacks.
F can be lifted to a functor $\overline{F} : \mathsf{Rel} \to \mathsf{Rel}$ such that:

$$\overline{F}(f^*) \Leftrightarrow (Ff)^* \qquad \frac{x \rightarrow y}{\overline{F}x \rightarrow \overline{F}y} \qquad \overline{F}(x^\smile) \Leftrightarrow (\overline{F}x)^\smile$$

Relation lifting

Let $F : \mathrm{Set} \to \mathrm{Set}$ be a functor preserving weak pullbacks.
F can be lifted to a functor $\bar{F} : \mathrm{Rel} \to \mathrm{Rel}$ such that:

$$\bar{F}(f^*) \leftrightarrow (Ff)^* \qquad \frac{x \to y}{\bar{F}x \to \bar{F}y} \qquad \bar{F}(x^\smile) \leftrightarrow (\bar{F}x)^\smile$$

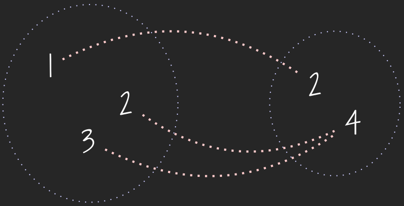
lemma

F extends as a monotone functor $\mathrm{Rel} \to \mathrm{Rel}$ iff it preserves weak pullbacks.

Example: $\bar{\mathcal{P}}(x)$

$$(A,B) \in \bar{\mathcal{P}}(x) \Leftrightarrow \begin{cases} \forall a \in A, \exists b \in B : & (a,b) \in x \\ \forall b \in B, \exists a \in A : & (a,b) \in x. \end{cases}$$

$\{1,2,3\} \;\; \bar{\mathcal{P}}(<) \;\; \{2,4\}.$
$\{0\} \;\; \bar{\mathcal{P}}(<) \;\; \{2,4\}.$
$(\{1,2,3\},\{0,4\}) \notin \bar{\mathcal{P}}(<).$



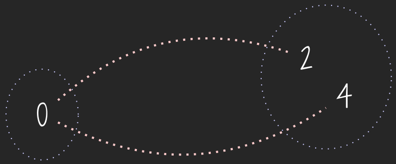
Example: $\bar{\mathcal{P}}(x)$

$$(A,B) \in \bar{\mathcal{P}}(x) \Leftrightarrow \begin{cases} \forall a \in A, \exists b \in B : \ (a,b) \in x \\ \forall b \in B, \exists a \in A : \ (a,b) \in x. \end{cases}$$

$\{1,2,3\} \ \bar{\mathcal{P}}(<) \ \{2,4\}.$

$\{0\} \ \bar{\mathcal{P}}(<) \ \{2,4\}.$

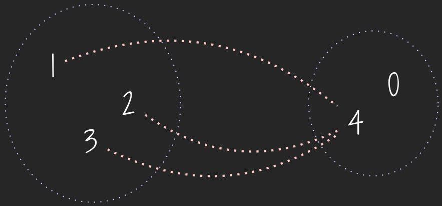
$(\{1,2,3\},\{0,4\}) \notin \bar{\mathcal{P}}(<).$



Example: $\bar{\mathcal{P}}(x)$

$$(A,B) \in \bar{\mathcal{P}}(x) \Leftrightarrow \begin{cases} \forall a \in A, \exists b \in B : & (a,b) \in x \\ \forall b \in B, \exists a \in A : & (a,b) \in x. \end{cases}$$

$\{1,2,3\} \ \bar{\mathcal{P}}(<) \ \{2,4\}.$
$\{0\} \ \bar{\mathcal{P}}(<) \ \{2,4\}.$
$(\{1,2,3\},\{0,4\}) \notin \bar{\mathcal{P}}(<).$



Example: $\bar{T}_\Sigma(X)$

Let Σ be some signature with finite arities.

