An Equational Logic for PROPs

Gordon Plotkin

Laboratory for the Foundations of Computer Science, School of Informatics, University of Edinburgh

Motivation and programme

- Would like to program with various forms of graphs
- For example, in rule-based modelling in biology: $G \xrightarrow{r} G'$
- Various forms of dags used: place graphs, link graphs, bigraphs, kappa graphs
- Would like general algebraic framework including textual notation.
- Then can consider equational technology: matching, rewriting, etc (modulo/quantitative).
- Begin (here) by seeking an equational logic for dags
- Which means for symmetric monoidal theories (aka PROPS)
Regard well known equivalence

\[ \text{FinEqPres} \cong \text{LawTh} \]

between the categories of finitary equational presentations and Lawvere theories as a justification for equational presentations and equational reasoning, and proceed accordingly in the analogous linear cases.

Warm up with operads and linear equational tree logic.

Then do PROPS and (a) linear equational dag logic.
Symmetric Operads

These are one object symmetric multicategories $P$, so have:

- **Operations** A set $P(n)$ of arity $n$, for each $n \geq 0$
- **Composition functions** For each $n \geq 0$:

$$P(n) \times (P(m_0) \times \ldots \times P(m_{n-1})) \rightarrow P(m_0 + \ldots + m_{n-1})$$

- An identity

$$e \in P(1)$$

- A left action of the symmetric group. For each $n \geq 0$:

$$S_n \times P(n) \rightarrow P(n)$$

all subject to evident axioms.
Morphisms of operads

Morphisms \( P \xrightarrow{F} Q \) are the multicategory morphisms, so consist of families of maps

\[
P(n) \xrightarrow{F_n} Q(n)
\]

which preserve

- **Composition:**

\[
F(f \circ (g_0, \ldots, g_{n-1})) = F(f) \circ (F(g_0), \ldots, F(g_{n-1}))
\]

- **Identity**

\[
F(e) = e
\]

- **Actions**

\[
F(\pi \star f) = \pi \star F(f)
\]
Linear equational tree logic

- **Signature** $\Sigma$: operation symbols $\text{op} : n$ as usual.
- **Linear Terms** $t := x \mid \text{op}(t_1, \ldots, t_n)$: as usual, but restricted so that no variable appears twice.
- **Linear Equations** $t = u$ as usual, but with the same variables occurring on both sides.
- **Presentations** $\langle \Sigma, Ax \rangle$ where $Ax$ is a finite set of equations (and we may just write $Ax$).
- **Proof** $\vdash_{Ax} t = u$ To come next.
Equality is an equivalence relation

\[
\begin{align*}
    t &= t \\
    t &= u & u &= v \\
    t &= u & u &= t
\end{align*}
\]

Congruence

\[
\begin{align*}
    t_i &= u_i & (i = 0, n - 1) \\
    f(t_0, \ldots, t_{n-1}) &= f(u_0, \ldots, u_{n-1})
\end{align*}
\]

provided the terms in the conclusion are linear.

Substitution

\[
\begin{align*}
    t &= u \\
    t[v_0/y_0, \ldots, v_{n-1}/y_{n-1}] &= u[v_0/y_0, \ldots, v_{n-1}/y_{n-1}]
\end{align*}
\]

provided the terms in the conclusion are linear.

Remark (Cartesian) equational logic is conservative over linear equational tree logic.
From linear presentations to operads

Notation

\[[t]_{Ax} = \text{def } \{u | \vdash_{Ax} t = u\}\]

and we may omit the suffix.

The operad \(\mathcal{O}(Ax)\) has:

- **n–ary Operations** \([t]\) where \(V(t) = \{z_0, \ldots, z_{n-1}\}\).
- **Composition**

\[[t] \circ ([t_0], \ldots, [t_{n-1}]) = [t[t_0 \sigma_0/z_0, \ldots, t_{n-1} \sigma_{n-1}/z_{n-1}]]\]

where the substitution \(\sigma_i\) shifts \(z_j\) to \(z_{j+o}\) where \(o = \sum_{k<i} m_k\).

