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

\[ L \xrightarrow{M} M \]

be made compositional in a canonical way?

Computer science
Does every behaviour

\[ L \xrightarrow{B} B \]

have a fully abstract model?

Algebraic language theory: Shützenberger 1965, Steinby 1992
Does every colouring

\[ L \xrightarrow{C} C \]

have a syntactic monoid/algebra?
Frege The meaning of a linguistic phrase is determined by the meaning of its parts.

Examples:

- **Arithmetic**
  \[ M((1 + 3) + 5) \text{ is determined by } M(1 + 3) \text{ and } M(5). \]

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

- **Logic**
  \[ M((0 = 0) \land (1 = 2)) \text{ is determined by } M(0 = 0) \text{ and } M(1 = 2). \]

- **Natural Language**
  \[ M(\text{Jack likes Jill}) \text{ is determined by } M(\text{Jack}) \text{ and } M(\text{Jill}). \]
A reasonably general model of syntax

- **Denotational semantics**: A function \( \mathcal{M} : L \rightarrow M \).
- **Signature**: A collection \( \Sigma \) of finitary operation symbols \( \text{op} : n \).
- **Phrases**: The set \( L_\Sigma \) of closed terms:
  \[
  t ::= \text{op}(t_1, \ldots, t_n) \quad (\text{op} : n)
  \]

- **Compositionality**: \( \mathcal{M}(\text{op}(t_1, \ldots, t_n)) \) is determined by \( \mathcal{M}(t_1), \ldots, \mathcal{M}(t_n) \).

- **Substitutivity**:
  \[
  \frac{t_1 \sim_\mathcal{M} u_1, \ldots, t_n \sim_\mathcal{M} u_n}{\text{op}(t_1, \ldots, t_n) \sim_\mathcal{M} \text{op}(u_1, \ldots, u_n)}
  \]

where
\[
  t \sim_\mathcal{M} u \iff \text{def} \quad \mathcal{M}(t) = \mathcal{M}(u)
\]
Algebras and initial Algebras

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

$$h(\text{op}_A(a_1, \ldots, a_n)) = \text{op}_B(h(a_1), \ldots, h(a_n))$$

- **Initial Algebra** $\mathcal{I}$: This is $L_\Sigma$ with:

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

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

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

where:

$$h_\mathcal{B}(\text{op}_\mathcal{I}(t_1, \ldots, t_n)) = \text{op}_\mathcal{B}(h_\mathcal{B}(t_1), \ldots, h_\mathcal{B}(t_n))$$
Homomorphism $\Rightarrow$ compositional: for any algebra $\mathcal{M}$

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

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

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

$$L_\Sigma$$

$$\downarrow h_\mathcal{R}$$

$$\mathcal{R}$$

$$\downarrow \mathcal{M}$$

$$M$$

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

$$\text{op}_R(\mathcal{M}(t_1), \ldots, \mathcal{M}(t_n)) = \mathcal{M}(\text{op}(t_1, \ldots, t_n))$$
Congruences

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

  $$a_1 \sim a'_1, \ldots, a_n \sim a'_n \implies \text{op}_A(a_1, \ldots, a_n) \sim \text{op}_A(a'_1, \ldots, a'_n)$$

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

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

- and an evident homomorphism:

  $$h : \mathcal{A} \to \mathcal{A}/\sim$$
Substitutivity is just that $\sim_{\mathcal{M}}$ is a congruence on the initial algebra, and we get a factorisation

$$
\begin{array}{ccc}
L_{\Sigma} & \xrightarrow{m} & M \\
\downarrow h & & \downarrow M \\
L_{\Sigma}/\sim_{\mathcal{M}} & \xrightarrow{m} & M
\end{array}
$$

where

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

This is equivalent to the previous factorisation.
A homomorphic semantics $h_A : L_\Sigma \rightarrow A$ is adequate for a semantics $M : L_\Sigma \rightarrow M$ if $h_A$ determines $M$, i.e., if $M$ factors through $h_A$:

\[
\begin{array}{ccc}
L_\Sigma & \xrightarrow{h_A} & A \\
\downarrow & & \downarrow m \\
A & \xrightarrow{M} & M
\end{array}
\]
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 $L_\Sigma$.

