Compositionality, Adequacy, and Full Abstraction: An Algebraic Viewpoint

Gordon Plotkin

Workshop on Compositionality
Simons Institute, December, 2016

Three questions

Semantics of Natural Language: Hodges 2001

Can every semantics

$$L \xrightarrow{\mathcal{M}} M$$

be made compositional in a canonical way?

Computer science

Does every behaviour

$$L \xrightarrow{\mathcal{B}} B$$

have a fully abstract model?

Algebraic language theory: Shützenberger 1965, Steinby 1992

Does every colouring

$$L \xrightarrow{\mathcal{C}} C$$

have a syntactic monoid/algebra?

Compositionality

Frege The meaning of a linguistic phrase is determined by the meaning of its parts.

Examples:

- Arithmetic
  $\mathcal{M}((1+3)+5)$ is determined by $\mathcal{M}(1+3)$ and $\mathcal{M}(5)$.

- Programming Languages
  $\mathcal{M}(x := 3; \text{if } y = 0 \text{ then } y := 1 \text{ else } z := 2)$ is determined by $\mathcal{M}(x := 3)$ and $\mathcal{M}(\text{if } y = 0 \text{ then } y := 1 \text{ else } z := 2)$.

- Logic
  $\mathcal{M}((0=0) \wedge (1=2))$ is determined by $\mathcal{M}(0=0)$ and $\mathcal{M}(1=2)$.

- Natural Language
  $\mathcal{M}$(Jack likes Jill) is determined by $\mathcal{M}$(Jack) and $\mathcal{M}$(Jill).

A reasonably general model of syntax

- **Denotational semantics** A function $\mathcal{M} : L \longrightarrow M$.
- **Signature** A collection Σ of finitary operation symbols $\mathrm{op} : n$
- **Phrases** The set L_Σ of closed terms:

$$t ::= \mathrm{op}(t_1, \ldots, t_n) \quad (\mathrm{op} : n)$$

- **Compositionality**

  $\mathcal{M}(\mathrm{op}(t_1, \ldots, t_n))$ is determined by $\mathcal{M}(t_1)$, ... , $\mathcal{M}(t_n)$.

- **Substitutivity**

$$\frac{t_1 \sim_{\mathcal{M}} u_1, \ldots, t_n \sim_{\mathcal{M}} u_n}{\mathrm{op}(t_1, \ldots, t_n) \sim_{\mathcal{M}} \mathrm{op}(u_1, \ldots, u_n)}$$

where

$$t \sim_{\mathcal{M}} u \quad \Longleftrightarrow_{\mathrm{def}} \quad \mathcal{M}(t) = \mathcal{M}(u)$$

Algebras and initial Algebras

- Σ-algebras $\mathcal{A}$: a set A and maps $\mathrm{op}_{\mathcal{A}} : A^n \to A$ (for $\mathrm{op} : n$).
- Homomorphisms: maps $h : \mathcal{A} \to \mathcal{B}$ preserving the operations:

$$h(\mathrm{op}_{\mathcal{A}}(a_1, \ldots, a_n)) = \mathrm{op}_{\mathcal{B}}(h(a_1), \ldots, h(a_n))$$

- Initial Algebra $\mathcal{I}$ This is L_Σ with:

$$\mathrm{op}_{\mathcal{I}}(t_1, \ldots, t_n) = \mathrm{op}(t_1, \ldots, t_n)$$

  For any other algebra $\mathcal{B}$ there is a unique homomorphism

$$h_{\mathcal{B}} : \mathcal{I} \to \mathcal{B}$$

  where:

$$h_{\mathcal{B}}(\mathrm{op}_{\mathcal{I}}(t_1, \ldots, t_n)) = \mathrm{op}_{\mathcal{B}}(h_{\mathcal{B}}(t_1), \ldots, h_{\mathcal{B}}(t_n))$$

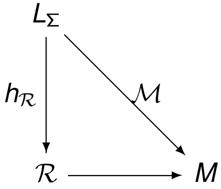
Initial Algebras and Compositionality

- Homomorphism $\implies$ compositional: for any algebra $\mathcal{M}$

  $$h_{\mathcal{M}} : \mathcal{I} \to \mathcal{M}$$

  is compositional, as $h_{\mathcal{M}}(\mathrm{op}(t_1, \ldots, t_n))$ is determined by $h_{\mathcal{M}}(t_1)$, ... , $h_{\mathcal{M}}(t_n)$.