- **Identity** \(e = [z_0]\)
- **Action** \(\pi \ast [t] = [t[z_{\pi 0}/z_0, \ldots, z_{\pi (n-1)}/z_{n-1}]]\)
Translations between presentations

- A signature translation $\tau: \Sigma \rightarrow \Sigma'$ is an assignment
  
  $\text{op} \in \Sigma \mapsto t_{\text{op}} \in \Sigma'$-linear terms

  where, for $\text{op}: n$ in $\Sigma$, $V(t_{\text{op}}) = \{z_0, \ldots, z_{n-1}\}$.

- Translating $\Sigma$-terms to $\Sigma'$-terms:

  $x^\tau = x$

  $\text{op}(t_0, \ldots, t_{n-1})^\tau = \text{op}_{\tau}[t_0^\tau/z_0, \ldots, t_{n-1}^\tau/z_{n-1}]$

- A presentation translation $\langle \Sigma, Ax \rangle^\tau \rightarrow \langle \Sigma', Ax' \rangle$ is a signature translation $\tau: \Sigma \rightarrow \Sigma'$ such that:

  $t = u \in Ax \Rightarrow \vdash_{Ax'} t^\tau = u^\tau$
The category of linear presentations

- **Morphisms**

\[ \text{Ax} \xrightarrow{[\tau]} \text{Ax} \]

where \( \tau \) is a translation from \( \text{Ax} \) to \( \text{Ax}' \) and \( [\tau](\text{op}) =_{\text{def}} [\tau(\text{op})] \)

- **Composition**

\[ \text{Ax} \xrightarrow{[\tau]} \text{Ax}' \xrightarrow{[\tau']} \text{Ax}'' = \text{Ax} \xrightarrow{[\tau'']} \text{Ax}'' \]

where \( \tau''(\text{op}) = \tau(\text{op})^{\tau'} \)

- **Identity**

\[ \text{op} \leftrightarrow [\text{op}(z_0, \ldots, z_{n-1})] \]

where \( \text{op}: n \).
\( \mathcal{O} \) as a functor

The action on operations is defined by:

\[
\mathcal{O}([\tau])([t]) = [t^\tau]
\]

Fact (Szawiel, Zawadowski)

\( \mathcal{O} \) is an equivalence of categories

Remark \( \mathcal{O}(\Sigma, \text{Ax}) \) is the free operad over \( \langle \Sigma, \text{Ax} \rangle \), i.e., with semantics (i.e., assigned morphisms) of \( \text{Ax} \) in \( \mathcal{O}(\Sigma, \text{Ax}) \) satisfying the equations.

(Such semantics for \( \text{Ax} \) in an operad \( \mathcal{P} \) correspond to translations from \( \text{Ax} \) to the presentation corresponding to \( \mathcal{P} \).)
Fragments of the proof that $\mathcal{O}$ is an equivalence

- **That $\mathcal{O}$ is full.** Given $\mathcal{O}(Ax) \xrightarrow{F} \mathcal{O}(Ax')$ define $\mathcal{O}(Ax) \xrightarrow{[\tau]} \mathcal{O}(Ax')$ so that $[\tau](\text{op}) = F([\text{op}(z_0, \ldots, z_{n-1})])$

- **That $\mathcal{O}$ is essentially onto.** Given $\mathcal{P}$ construct $\langle \Sigma, Ax \rangle$ so that, for each $n$, every element of $\mathcal{P}(n)$ is an operation symbol of arity $n$, and take $Ax$ to be the diagram of $\mathcal{P}$, meaning that there are equations:
  - expressing composition,
  - stating that $e$ is the identity, and
  - expressing the action of the symmetric group.
These are structures:

$$\mathcal{B}^{\text{op}} \xrightarrow{\ell} \mathcal{L}$$

where

- $\mathcal{B}$ is the category of all natural numbers and bijections over them,
- $\mathcal{L}$ is a small symmetric monoidal category, and
- $\ell$ is a strict symmetric monoidal identity-on-objects functor.
Maps of PROPs