$$e \in T_\Sigma(A) ::= \mathtt{var}\,(a) \;\mid\; \mathtt{f}(e_1,\dots,e_n) \qquad \text{with } a \in A,\ \mathtt{f}^{(n)} \in \Sigma.$$

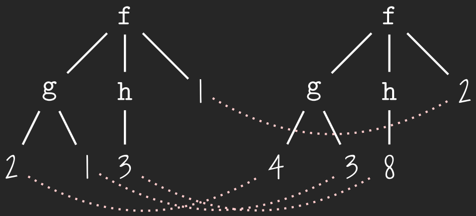
$$\frac{(a,b) \in X}{(\mathtt{var}\,(a)\,,\mathtt{var}\,(b)) \in \bar{T}_\Sigma(X)} \qquad \frac{\forall i (e_i, f_i) \in \bar{T}_\Sigma(X)}{(\mathtt{f}(e_1,\dots,e_n), \mathtt{f}(f_1,\dots,f_n)) \in \bar{T}_\Sigma(X)}$$

Example: $\bar{T}_\Sigma(X)$

Let Σ be some signature with finite arities.

$$e \in T_\Sigma(A) ::= \mathtt{var}\,(a) \;\mid\; \mathtt{f}(e_1, \ldots, e_n) \qquad \text{with } a \in A,\ \mathtt{f}^{(n)} \in \Sigma.$$

Example: $\Sigma = \left\{\mathtt{f}^{(3)}, \mathtt{g}^{(2)}, \mathtt{h}^{(1)}\right\}$ and $A = \mathbb{N}$.



$$\mathtt{f}(\mathtt{g}(2,1), h(3), 1) \;\; \bar{T}_\Sigma(<) \;\; \mathtt{f}(\mathtt{g}(4,3), h(8), 2)$$

Natural relations

☞ $\rho : F \Longrightarrow G$ "natural relation"

$$\rho \,;\, Gf_* \;\rightarrow\; Ff_* \,;\, \rho$$

Natural relations

☞ $\rho : F \Longrightarrow\!\!\!\!\!\!\ominus\; G$ "natural relation"

$$\rho \,;\, Gf_* \;\rightharpoonup\; Ff_* \,;\, \rho$$

☞ $\rho : F_* \Rightarrow G_*$ "left-linear relation"

$$\rho \,;\, Gf_* \;\leftrightarrow\; Ff_* \,;\, \rho$$

$$\Leftrightarrow Fx \,;\, \rho \;\rightharpoonup\; \rho \,;\, Gx$$

Natural relations

☞ $\rho : F \Longrightarrow\!\!\!\!\!\!\!\circ\;\; G$ <u>"natural relation"</u>

$$\rho \,;\, Gf_* \;\rightharpoonup\; Ff_* \,;\, \rho$$

☞ $\rho : F_* \Rightarrow G_*$ <u>"left-linear relation"</u>

$$\rho \,;\, Gf_* \;\leftrightarrow\; Ff_* \,;\, \rho$$

$$\Leftrightarrow Fx \,;\, \rho \;\rightharpoonup\; \rho \,;\, Gx$$

☞ $\rho : \bar{F} \Rightarrow \bar{G}$ <u>"linear relation"</u>

$$\rho \,;\, \bar{G}x \;\leftrightarrow\; \bar{F}x \,;\, \rho$$

Higher order representations

A HO-representation is given by:

- ☞ a pair of Set-endofunctors $\mathcal{E}$ and $\mathcal{T}$;
- ☞ a right-linear relation $\models : \mathcal{T} \Longleftrightarrow \mathcal{E}$; $\quad \forall x,\ \models ; \bar{\mathcal{E}}x \rightarrow \bar{\mathcal{T}}x ; \models$
- ☞ a natural relation $\leqq : \mathcal{E} \Longleftrightarrow \mathcal{E}$; $\quad \forall f,\ \mathcal{E}f^* ; \leqq \rightarrow \leqq ; \mathcal{E}f^*$

such that for every set A, $\mathcal{R}(A) := \langle \mathcal{T}(A), \mathcal{E}(A), \models_A, \leqq_A \rangle$ is a representation.

