Algebraic Representations of Graphs

Jack Liell-Cock

Department of Computer Science
University of Oxford

UNIVERSITY OF OXFORD

TT2023

Outline

Outline

Algebras & Datatypes

Construction primitives for injection and composition

```
data List a =
      Singleton a
    | Empty
    | List a ++ List a
```

Equational laws for equivalent constructions

- `(a ++ b) ++ c = a ++ (b ++ c)`
- `a ++ Empty = a`
- `Empty ++ a = a`

Algebras & Datatypes

- Computations become recursive functions (which preserve the equational laws)
- For example, we can sum a list of integers because $(\mathbb{Z}, 0, +)$ satisfies the list axioms

```
sum :: List Int -> Int
sum (Singleton n) = n
sum Empty         = 0
sum (a ++ b)      = sum a + sum b
```

- The recursion scheme derives naturally from the construction primitives

Outline

Node Graph Constructors

Node graph algebra introduced by Mokhov (2017)

```
data NodeGraph a =
      Empty
    | Node a
    | Overlay (NodeGraph a) (NodeGraph a)
    | Connect (NodeGraph a) (NodeGraph a)
```

For graph representation $R = (N, E)$ where N is the set of nodes and $E \subseteq N \times N$ are the edges, the operators correspond to

- `Empty`: $\varepsilon = (\emptyset, \emptyset)$
- `Node`: $\dot{x} = (\{x\}, \emptyset)$
- `Overlay`: $(N, E) + (N', E') = (N \cup N', E \cup E')$
- `Connect`: $(N, E) \gg (N', E') = (N \cup N', E \cup E' \cup N \times N')$

Outline

Algebras & Datatypes

Node Graphs

Edge Graphs

Edge Graph Folds

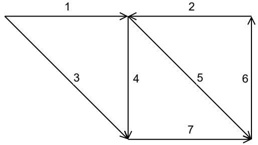
Flow Representation

Definition

A flow representation for a set of edges E is a subset $\gamma \subseteq \mathbb{P}(E) \times \mathbb{P}(E)$ such that

1) $\bigcup_{x\in\gamma} \pi_1 x = E$ and $\bigcup_{x\in\gamma} \pi_2 x = E$
2) $\forall x \neq y \in \gamma,\ \pi_1 x \cap \pi_1 y = \emptyset$ and $\pi_2 x \cap \pi_2 y = \emptyset$
3) $(\emptyset, \emptyset) \notin \gamma$

where π_i are the projections and $\mathbb{P}$ is the powerset operator. The set of all flow representations is Γ.



$$\{(\emptyset, \{1,3\}), (\{1,2\}, \{4,5\}), (\{6\}, \{2\}), (\{3,4\}, \{7\}), (\{7,5\}, \{6\})\}$$

Equivalence of Representations

Let G be the set of all multigraph representations (N, E, σ, τ) where N is the set of nodes, E is the set of edges and $\sigma, \tau : E \to N$ select the source and target for each edge.

Definition (Node Agnosticism)

$(N, E, \sigma, \tau) \sim (N', E, \sigma', \tau')$ if there exists functions $\phi : N \leftrightarrows N' : \psi$ such that $\phi \circ \sigma = \sigma'$, $\phi \circ \tau = \tau'$, $\psi \circ \sigma' = \sigma$, and $\psi \circ \tau' = \tau$

Theorem

$G/\sim \,\cong \Gamma$

Edge Graph Constructors

A similar story is followed for the edge construction

```
data EdgeGraph a =
      Empty
    | Edge a
    | Overlay (EdgeGraph a) (EdgeGraph a)
    | Into (EdgeGraph a) (EdgeGraph a)
    | Pits (EdgeGraph a) (EdgeGraph a)
    | Tips (EdgeGraph a) (EdgeGraph a)
```

`Edge` and `Empty` are injection points. The remaining operators are higher-order compositions.

Constructor Definitions

Definition (Edge Graph Ordering)

For $\gamma, \delta \in \Gamma$, $\gamma \leq \delta$ if for all $x \in \gamma$, there exists a $y \in \delta$ such that $\pi_1 x \subseteq \pi_1 y$ and $\pi_2 x \subseteq \pi_2 y$.