A map from $\mathbb{B}^{\text{op}} \xrightarrow{\ell} \mathcal{L}$ to $\mathbb{B}^{\text{op}} \xrightarrow{\ell'} \mathcal{L}'$, is a strict monoidal functor

$$\mathcal{F}: \mathcal{L} \to \mathcal{L}'$$

such that the following diagram commutes:

It is necessarily the identity on objects and symmetric.
Symmetric monoidal equational logic: Example term
Symmetric monoidal equational logic: Example term
Symmetric monoidal equational logic: Example term

\{ A(x_0, x_1; y_0, y_1), B(x_2; y_2, z_1), D(y_0;), C(y_1, y_2; z_0) \}
Symmetric monoidal equational logic: Example term

\{A(x_0, x_1; y_0, y_1), B(x_2; y_2, z_1), D(y_0; ), C(y_1, y_2; z_0)\}
Symmetric monoidal equational logic: Example term

\{ A(x_0, x_1; y_0, y_1), B(x_2; y_2, z_1), D(y_0; ), C(y_1, y_2; z_0) \}
Symmetric monoidal equational logic: Example term

\[ A^{y_0, y_1}_{x_0, x_1} B^{y_2}_{x_2} D^{y_0}_{z_0} C^{y_1, y_2}_{z_0} \]
Symmetric monoidal equational logic

- **Signature** $\Sigma$ of operation symbols $\text{op}$ with arities and co-arities: $\text{op}: m \to n$

- **Atomic terms** These are either
  - **Wires**
    $$ a = x \leftrightarrow y $$
    when:
    $$ \text{IV}(a) = \text{def} \{ x \} \quad \text{OV}(a) = \text{def} \{ y \} $$
  - **Boxes**
    $$ a = \text{op}(x_0, \ldots, x_{m-1}; y_0, \ldots, y_{n-1}) $$
    for $\text{op}: m \to n$, where no two of $x_0, \ldots, x_{m-1}, y_0, \ldots, y_{n-1}$ are the same, when:
    $$ \text{IV}(a) = \text{def} \{ x_0, \ldots, x_{m-1} \} \quad \text{OV}(a) = \text{def} \{ y_0, \ldots, y_{n-1} \} $$
Terms are acyclic multisets of atomic terms:

\[ t = \text{def} \left\{ a_0, \ldots, a_{n-1} \right\} : \text{IV}(t) \rightarrow \text{OV}(t) \]

where no two atomic terms \( a_i \) have a common input or output variable, and when:

\[
\text{IV}(t) = \text{def} \bigcup_i \text{IV}(a_i) \setminus \bigcup_i \text{OV}(a_i) \quad \text{OV}(t) = \text{def} \bigcup_i \text{OV}(a_i) \setminus \bigcup_i \text{IV}(a_i)
\]

The term \( t \) is said to be acyclic if this graph on its variables is:

\[
\left\{ \langle x, y \rangle \mid \exists i. x \in \text{IV}(a_i) \land y \in \text{OV}(a_i) \right\}
\]
More on terms

- **Free variables**

\[ \text{FV}(t) = \text{def } \text{IV}(t) \cup \text{OV}(t) \]

- **Bound Variables**

\[ \text{BV}(t) = \text{def } \bigcup_i \text{IV}(a_i) \cap \bigcup_i \text{OV}(a_i) \]

We identify terms up to \( \alpha \)-equivalence; acyclicity is invariant under \( \alpha \)-equivalence.

- **Substitution**

\[ t[y_1/x_1, \ldots, y_n/x_n] \]

where the \( y_i \) are all different, the \( x_i \) are all different, \( x_i \in \text{FV}(t) \) for all \( i \), and \( y_i \notin \text{FV}(t) \backslash \{x_j \mid j = 1, n\} \).
Equational reasoning

Equations

\[ t = u \]

provided \( \text{IV}(t) = \text{IV}(u) \) and \( \text{OV}(t) = \text{OV}(u) \).

Rules Those of an equivalence relation, plus:

- **Congruence**
  \[
  \frac{t = u}{t, a = u, a}
  \]
  provided \( t, a \) and \( u, a \) are terms (and omitting multiset brackets).