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 $L_\Sigma$ 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 $h_A$ adequate for a semantics $\mathcal{M}$ we have:

$$ t \sim_{h_A} u \implies t \approx_{\mathcal{M}} u $$

**Proof.**

Suppose that $h_A(t) = h_A(u)$. Then we have:

$$ h_A(C[t]) = C_A(h_A(t)) = C_A(h_A(u)) = h_A(C[u]) $$

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

Gordon Plotkin
Compositionality, Adequacy, and Full Abstraction
Saying it diagramatically

That the lemma holds.

\[
\begin{array}{ccc}
L_\Sigma & \xrightarrow{h_A} & M_{\text{Ctx}} \\
\downarrow & & \downarrow \\
A & \xrightarrow{m_{\text{Ctx}}} & M^{\text{Ctx}}
\end{array}
\]

where,

\[
M_{\text{Ctx}}(t)(C) = \text{def} \ M(C[t]) \quad m_{\text{Ctx}}(a)(C) = \text{def} \ m(C_A(a))
\]

Full abstraction:

\[
M_{\text{Ctx}}(t) = M_{\text{Ctx}}(u) \implies h_A(t) = h_A(u)
\]

Semantical full abstraction: \( m_{\text{ctxt}} \) is a mono

Semantical \( \implies \) ordinary; converse holds if \( h_A \) surjective.
Algebraic characterisation of $\cong_M$

Fact

$\cong_M$ is the largest congruence on $L_\Sigma$, compatible with $\sim_M$.

Proof.

Compatibility Evidently $\cong_M \subseteq \sim_M$.

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

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

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

$$t \equiv u \implies C[t] \equiv C[u] \implies C[t] \sim_M C[u]$$
Wilfrid Hodges construction: $L_\Sigma / \approx \mathcal{M}$

Adequacy

\[ L_\Sigma \]
\[ h_{L_\Sigma / \approx \mathcal{M}} \]
\[ L_\Sigma / \approx \mathcal{M} \]
\[ M \]

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

Semantic full abstraction

\[ L_\Sigma \]
\[ h_{L_\Sigma / \approx \mathcal{M}} \]
\[ L_\Sigma / \approx \mathcal{M} \]
\[ M^{\text{Ctx}} \]

where $m_{\text{Ctx}}([t]) = \text{def } \mathcal{M}_{\text{Ctx}}(t)$
The syntactic algebra

We can generalise from $L_\Sigma$ to any algebra $A$ and any "semantics" (aka colouring) $M : A \rightarrow M$.

Set:

$$a \sim_M b \iff M(a) = M(b)$$

$$a \approx_M b \iff \forall C. C_A(a) \sim_M C_A(b)$$

Then $\approx_M$ is the largest congruence on $A$, compatible with $\sim_M$.

We have (generalised) full abstraction:

$$A \xrightarrow{h} M_{\text{Ctx}}$$

$$A/\approx_M \xrightarrow{m_{\text{Ctx}}} M_{\text{Ctx}}$$

where $m([a]) = M(a)$. 
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 $\Sigma$ Set $S$ of sorts
Operations $\text{op} : s_1, \ldots, s_k \rightarrow s'$

Contexts $C : s \rightarrow s'$

Algebras $\mathcal{A}$ Carriers: $A_s$
Operations: $\text{op}_\mathcal{A} : A_{s_1} \times \cdots \times A_{s_k} \rightarrow A_{s'}$

Congruences Suitable families of equivalence relations
$\sim_s \subseteq A^2_s$ 

Colourings $c_s : A_s \rightarrow C_s$

Congruence for syntactic algebra

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

Some examples:

**Lambda calculus**

\[ \lambda x. M \quad \text{app}(M, N) \]

**Integration**

\[ \int_a^b f(x) \, dx \]

**Quantifiers**

\[ \forall x. \varphi(x) \]
Binding signatures and terms

Binding signatures \( \Sigma \)

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

Examples

- Lambda calculus: \( \text{app} : 0, 0 \quad \lambda : 1 \)
- Integration \( \int : 0, 0, 1 \)
- Quantifiers \( \forall : 1 \)

Binding terms and their free variables

\( x_1, \ldots, x_n \vdash M \)

Example

\[
\text{op}((x_{1,1}, \ldots, x_{1,b_1}). M_1, \ldots, (x_{k,1}, \ldots, x_{k,b_k}). M_k)
\]

\[
\quad x_1, \ldots, x_n, x_i, 1, \ldots, x_{i,b_i} \vdash M_i \quad (i = 1, k)
\]

\[
x_1, \ldots, x_n \vdash \text{op}((x_{1,1}, \ldots, x_{1,b_1}). M_1, \ldots, (x_{k,1}, \ldots, x_{k,b_k}). M_k)
\]
Clones