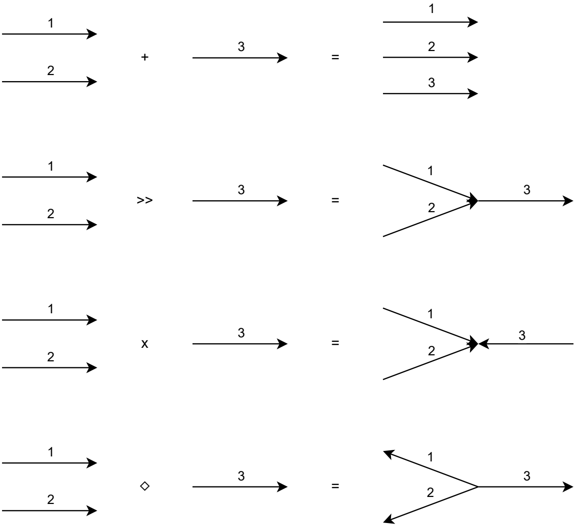
Theorem

$(\Gamma, \leq)$ *is a join-semilattice.*

Then the construction primitives are defined as

- `Empty`: $\varepsilon = \emptyset$
- `Edge`: $\vec{x} = \{(\emptyset, \{x\}), (\{x\}, \emptyset)\}$
- `Overlay`: $a + b = \mathrm{lub}(a, b)$
- `Into`: $a \gg b = \mathrm{lub}(a, b, \dots)$
- `Tips`: $a \times b = \mathrm{lub}(a, b, \dots)$
- `Pits`: $a \diamond b = \mathrm{lub}(a, b, \dots)$

Constructor Definitions



Axioms

- $(\Gamma, \varepsilon, +, \gg)$ is a united monoid.
- $(\Gamma, \varepsilon, +, \diamond)$ is a commutative, united monoid.
- $(\Gamma, \varepsilon, +, \times)$ is a commutative, united monoid.
- Decomposition: for $\square$ and $\blacksquare$ any of $\gg$, $\diamond$ and $\times$

$$a \square (b \blacksquare c) = a \square b + a \square c + b \blacksquare c,$$
$$(a \square b) \blacksquare c = a \square b + a \blacksquare c + b \blacksquare c.$$

- Reflexivity:

$$\vec{x} \diamond \vec{x} = \vec{x},$$
$$\vec{x} \times \vec{x} = \vec{x}.$$

Axioms

- Transitivity: for all $a \neq \varepsilon$,

$$
\begin{aligned}
(a \diamond b) + (a \diamond c) &= a \diamond b \diamond c, \\
(b \gg a) + (a \diamond c) &= b \gg (a \diamond c), \\
(a \gg b) + (a \gg c) &= a \gg (b \diamond c), \\
(a \times b) + (a \gg c) &= (a \times b) \gg c, \\
(b \gg a) + (c \gg a) &= (b \times c) \gg a, \\
(a \times b) + (a \times c) &= a \times b \times c.
\end{aligned}
$$

Theorem

Γ *is the free algebra for the edge graph axioms.*

Outline

Edge Graph Folds

- Γ is the free edge graph algebra, so there exists a unique homomorphism to any other edge graph algebra
- For A an edge graph algebra with operators $e : A$, $f : E \to A$ and $o, i, p, t : A \times A \to A$, we write $(\!|e, f, o, i, p, t|\!)$ for the unique homomorphism $\Gamma \to A$
- Determining a graph algorithm amounts to finding a model that captures the solution properties
- For example,
  - $(\!|\varepsilon, \vec{\sqcup}, +, \gg, \diamond, \times|\!)$ is the identity fold
  - $(\!|\emptyset, \{\cdot\}, \cup, \cup, \cup, \cup|\!)$ reduces a graph to its underlying edges
  - $(\!|\varepsilon, \vec{\sqcup}, +, \ll, \times, \diamond|\!)$ where $a \ll b = b \gg a$ takes the graph transpose

Shortest Paths Fold

```haskell
data End a = Pit a | Tip a deriving (Eq, Ord)
type ShortestPaths a = Map (End a, End a) a

h :: (Ord a, Num a) => EdgeGraph a -> ShortestPaths a
h Empty       = Map.empty
h (Edge x)    = Map.fromList [
        ((Pit x, Pit x), 0),
        ((Pit x, Tip x), x),
        ((Tip x, Tip x), 0)]
h (Overlay x y) = closure (Map.unionWith min (h x) (h y))
h (Into x y) = closure (connect Tip Pit (h x) (h y))
h (Tips x y) = closure (connect Pit Pit (h x) (h y))
h (Pits x y) = closure (connect Tip Tip (h x) (h y))
```

Any questions?