Higher order representations

Equivalent to the interpretation function being a natural transformation $\mathcal{E} \Rightarrow \mathcal{PT}$.

A HO-representation is given by:

- a pair of Set-endofunctors $\mathcal{E}$ and $\mathcal{T}$;
- a right-linear relation $\models : \mathcal{T} \Rrightarrow \mathcal{E}$; $\qquad \forall x,\ \models ; \bar{\mathcal{E}}x \to \bar{\mathcal{T}}x ; \models$
- a natural relation $\leqq : \mathcal{E} \Rrightarrow \mathcal{E}$; $\qquad \forall f,\ \mathcal{E}f^* ; \leqq \ \to\ \leqq ; \mathcal{E}f^*$

such that for every set A, $\mathcal{R}(A) := \langle \mathcal{T}(A), \mathcal{E}(A), \models_A, \leqq_A \rangle$ is a representation.

Higher order representations

Equivalent to the interpretation function being a natural transformation $\mathcal{E} \Rightarrow \mathcal{P}\mathcal{T}$.

A HO-representation is given by:

- a pair of Set-endofunctors $\mathcal{E}$ and $\mathcal{T}$;
- a right-linear relation $\vDash : \mathcal{T} \Longrightarrow \mathcal{E}$; $\qquad \forall x, \vDash ; \bar{\mathcal{E}}x \subseteq \bar{\mathcal{T}}x ; \vDash$
- a natural relation $\leqq : \mathcal{E} \Longrightarrow \mathcal{E}$; $\qquad \forall f, \mathcal{E}f^* ; \leqq \subseteq \leqq ; \mathcal{E}f^*$

such that for every set A, $\mathcal{R}(A) := \langle \mathcal{T}(A), \mathcal{E}(A), \vDash_A, \leqq_A \rangle$ is a representation.

$e \leqq e' \Rightarrow \mathcal{E}f(e) \leqq \mathcal{E}f(e')$.

Higher order representations

Equivalent to the interpretation function being a natural transformation $\mathcal{E} \Rightarrow \mathcal{P}\mathcal{T}$.

A HO-representation is given by:

- ☞ a pair of Set-endofunctors $\mathcal{E}$ and $\mathcal{T}$;
- ☞ a right-linear relation $\vDash : \mathcal{T} \Longrightarrow \mathcal{E}$; $\quad \forall x, \vDash ; \bar{\mathcal{E}} x \rightarrow \bar{\mathcal{T}} x ; \vDash$
- ☞ a natural relation $\leqq : \mathcal{E} \Longrightarrow \mathcal{E}$; $\quad \forall f, \mathcal{E} f^* ; \leqq \rightarrow \leqq ; \mathcal{E} f^*$

such that for every set A, $\mathcal{R}(A) := \langle \mathcal{T}(A), \mathcal{E}(A), \vDash_A, \leqq_A \rangle$ is a representation.

$e \leqq e' \Rightarrow \mathcal{E} f(e) \leqq \mathcal{E} f(e')$.

Theorem

HO-representations are exactly the functors $\mathsf{Set} \to \mathsf{Repr}$.

Overlaying representations

Can we use HO-representations to get functors Repr $\to$ Repr?

Overlaying representations

Can we use HO-representations to get functors Repr $\to$ Repr?

Yes! Let's do it!

Overlayed representations

Let $\mathcal{R} = \langle \mathcal{T}, \mathcal{E}, \Lleftarrow, \leqq \rangle$ be an HO-representation.
$\widehat{\mathcal{R}}$ is defined on an object $R = \langle T, E, \models, \leq \rangle$ as

$$\widehat{\mathcal{R}}(R) := \langle \mathcal{T}(T), \mathcal{E}(E), \Vdash, \lessapprox \rangle.$$

with: $\Vdash := \mathcal{T}(\models); \Lleftarrow$ and $\lessapprox := (\mathcal{E} \leq + \leqq)^\star$.

reflexive-transitive closure

The image of a morphism is $\widehat{\mathcal{R}}\langle f, g \rangle := \langle \mathcal{T}f, \mathcal{E}g \rangle$.