- Compositional $\implies$ homomorphism: any compositional semantics $\mathcal{M} : L_\Sigma \to M$ can be factored as a homomorphism followed by an inclusion:



  where R is $\mathcal{M}(L_\Sigma)$ and

  $$\mathrm{op}_{\mathcal{R}}(\mathcal{M}(t_1), \ldots, \mathcal{M}(t_n)) = \mathcal{M}(\mathrm{op}(t_1, \ldots, t_n))$$

Congruences

- Congruence on $\mathcal{A}$: equivalence relation $\sim$ on A respecting the operations:

$$a_1 \sim a'_1, \ldots, a_n \sim a'_n \implies \mathrm{op}_{\mathcal{A}}(a_1, \ldots, a_n) \sim \mathrm{op}_{\mathcal{A}}(a'_1, \ldots, a'_n)$$

- There is an algebra $\mathcal{A}/{\sim}$ on the set of $\sim$-equivalence classes:

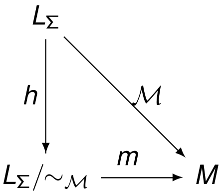
$$\mathrm{op}_{\mathcal{A}/\sim}([a_1] \ldots, [a_n]) = [\mathrm{op}_{\mathcal{A}}(a_1, \ldots, a_n)]$$

- and an evident homomorphism:

$$h : \mathcal{A} \to \mathcal{A}/{\sim}$$

Substitutivity and congruence

Substitutivity is just that $\sim_{\mathcal{M}}$ is a congruence on the initial algebra, and we get a factorisation



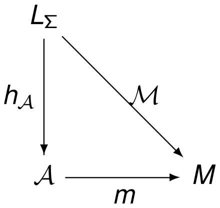
where

$$m([t]) =_{\text{def}} \mathcal{M}(t)$$

This is equivalent to the previous factorisation.

Adequate compositional semantics

A homomorphic semantics $h_{\mathcal{A}} : L_\Sigma \to \mathcal{A}$ is adequate for a semantics $\mathcal{M} : L_\Sigma \to M$ if $h_{\mathcal{A}}$ determines $\mathcal{M}$, i.e., if $\mathcal{M}$ factors through $h_{\mathcal{A}}$:



Contexts and full abstraction

A context is a term $C[\;]$ with one or more holes in it, equivalently a term with a single variable.

It defines a unary function $t \mapsto C[t]$ on L_Σ.

More generally, it defines a unary function $C_{\mathcal{A}}$ on any algebra $\mathcal{A}$. Then, for any homomorphism $h : \mathcal{A} \to \mathcal{B}$, and $a \in A$, we have

$$h(C_{\mathcal{A}}(a)) = C_{\mathcal{B}}(h(a))$$

Contextual equivalence for a semantics $\mathcal{M} : L_\Sigma \to M$ is defined on L_Σ by:

$$t \approx_{\mathcal{M}} u \iff \forall C[\;].C[t] \sim_{\mathcal{M}} C[u]$$

(We can equivalently restrict to contexts with only one hole.)

A semantics $\mathcal{N} : L_\Sigma \to N$ is fully abstract iff

$$t \sim_{\mathcal{N}} u \iff t \approx_{\mathcal{M}} u$$

Adequate semantics and full abstraction

Fact

For any adequate homomorphic semantics $h_{\mathcal{A}}$ adequate for a semantics $\mathcal{M}$ we have:

$$t \sim_{h_{\mathcal{A}}} u \implies t \approx_{\mathcal{M}} u$$

Proof.

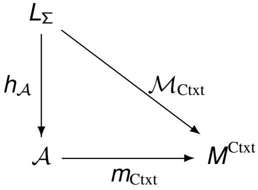
Suppose that $h_{\mathcal{A}}(t) = h_{\mathcal{A}}(u)$. Then we have:

$$h_{\mathcal{A}}(C[t]) = C_{\mathcal{A}}(h_{\mathcal{A}}(t)) = C_{\mathcal{A}}(h_{\mathcal{A}}(u)) = h_{\mathcal{A}}(C[u])$$

and so, by adequacy, $C[t] \sim_{\mathcal{M}} C[u]$. □

Saying it diagramatically

- That the lemma holds.



where,

$$\mathcal{M}_{\mathrm{Ctxt}}(t)(C) =_{\mathrm{def}} \mathcal{M}(C[t])) \qquad m_{\mathrm{Ctxt}}(a)(C) =_{\mathrm{def}} m(C_\mathcal{A}(a))$$

- Full abstraction:

$$\mathcal{M}_{\mathrm{Ctxt}}(t) = \mathcal{M}_{\mathrm{Ctxt}}(u) \implies h_\mathcal{A}(t) = h_\mathcal{A}(u)$$

- Semantical full abstraction: m_{ctxt} is a mono

- Semantical $\implies$ ordinary; converse holds if $h_\mathcal{A}$ surjective.

