Skew-closed objects, typings of linear lambda terms, and flows on trivalent graphs

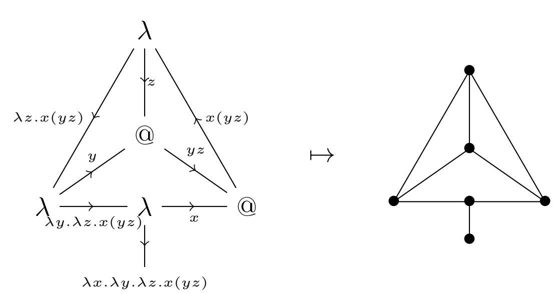
Noam Zeilberger

June 30, 2017

1 Introduction

Lambda calculus turns out to have a variety of links to the study of graphs embedded on surfaces. These connections were originally discovered somewhat accidentally by way of combinatorics (cf. [1, 6, 7]), but seem to hold up on closer inspection, and point to deeper correspondences. They present a new kind of application for string diagrams, and a fresh perspective on linear logic proof-nets.

In [8] I gave a conceptual explanation for the bijection (originally in [1]) between (α-equivalence classes of) linear lambda terms and (isomorphism classes of) rooted trivalent maps[1], taking as a starting point the idea (à la Dana Scott) that linear lambda terms may be interpreted as generalized endomorphisms of a **reflexive object** in a symmetric monoidal closed bicategory (i.e., an object U equipped with an adjunction $@ \dashv \lambda$ to its space of endomorphisms $U \multimap U$). By expressing the signature of a reflexive object in the graphical language of *compact closed* bicategories, one recovers a folklore representation of linear lambda terms as decorated trivalent graphs (sometimes referred to as *lambda-graphs*, and isomorphic to a class of proof-nets). The rooted trivalent map corresponding to a linear lambda term is then obtained simply by forgetting these decorations, e.g.:



That this forgetful transformation is reversible is a bit less obvious, but exploits the fact that there is a recursive decomposition of rooted trivalent maps that exactly mirrors the inductive structure of linear lambda terms (see [8] for details).

Note that because this analysis appeals to the graphical language of compact closed (rather than symmetric monoidal closed) bicategories, it is possible to build string diagrams which do not represent actual lambda terms[2], but only "pseudo-terms". This is a phenomenon familiar from the literature on proof-nets, where it is dealt with through various "correctness criteria" (to which we return below).

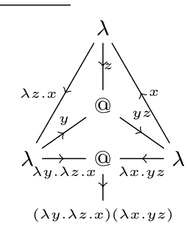
2 Imploid-typings and G-flows

On the one hand, it is worth emphasizing that an "untyped" perspective was important for finding the aforementioned links, and to some extent for explaining them as well (like we just saw). Still, it is natural to wonder about types. Seen through the lens of graph theory, typing may be naturally posed as an *edge-coloring* problem: assign each edge (= subterm) a color (= type) so as to satisfy certain constraints at the vertices (= applications and abstractions). To make this analogy precise, we first introduce a useful algebraic gadget: a preorder $(P, \leq)$ equipped with an operation $\multimap$: $P^{\mathrm{op}} \times P \to P$ and an element $I \in P$, satisfying laws of *composition*, *identity*, *unit*, and *double-negation introduction:*

$$B \multimap C \leq (A \multimap B) \multimap (A \multimap C) \tag{1}$$

[1] A **rooted trivalent map** is a trivalent graph (potentially with free edges) equipped with an embedding into an oriented surface of arbitrary genus (potentially with boundary), and a distinguished univalent vertex marking the root.

[2] For example:

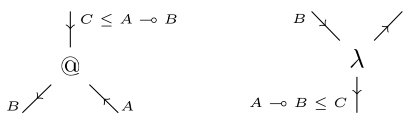


$$I \leq A \multimap A \tag{2}$$

$$I \multimap A \leq A \tag{3}$$

$$A \leq (A \multimap B) \multimap B \tag{4}$$

We refer to such gadgets (lovingly) as (commutative unital) **imploids** – a bit more dryly, if one drops the commutativity axiom (4), they are simply the preorder instantiation of Street's definition of a *skew-closed category* [4]. Given an arbitrary imploid P, we define a P**-typing** of a linear lambda term M as an assignment of elements of P to subterms of M satisfying the following constraints at every application/abstraction:

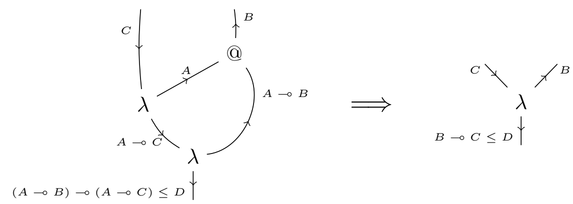


(5)

One motivation for this definition is its close relation to the classical concept of an abelian group-valued *flow* on a graph [5]. Indeed, taking the discrete preorder and setting $A \multimap B \stackrel{\text{def}}{=} B \cdot A^{-1}$, it is easy to check that any abelian group G defines a commutative unital imploid, and that a G-typing of a linear lambda term is the same thing as a G-flow on its underlying trivalent graph. For example, exhibiting a $\mathbb{Z}_2$-typing of a closed term is equivalent to specifying an element of the *cycle space* of its underlying trivalent graph, i.e., a disjoint collection of cycles (these form a vector space over $\mathbb{Z}_2$, whose dimension is called the *circuit rank* of the graph). We can also speak of "proper" P-typings, which are related to "nowhere-zero" G-flows. For example, it is a consequence of the definition that for any linear lambda term M and commutative unital imploid P, a P-typing of M necessarily assigns a type $\geq I$ to all of its *closed* subterms. This translates to a well-known property of nowhere-zero G-flows (cf. [5, 2]), since closed subterms correspond to bridges of the underlying graph [8].

3 Skew-closed objects

Formally, the conditions in (5) may be seen as defining *posetal distributors* $@ : P^{\text{op}} \times P \times P^{\text{op}} \to 2$ and $\lambda : P^{\text{op}} \times P \times P \to 2$, and these are exactly the adjoint pair of distributors associated to the operation $\multimap: P^{\text{op}} \times P \to P$. Observe that here the adjunction $\lambda \dashv @$ goes in the opposite direction of that for a reflexive object, i.e., the unit and the counit correspond to β-*expansion* and η-*reduction.* If we continue down this line of reasoning we arrive at a more abstract definition of "imploid" as a **skew-closed object** in a compact closed bicategory, which can be presented in purely diagrammatic terms using the standard graphical language. For example, the composition law (1) is equivalent to the following "triangle move" (annotated to make the correspondence clearer):



Skew-closed objects appear to provide another useful diagrammatic perspective on linear lambda calculus, complementary to that of reflexive objects. Indeed, this perspective is closely related to that of *combinatory logic:* in collaboration with Jason Reed, we have shown how to use these moves to derive analogues of classical combinatory completeness theorems for several natural fragments of linear lambda calculus [3]. Such completeness theorems can also be viewed as a sort of topological "correctness criterion", in the sense that an arbitrary string diagram represents a term in (some fragment of) lambda calculus just in case it can be reduced to a trivial diagram via a sequence of (some restricted set of) moves.

References

[1] O. Bodini, D. Gardy, and A. Jacquot. Asymptotics and random sampling for BCI and BCK lambda terms. *Theoretical Comp. Sci.*, 502:227–238, 2013.

[2] F. Jaeger. Flows and Generalized Coloring Theorems in Graphs. *J. Comb. Theory*, Series B 26:205–216, 1979.

[3] J. Reed and N. Zeilberger. Some topological combinatory completeness theorems. In preparation, 2017.

[4] R. Street. Skew-closed categories. *J. Pure and Appl. Alg.* 217(6):973–988, 2013.

[5] W. T. Tutte. A contribution to the theory of chromatic polynomials. *Canad. J. Math.* 6:80–91, 1954.

[6] N. Zeilberger and A. Giorgetti. A correspondence between rooted planar maps and normal planar lambda terms. *Log. Meth. in Comp. Sci.*, 11(3:22):1–39, 2015.

[7] N. Zeilberger. Counting isomorphism classes of β-normal linear lambda terms. arXiv:1509.07596

[8] N. Zeilberger. Linear lambda terms as invariants of rooted trivalent maps. *J. Func. Prog.*, 26:e21, 2016.