Overlayed representations

Let $\mathcal{R} = \langle \mathcal{T}, \mathcal{E}, \sqsubseteq, \leqq \rangle$ be an HO-representation.
$\widehat{\mathcal{R}}$ is defined on an object $R = \langle T, E, \models, \leq \rangle$ as

$$\widehat{\mathcal{R}}(R) := \langle \mathcal{T}(T), \mathcal{E}(E), \Vdash, \lessapprox \rangle.$$

with: $\Vdash := \mathcal{T}(\models) ; \sqsubseteq$ and $\lessapprox := (\mathcal{E} \leq + \leqq)^{\star}$.

reflexive-transitive closure

The image of a morphism is $\widehat{\mathcal{R}}\langle f, g \rangle := \langle \mathcal{T}f, \mathcal{E}g \rangle$.

lemma

$\widehat{\mathcal{R}}$ is a functor $\mathsf{Repr} \to \mathsf{Repr}$.

What about exactness?

☞ In general, $\widehat{\mathcal{R}}$ is not exactness preserving, even if $\mathcal{R}(A)$ is always exact.

What about exactness?

- ☞ In general, $\widehat{\mathcal{R}}$ is not exactness preserving, even if $\mathcal{R}(A)$ is always exact.
- ☞ We can get exactness preserving functors on sub-categories, e.g. <u>ideal representations</u>:

$$\langle T, E, \trianglelefteq, \iota, \leq \rangle$$

with $\trianglelefteq$ a preorder on T and $\iota : E \to T$.

$$\models \;\; \Leftrightarrow \;\; \trianglelefteq ; \iota^*$$

i.e. $I(e) = \{ t \in T \mid t \trianglelefteq \iota(e) \}$.

What about exactness?

- In general, $\widehat{\mathcal{R}}$ is not exactness preserving, even if $\mathcal{R}(A)$ is always exact.
- We can get exactness preserving functors on sub-categories, e.g. <u>ideal representations</u>:

$$\langle T, E, \trianglelefteq, \iota, \leq \rangle$$

with $\trianglelefteq$ a preorder on T and $\iota : E \to T$.

$$\models \;\Leftrightarrow\; \trianglelefteq ; \iota^*$$

i.e. $I(e) = \{ t \in T \mid t \trianglelefteq \iota(e) \}$.

- More work to be done...

Representations

Plan

I. Background

II. Representations

III. Reductions

IV. Higher order representations

V. Conclusion

Representations: features

☛ Bundleing together syntax + semantics

Representations: features

- ☞ Bundleing together syntax + semantics
- ☞ Point-free theory through relation algebra

Representations: features

- Bundleing together syntax + semantics
- Point-free theory through relation algebra
- General proof techniques e.g. reductions

Representations: features

- ☛ Bundleing together syntax + semantics
- ☛ Point-free theory through relation algebra
- ☛ General proof techniques e.g. reductions
- ☛ High-level/clutter-free prespective

Representations: features

- ☛ Bundleing together syntax + semantics
- ☛ Point-free theory through relation algebra
- ☛ General proof techniques e.g. reductions
- ☛ High-level/clutter-free prespective
- ☛ Higher-order extensions

Going forwards

☛ Populating the model

Going forwards

- ☛ Populating the model
- ☛ Taxonomy of representations

Going forwards

- ☛ Populating the model
- ☛ Taxonomy of representations
- ☛ More exactness preserving constructions

Going forwards

- ☞ Populating the model
- ☞ Taxonomy of representations
- ☞ More exactness preserving constructions

- ☞ Within reach: multi Kleene algebra, with n sequential products/stars and m parallel products/stars.

That's all folks!

Merci de votre attention!

https://paul.brunet-zamansky.fr