Appendix A: Node Graph Axioms

The set of axioms for node graphs are

- $(R, \varepsilon, +)$ is a commutative, idempotent monoid
  - $a + (b + c) = (a + b) + c$
  - $a + b = b + a$
  - $a + \varepsilon = a$
  - $a + a = a$
- $(R, \gg, \varepsilon)$ is a monoid
  - $a \gg (b \gg c) = (a \gg b) \gg c$
  - $\varepsilon \gg a = a$
  - $a \gg \varepsilon = a$
- $\gg$ distributes over $+$
  - $a \gg (b + c) = a \gg b + a \gg c$
  - $(a + b) \gg c = a \gg c + b \gg c$
- The decomposition axiom
  - $a \gg b \gg c = a \gg b + a \gg c + b \gg c$

Note identity and idempotency of $+$ can be derived from the remaining axioms

Apeendix A: Node Graph Axioms

For refined graph classes, additional axioms can be introduced

- Reflexive graphs

$$\dot{x} \gg \dot{x} = \dot{x}$$

- Undirected graphs

$$a \gg b = b \gg a$$

- Transitive graphs

$$\forall b \neq \varepsilon, a \gg b + b \gg c = a \gg b \gg c$$

- Hypergraphs - replace decomposition axiom with

$$a \gg b \gg c \gg d =$$
$$a \gg b \gg c + a \gg b \gg d + a \gg c \gg d + b \gg c \gg d$$

Appendix B: United Monoids

A common algebraic structure in graph algebras is the united monoid, introduced by Mokhov (2022)

Definition

$(X, \varepsilon, \oplus, \otimes)$ is a united monoid if

- $(X, \varepsilon, \oplus)$ is a commutative monoid
- $(X, \varepsilon, \otimes)$ is a monoid
- $\otimes$ distributes over $\oplus$

Some immediate facts:

- $\oplus$ is idempotent
  - $x \oplus x = (x \otimes \varepsilon) \oplus (x \otimes \varepsilon) = x \otimes (\varepsilon \oplus \varepsilon) = x$
- The containment laws
  - $x \otimes y = (x \otimes y) \oplus x = (x \otimes y) \oplus y = (x \otimes y) \oplus x \oplus y$

Appendix C: Hypergraphs

What if we label the edges numerically?

- Assign the tip as side 1 and the pit as side 2
- Operators $\gg$, $\times$ and $\diamond$ reduce to $\left(\times_i^j\right)_{0<i,j\leq 2}$
- Each $\times_i^j$ is a lub, but also connects each i-th side in the first graph to each j-th side in the second graph
- In particular
  - $\gg \rightsquigarrow \times_1^2$
  - $\times \rightsquigarrow \times_1^1$
  - $\diamond \rightsquigarrow \times_1^2$
  - $\ll \rightsquigarrow \times_2^1$

Appendix C: Hypergraphs

With indexed operators, the edge graph algebra becomes

- $(\Gamma, \varepsilon, +, \times_j^i)$ are united monoids.
- The symmetric axiom, $a \times_i^j b = b \times_j^i a$.
- The decomposition axiom, $a \times_i^j (b \times_k^l c) = a \times_i^j b + a \times_i^j c + b \times_k^l c$.
- The transitive axiom, $\forall a \neq \varepsilon,\ a \times_i^j b + a \times_i^k c = a \times_i^j (b \times_j^k c)$.
- The reflexive axiom, $\vec{x} \times_i^i \vec{x} = \vec{x}$.

Appendix C: Hypergraphs

- For $N = 1$ we subsume Mokhov's symmetric, transitive, reflexive node graphs
- Dropping the corresponding axioms for $N = 1$ recovers Mokhov's wider range of graphs
- For $N > 2$ the notion generalises to simplicial hypergraphs
- Dropping the same axioms for $N > 2$ leads to a globular hypergraph structure

References I

Mokhov, A. (2017). Algebraic graphs with class (functional pearl). In *Proceedings of the 10th ACM SIGPLAN International Symposium on Haskell*, Haskell 2017, pages 2–13, New York, NY, USA. Association for Computing Machinery.

Mokhov, A. (2022). United Monoids: Finding Simplicial Sets and Labelled Algebraic Graphs in Trees. *The Art, Science, and Engineering of Programming*, 6(3):12. arXiv:2202.09230 [cs].