- Families of sets $C_n$
- Projections and Composition

$\pi_{n,i}^C \in C_n \quad (i = 1, n)$

$\text{Comp}_{n,m}^C : C_n \times C_m \rightarrow C_m$

- Axioms

$\pi_{n,i}^C(f_1, \ldots, f_n) = f_i \quad f(\pi_{n,1}^C, \ldots, \pi_{n,n}^C) = f$

$f(g)(h) = f(g_1(h), \ldots, g_n(h))$
Binding algebras (cntnd)

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

- Maps

$$\text{op}_{C,p} : C_{b_1+p} \times \cdots \times C_{b_k+p} \to C_p$$

Think of $p$ as the number of parameters.

- Uniformity in parameters

$$\text{op}_{C,p}(f_1, \ldots, f_k)(g) = \text{op}_{C,q}(f_1(g), \ldots, f_k(g))$$
Semantics of binding terms

Semantics

\[ C(x_1, \ldots, x_n \vdash M) \in C_n \]

or just write

\[ C(M) \]

Example

\[ C(\text{op}((x_{1,1}, \ldots, x_{1,b_1} \cdot M_1, \ldots, (x_{k,1}, \ldots, x_{k,b_k} \cdot M_k))) = \text{op}_C(C(M_1), \ldots, C(M_k)) \]
Families of maps

\[ h_n : C_n \rightarrow D_n \]
such that:

\[ h_n(\pi^C_{n,i}) = \pi^D_{n,i} \]

\[ h_m(f(g_1, \ldots, g_n)) = h_n(f)(h_m(g_1), \ldots, h_m(g_n)) \]

\[ h_p(op_{C,p}(f_1, \ldots, f_n)) = op_{D,p}(h_p(f_1), \ldots, h_p(f_n)) \]
Families of equivalence relations

\[ \sim_n \subseteq C_n^2 \]

such that

\[ \pi_{n,i} \sim_n \pi_{n,i} \]

\[
\begin{array}{c}
\frac{f \sim_n f', g_1 \sim_m g'_1, \ldots, g_n \sim_m g'_n}{f(g_1, \ldots, g_n) \sim f'(g'_1, \ldots, g'_n)} \\
\frac{f_1 \sim_{b_1+p} f'_1, \ldots, f_k \sim_{b_k+p} f'_k}{\operatorname{op}_p(f_1, \ldots, f_k) \sim_p \operatorname{op}_p(f'_1, \ldots, f'_k)}
\end{array}
\]
Initial $\Sigma$-Algebra

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

$\pi_{k,j}^L = \text{def } z_j$

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

$\text{op}_{L,p}(M_1, \ldots, M_k) = \text{def } \text{op}(\ldots \left( ((z_1, \ldots, z_{b_1}) \cdot M_1)[z_1/z_{b_1+1}, \ldots, z_p/z_{b_1+p}] \right) \ldots)$
**Contexts** Terms $C$ with a hole. For example:

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

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:

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

**Fully abstract model** There is a binding algebra $C$ such that:

$$C(M) = C(N) \iff M \approx N$$
Example binding algebra equational theories

Lambda calculus

\[ \text{app}(\lambda x. f(x), y) = f(x) \quad (\beta) \]

\[ \lambda y. \text{app}(x, y) = x \quad (\eta) \]

Algebraic logic

\[ (\forall x. f(x)) \land f(y) = f(y) \]

\[ \forall x. (f(x) \land y) = \forall x. f(x) \land y \]

\[ \forall x. \top = \top \]

**Note** Both of these use a unary function variable $f$. 
Example contextual equivalence for first-order logic

For sentences \( \varphi, \psi \) set:

\[
\varphi \sim \psi \iff (\vdash \varphi \text{ iff } \vdash \psi)
\]

Then for formulas \( \varphi, \psi \) with free variables \( x_1, \ldots, x_n \) we have:

\[
\phi \approx \psi \iff \vdash \forall x_1, \ldots, x_n. \varphi \equiv \psi
\]
**Lemma**

Let $A$ 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 $A$ 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.
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

$$a_i \sim a_0$$
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?