- **Rewiring**
  \[
  t, x \mapsto y = t[y/x] \quad (x \in \text{OV}(t), y \notin \text{FV}(t) \setminus \{x\})
  \]
  \[
  t, x \mapsto y = t[x/y] \quad (y \in \text{IV}(t), x \notin \text{FV}(t) \setminus \{y\})
  \]

Plotkin

An Equational Logic for PROPs
Remarks on logic

- The following **Substitution Rule** is derived:

\[
t = u
\]

\[
t[y_1/x_1, \ldots, y_n/x_n] = u[y_1/x_1, \ldots, y_n/x_n]
\]

- The logic is conservative over the linear tree equational logic.
- The normal forms are those terms in which all variables of wires are either input or output variables.
- The normal forms are in a 1-1 correspondence with isomorphism classes of dags, which correspondence forms an isomorphism of the two presentations of the absolutely free PROP over \( \Sigma \).
This is $P(\Sigma, Ax) = B^{\text{op}} \xrightarrow{\ell} L$. The category $L$ has:

- **Morphisms** These are equivalence classes of terms:

  $$[t] : m \to n$$

  where $\text{IV}(t) = \{z_0, \ldots, z_{m-1}\}$ and $\text{OV}(t) = \{z'_0, \ldots, z'_{n-1}\}$. (We assume available two mutually disjoint fixed infinite sequences $z_0, z_1, \ldots$ and $z'_0, z'_1, \ldots$ of distinct variables.)

- **Composition**

  $$l \xrightarrow{[t]} m \xrightarrow{[u]} n = l \xrightarrow{[t[z''_0/z'_0, \ldots, z''_{m-1}/z'_{m-1}], u[z''_0/z_0, \ldots, z''_{m-1}/z_{m-1}]]} n$$

  ($z''_{i'}$ fresh)
and

- **Tensor of morphisms**

\[
(m \xrightarrow{[t]} n) \otimes (m' \xrightarrow{[u]} n') = \overbrace{(m + m')}^{[t,u[z_m/z_0, \ldots, z_{m+m'-1}/z_{m'-1}][z'_n/z'_0, \ldots, z'_{n+n'-1}/z'_{n'-1}]]} \rightarrow (n + n')
\]

The functor \(\ell\) is given by:

\[
\ell(\pi : n \cong n) = [z_\pi(0) \mapsto z'_0, \ldots, z_\pi(n-1) \mapsto z'_{n-1}]
\]

Plotkin

An Equational Logic for PROPs
A signature translation $\Sigma \xrightarrow{\tau} \Sigma'$ is an assignment:

$$\text{op} \in \Sigma \leftrightarrow t_{\text{op}} \in \Sigma'\text{-terms}$$

where $\text{IV}(t_{\text{op}}) = \{z_0, \ldots, z_{m-1}\}$ and $\text{OV}(t_{\text{op}}) = \{z_0', \ldots, z_{n-1}'\}$ if $\text{op} : m \rightarrow n$.

Translating $\Sigma$-terms to $\Sigma'$-terms:

$$(x \leftrightarrow y)^{\tau} = x \leftrightarrow y$$

$$\text{op}(x_0, \ldots, x_{m-1}; y_0, \ldots, y_{n-1})^{\tau} = \text{op}_{\tau}[x_0/z_0, \ldots, x_{m-1}/z_{m-1}, y_0/z_0', \ldots, y_{n-1}/z_{n-1}']$$

$$\{a_0, \ldots, a_{n-1}\}^{\tau} = \{a_0^{\tau}, \ldots, a_{n-1}^{\tau}\}$$

Category of presentations Defined as before, with morphisms $\text{Ax} \xrightarrow{[\tau]} \text{Ax}'$ where $\tau$ is a correct translation from $\text{Ax}$ to $\text{Ax}'$. 
**$\mathcal{P}$ as a functor**

The action on operations is defined by:

$$\mathcal{P}([\tau])([t]) = [t^\tau]$$

**Fact**

$\mathcal{P}$ is an equivalence of categories

**Remark** $\mathcal{P}(\Sigma, Ax)$ is the free PROP over $\langle \Sigma, Ax \rangle$, i.e., with assigned morphisms satisfying the equations.
An example: Milner’s Bigraphs