Algebraic characterisation of $\approx_{\mathcal{M}}$

Fact

$\approx_{\mathcal{M}}$ is the largest congruence on L_Σ, compatible with $\sim_{\mathcal{M}}$.

Proof.

Compatibility Evidently $\approx_{\mathcal{M}} \subseteq \sim_{\mathcal{M}}$.

Congruence Suppose $t \approx t'$, $u \approx u'$ then, for any C and $\mathrm{op} : 2$ we have:

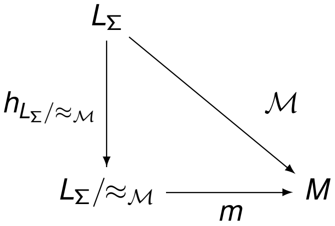
$$\mathcal{M}(C[\mathrm{op}(t, u)]) = \mathcal{M}(C[\mathrm{op}(t, u')]) = \mathcal{M}(C[\mathrm{op}(t', u')])$$

Largest Let $\cong$ be any such congruence. Then for any t, u and C we have:

$$t \cong u \implies C[t] \cong C[u] \implies C[t] \sim_{\mathcal{M}} C[u]$$

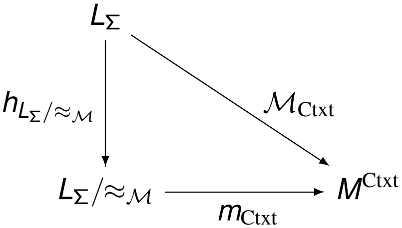
Wilfrid Hodges construction: $L_\Sigma / \approx_{\mathcal{M}}$

Adequacy



where $m([t]) =_{\text{def}} \mathcal{M}(t)$

Semantic full abstraction



The syntactic algebra

We can generalise from L_Σ to any algebra $\mathcal{A}$ and any "semantics" (aka colouring) $\mathcal{M} : \mathcal{A} \to M$.

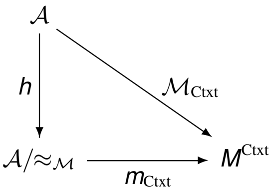
Set:

$$a \sim_{\mathcal{M}} b \iff \mathcal{M}(a) = \mathcal{M}(b)$$

$$a \approx_{\mathcal{M}} b \iff \forall C. C_{\mathcal{A}}(a) \sim_{\mathcal{M}} C_{\mathcal{A}}(b)$$

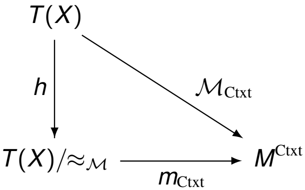
Then $\approx_{\mathcal{M}}$ is the largest congruence on $\mathcal{A}$, compatible with $\sim_{\mathcal{M}}$.

We have (generalised) full abstraction:



where $m([a]) = \mathcal{M}(a)$.

Finitary monads (= free algebras for equational theories)



Applications:

- Syntax with built-in equations, e.g. that program composition is associative.
- Monoids: when have the Shutzenberger syntactic monoid recognising language (= boolean colouring).

  In this case the one-hole contexts have the form $u[\ ]v$ where u, v are words with letters from X.

Multisorted algebra

Signatures Σ Set S of sorts
Operations $\mathrm{op} : s_1, \ldots, s_k \to s'$

Contexts $C : s \to s'$

Algebras $\mathcal{A}$ Carriers: A_s
Operations: $op_{\mathcal{A}} : A_{s_1} \times \cdots \times A_{s_k} \to A_{s'}$

Congruences Suitable families of equivalence relations
$\sim_s \;\subseteq\; A_s^2$

Colourings $c_s : A_s \to C_s$

Congruence for syntactic algebra

$$a \approx_{c,s} b \iff \forall s'. \forall C : s \to s'. C_{\mathcal{A}}(a) \sim_{s'} C_{\mathcal{A}}(a)$$

Syntax with binders

Some examples:

Lambda calculus

$$\lambda x.\, M \qquad \mathrm{app}(M, N)$$

Integration

$$\int_a^b f(x)dx$$

Quantifiers

$$\forall x.\varphi(x)$$

Binding signatures and terms

Binding signatures Σ

$$\mathrm{op} : b_1, \ldots, b_k \text{ for some } k \text{ and } b_1, \ldots, b_k \text{ in } \mathbb{N}.$$