A contextual bigraph $H : \langle 3, \{x, x'\} \rangle \to \langle 2, \emptyset \rangle$

**its place graph**

**its link graph**
Some structure in place graphs

- A commutative monoid
  
  ![Diagram of a commutative monoid](image)

  \[1 : 0 \rightarrow 1\]

- with unary functions
  
  ![Diagram of a unary function](image)

  \[K : 1 \rightarrow 1\]
Equational theory for place graphs

- **Signature**
  - \(+ : 2 \rightarrow 1, \ 0 : 0 \rightarrow 1\)
  - \(K : 1 \rightarrow 1 (K \in \mathcal{K})\)

- **Axioms**

\[
\begin{align*}
+ (x, y; u), + (u, z; v) &= + (y, z; u), + (x, u; v) \\
+ (x, y; u) &= + (y, x; u) \\
0(; u), + (u, x; y) &= x \mapsto y = 0(; u), + (x, u; y)
\end{align*}
\]

**Note** We are omitting the multiset brackets.

**Fact**

*Milner’s place graphs form the free PROP generated by these axioms.*
Example abbreviatory conventions

- **Two conventions** For unary \( \text{op}: n \to 1 \) (eg, +, 0),

  \[
  \text{op}'(\ldots, \text{op}(\ldots), \ldots; \ldots) \equiv_{\text{def}} \text{op}(\ldots; x), \text{op}'(\ldots, x, \ldots; \ldots)
  \]

  \[
  \text{op}(\ldots)^x \equiv_{\text{def}} \text{op}(\ldots; x)
  \]

- **Examples**

  \[
  \begin{align*}
  +\bigl(+(x, y), z\bigr)^v &= +\bigl(x, +(y, z)\bigr)^v \\
  +(0(), x)^y &= x \mapsto y = x +^y 0()
  \end{align*}
  \]

- **With infix notation:**

  \[
  \begin{align*}
  (x + y) +^v z &= x +^v (y + z) \\
  0() +^y x &= x \mapsto y = x +^y 0()
  \end{align*}
  \]
Some elementary link graphs

\[ \begin{array}{c}
    y \\
    x_1 \ldots x_n
\end{array} \]  

\( y/X : X \rightarrow \{y\} \)  
Elementary Substitution

\( /x : x \rightarrow \epsilon \)  
Elementary Closure

\( K_{x_1,\ldots,x_n} : \epsilon \rightarrow X \)  
Atom
Equational theory for link graphs

- **Signature**
  - $\| : 2 \to 1$, $\text{NIL}: 0 \to 1$, $\text{NIL}^{-1}: 1 \to 0$
  - $K: 0 \to k$ ($K:k$).

- **Axioms**

\[
\begin{align*}
(x \| y) \| ^v z &= x \| ^v (y \| z) \\
x \| ^v y &= y \| ^v x \\
\text{NIL}() \| ^v x &= x \leftrightarrow y = x \| ^v \text{NIL}() \\
\text{NIL}^{-1}(\text{NIL}(); ) &=
\end{align*}
\]

- **Fact**

Milner’s link graphs form the free PROP generated by these axioms.
Further work

- Connect up to equational logic based on (a small variation of a fragment of) Bierman et al term calculus for intuitionistic linear logic:

\[
\frac{\Gamma \vdash M : m \quad \Delta \vdash N : n,}{\Gamma, \Delta \vdash M \otimes N : m + n}
\]

\[
\frac{\Gamma, \Delta \vdash M : m + n \quad \Delta, x : m, y : n \vdash N : k}{\Gamma, \Delta \vdash \text{let } x : m, y : n \text{ be } M \text{ in } N}
\]

- **Dynamics** Relate to term-rewriting/graph-rewriting. A rule is now just an oriented equation \( t \Rightarrow t' \), as usual, and its application takes on a simple form:

\[
\frac{t \Rightarrow t'}{t, u \Rightarrow t', u}
\]

(cf Krivine, Milner, Troina).

- Drop acyclicity and go on to traced PROPs/general graphs/abstract tensor expressions.