Examples

- Lambda calculus: $\mathrm{app} : 0, 0 \quad \lambda : 1$
- Integration $\int : 0, 0, 1$
- Quantifiers $\forall : 1$

Binding terms and their free variables

$$x_1, \ldots, x_n \vdash M$$

Example

$$\mathrm{op}((x_{1,1}, \ldots, x_{1,b_1}).\, M_1, \ldots, (x_{k,1}, \ldots, x_{k,b_k}).\, M_k)$$

$$\frac{x_1, \ldots, x_n, x_{i,1}, \ldots, x_{i,b_i} \vdash M_i \quad (i = 1, k)}{x_1, \ldots, x_n \vdash \mathrm{op}((x_{1,1}, \ldots, x_{1,b_1}).\, M_1, \ldots, (x_{k,1}, \ldots, x_{k,b_k}).\, M_k)}$$

Clones

- Families of sets $\mathcal{C}_n$
- Projections and Composition

$$\pi^{\mathcal{C}}_{n,i} \in \mathcal{C}_n \quad (i = 1, n)$$

$$\mathrm{Comp}^{\mathcal{C}}_{n,m} : \mathcal{C}_n \times \mathcal{C}^n_m \to \mathcal{C}_m$$

- Axioms

$$\pi^{\mathcal{C}}_{n,i}(f_1, \ldots, f_n) = f_i \qquad f(\pi^{\mathcal{C}}_{n,1}, \ldots, \pi^{\mathcal{C}}_{n,n}) = f$$

$$f(\mathbf{g})(\mathbf{h}) = f(g_1(\mathbf{h}), \ldots, g_n(\mathbf{h}))$$

Binding algebras (cntnd)

Operations $\mathrm{op} : b_1, \ldots, b_k$

- Maps

$$\mathrm{op}_{\mathcal{C},p} : \mathcal{C}_{b_1+p} \times \cdots \times \mathcal{C}_{b_k+p} \to \mathcal{C}_p$$

  Think of p as the number of parameters.

- Uniformity in parameters

$$\mathrm{op}_{\mathcal{C},p}(f_1, \ldots, f_k)(\mathbf{g}) = \mathrm{op}_{\mathcal{C},q}(f_1(\mathbf{g}), \ldots, f_k(\mathbf{g}))$$

Semantics of binding terms

Semantics

$$\mathcal{C}(x_1, \ldots, x_n \vdash M) \in \mathcal{C}_n$$

or just write

$$\mathcal{C}(M)$$

Example

$$\begin{aligned} & \mathcal{C}(\mathrm{op}((x_{1,1}, \ldots, x_{1,b_1}).\, M_1, \ldots, (x_{k,1}, \ldots, x_{k,b_k}).\, M_k)) \\ = \quad & \mathrm{op}_{\mathcal{C}}(\mathcal{C}(M_1), \ldots, \mathcal{C}(M_k)) \end{aligned}$$

Homomorphisms of binding algebras

Families of maps

$$h_n : \mathcal{C}_n \to \mathcal{D}_n$$

such that:

$$h_n(\pi^{\mathcal{C}}_{n,i}) = \pi^{\mathcal{D}}_{n,i}$$

$$h_m(f(g_1, \ldots, g_n)) = h_n(f)(h_m(g_1), \ldots, h_m(g_n))$$

$$h_p(op_{\mathcal{C},p}(f_1, \ldots, f_n)) = op_{\mathcal{D},p}(h_p(f_1), \ldots, h_p(f_n))$$

Congruences of binding algebras

Families of equivalence relations

$$\sim_n \;\subseteq\; \mathcal{C}_n^2$$

such that

$$\pi_{n,i} \sim_n \pi_{n,i}$$

$$\frac{f \sim_n f', g_1 \sim_m g'_1, \ldots, g_n \sim_m g'_n}{f(g_1, \ldots, g_n) \sim f'(g'_1, \ldots, g'_n)}$$

$$\frac{f_1 \sim_{b_1+p} f'_1, \ldots, f_k \sim_{b_k+p} f'_k}{\mathrm{op}_p(f_1, \ldots, f_k) \sim_p \mathrm{op}_p(f'_1, \ldots, f'_k)}$$

Initial Σ-Algebra

$$L_n =_{\text{def}} \{M \mid \mathrm{FV}(M) \subseteq \{z_1, \ldots, z_n\}\}$$

$$\pi^L_{k,i} =_{\text{def}} z_i$$

$$\mathrm{Comp}_L(M, N_1, \ldots, N_k) =_{\text{def}} M[N_1/z_1, \ldots, N_k/z_k]$$

$$\begin{aligned}\mathrm{op}_{L,p}(M_1, \ldots, M_k) \quad =_{\text{def}} \quad & \mathrm{op}(\\ & ((z_1, \ldots, z_{b_1}).\, M_1)[z_1/z_{b_1+1}, \ldots, z_p/z_{b_1+p}] \\ & , \ldots, \\ & ((z_1, \ldots, z_{b_k}).\, M_k)[z_1/z_{b_k+1}, \ldots, z_p/z_{b_k+p}] \\ &)\end{aligned}$$

Fully abstract models

Contexts Terms C with a hole. For example:

$$C[\ ] = \forall x.\,(\varphi(x) \wedge \forall y.\,[\ ])$$

is a context capturing x, y.

Contextual equivalence

Given an equivalence relation $\sim$ on closed terms, for M, N with free variables $x_1, \ldots, x_m$ set:

$$M \approx N \iff \begin{array}{l} \forall C \text{ capturing } y_1, \ldots, y_n. \\ \forall P_1, \ldots, P_m \text{ with free variables } y_1, \ldots, y_n. \\ \quad C[M[\mathbf{P}/\mathbf{x}]] \sim C[N[\mathbf{P}/\mathbf{x}]] \end{array}$$

Fully abstract model There is a binding algebra $\mathcal{C}$ such that:

$$\mathcal{C}(M) = \mathcal{C}(N) \iff M \approx N$$

Example binding algebra equational theories

Lambda calculus

$$\mathrm{app}(\lambda x.\, f(x), y) = f(x) \qquad (\beta)$$

$$\lambda y.\, \mathrm{app}(x, y) = x \qquad (\eta)$$

Algebraic logic

$$(\forall x.\, f(x)) \wedge f(y) = f(y)$$

$$\forall x.\, (f(x) \wedge y) = \forall x.\, f(x) \wedge y$$

$$\forall x.\, \top = \top$$

Note Both of these use a unary function variable f.

Example contextual equivalence for first-order logic

For sentences φ, ψ set:

$$\varphi \sim \psi \iff (\vdash \varphi \text{ iff } \vdash \psi)$$

Then for formulas φ, ψ with free variables $x_1, \ldots, x_n$ we have:

$$\phi \approx \psi \quad \iff \quad \vdash \forall x_1, \ldots, x_n.\, \varphi \equiv \psi$$

A counterexample: algebras with infinitary operations

Lemma

Let $\mathcal{A}$ be an algebra with two congruences $\approx_1, \approx_2$ such that $\sim$, the least equivalence relation containing their union, is not a congruence. Then $\mathcal{A}$ has no syntactic algebra wrt (the colouring corresponding to) $\sim$.

Proof.

Suppose $\approx$ is a maximal congruence such that $\approx \subseteq \sim$. By maximality,

$$\approx \;\supseteq\; \approx_1 \;\cup \approx_2$$

and so

$$\approx \;\supseteq\; \sim$$

So, as $\approx \subseteq \sim$,

$$\approx = \sim$$

So $\sim$ is a congruence, contrary to the hypothesis. □

Mikołaj Bojańczyk's counterexample

Signature:

- A countably infinitary function symbol f, and
- constants $a_i \ (i \geq 0)$.

Two congruences on the initial algebra:

1. $\approx_1$ is the congruence generated by $a_{2j} \sim a_{2j+1} \ (j \geq 0)$
2. $\approx_2$ is the congruence generated by $a_{2j+1} \sim a_{2j+2} \ (j \geq 0)$

We do not have

$$f(a_0, a_1, \ldots, a_n, \ldots) \sim f(a_0, a_0, \ldots, a_0, \ldots)$$

So $\sim$ is not a congruence as we do have

$$a_i \sim a_0$$

A conjecture

Say that a monad T on the category of sets admits syntactic algebras iff any T-algebra has a syntactic algebra wrt any colouring.

Conjecture T admits syntactic algebras *iff* it is finitary.

What next?

- Let $\mathcal{K}$ be locally finitely presentable as a cartesian closed category. Do all finitary enriched monads admit syntactic algebras?
- What else can we do/relate at a general level? Coalgebra and bisimulation? Predicate transformers? Monadic semantics? Logic of programs?
- What are the interesting connections between the semantics of programming languages and algebraic language theory? For example, duality plays a role in both (via predicate transformer semantics in the former).
- What happens beyond the cartesian case? Quantum programming languages, for example?