Balanced polymorphism and linear lambda calculus

Noam Zeilberger (MSR-Inria Joint Centre)

Abstract

A "pearl theorem" of lambda calculus says that every linear lambda term is simply typable, and moreover that its $\beta\eta$-normal form is uniquely determined by its principal type. Mints (1977) gave a proof-theoretic demonstration of this result, by showing that 1) every linear lambda term has a *balanced* principal typing sequent (where a sequent $\Gamma \to A$ is said to be balanced if each atomic formula occurring in it has exactly two occurrences, once in positive position and once in negative position), and 2) any balanced sequent is inhabited by at most one $\beta\eta$-normal form. We give a new more conceptual proof of Mints' pearl theorem, describing it first as a simple bijection between string diagrams for linear normal forms and provable balanced sequents, and second as a simple bidirectional type inference algorithm for linear lambda terms (that is exactly dual to standard bidirectional type checking).

Linear lambda calculus represents an extremal case of parametricity, in the sense that every linear lambda term is typable, and its normal form is uniquely identified by its principal type. For example, the normal forms on the left are uniquely identified by the types on the right, and vice versa:

$$\begin{array}{ccc}
\lambda x.x(\lambda y.y) & : & ((\alpha \multimap \alpha) \multimap \beta) \multimap \beta \\
\lambda x.\lambda y.xy & : & (\alpha \multimap \beta) \multimap (\alpha \multimap \beta) \\
\lambda x.\lambda y.y(x(\lambda z.z)) & : & ((\alpha \multimap \alpha) \multimap \beta) \multimap ((\beta \multimap \gamma) \multimap \gamma)
\end{array}$$

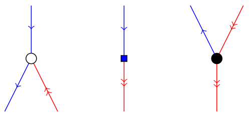
One consequence of this observation is that type inference is essentially equivalent to normalization, and it is in that context that I first saw this property asserted as a "pearl theorem" by Mairson [1]. The result should probably be attributed to Mints [2], who proved a version of it in the context of the so-called coherence problem for monoidal closed categories (also studied notably by Lambek, and by Kelly and Mac Lane). Mints' proof can be divided into the following pair of observations:

1. The principal type of a linear lambda term is *balanced*, in the sense that every type variable occurring in it occurs exactly twice, once positively and once negatively. For example, here I have colored in red and blue the positive and negative occurrences of the three type variables in the principal type of $\lambda x.\lambda y.y(x(\lambda z.z))$: $((\alpha \multimap \alpha) \multimap \beta) \multimap (\beta \multimap \gamma) \multimap \gamma$

2. Any balanced type (and more generally, any balanced typing sequent) has at most one inhabitant up to $\beta\eta$-equivalence.

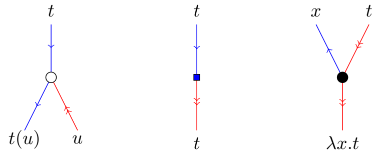
Both (1) and (2) are established by relatively straightforward proof-theoretic arguments (applying an induction on the length of terms).

Although the proof in [2] is not very complicated, a pearl theorem deserves a pearl proof.[1] Our starting point will be a simple string diagram representation of *normal* linear lambda terms that was introduced in [4] as a refinement of so-called lambda-graphs, and which is derived from the refinement type signature [3] encoding the well-known characterization of normal lambda terms in mutual induction with "neutral" terms. In this representation, normal and neutral terms correspond to certain diagrams constructed using three basic combinators which we call a-, s-, and ℓ-nodes (standing for $a$pplication, the $s$witch from neutral to normal, and ℓambda abstraction) on colored, oriented wires:



[1] There are other reasons why I think that the "balanced polymorphism" exhibited in linear lambda calculus is worth being studied on its own terms. For example, it corresponds closely to the notion of "end" in category theory.

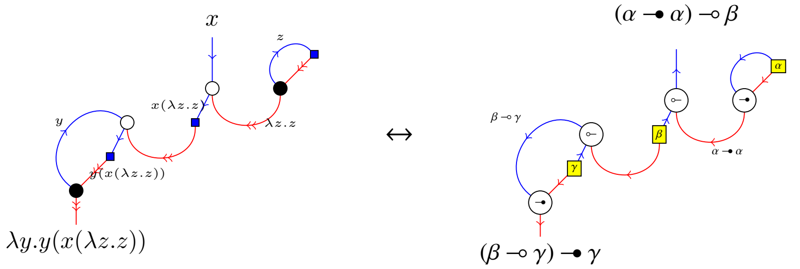
These components may be annotated like so (note that blue wires carry neutral terms and red wires carry normal terms):



From the string diagram of a normal lambda term, you obtain a balanced typing sequent by the following procedure:

1. reverse the orientation of blue (neutral) wires
2. turn each a-node into a negative-polarity implication (i.e., an implication occurring in negative position), and each ℓ-node into a positive-polarity implication
3. replace each s-node by a distinct type variable (with two outgoing wires, standing for its positive and negative occurrences)
4. read off a balanced principal typing sequent for the original term by starting at the type variables and pushing information along the direction of the wires

For example, here is the transformation applied on the diagram of the closed normal term $\lambda x.\lambda y.y(x(\lambda z.z))$ (with its outermost λx removed to view it as a term with one free variable):


x
z
$x(\lambda z.z)$
$\lambda z.z$
y
$y(x(\lambda z.z))$
$\lambda y.y(x(\lambda z.z))$
↔
$(\alpha -\!\!\bullet\, \alpha) \multimap \beta$
α
$\beta \multimap \gamma$
β
γ
$\alpha -\!\!\bullet\, \alpha$
$(\beta \multimap \gamma) -\!\!\bullet\, \gamma$

Moreover, this transformation is clearly reversible (since the choice of type variable names is irrelevant). Finally, we can turn this into a traditional type inference algorithm (much simpler than Hindley-Milner inference on this very special case), described in the following rules:

$$\frac{}{x:\alpha \Leftarrow x \Leftarrow \alpha} \qquad \frac{\Gamma \Leftarrow t \Leftarrow A \multimap B \quad \Delta \Leftarrow u \Rightarrow A}{\Gamma,\Delta \Leftarrow t(u) \Leftarrow B} \qquad \frac{\Gamma \Leftarrow t \Leftarrow \alpha \quad \alpha\ \text{fresh}}{\Gamma \Leftarrow t \Rightarrow \alpha} \qquad \frac{x:A,\Gamma \Leftarrow t \Rightarrow B}{\Gamma \Leftarrow \lambda x.t \Rightarrow A \multimap B}$$

Here the two judgment forms are $\Gamma \Leftarrow t \Leftarrow A$ ("checking against A, t synthesizes context Γ") and $\Gamma \Leftarrow t \Rightarrow A$ ("t synthesizes context Γ and type A"). The reader familiar with standard bidirectional type checking (where neutral terms synthesize and normal terms check) will remark that this system is exactly dual (suggesting that we might call it "bidirectional type inference").

References

[1] Harry G. Mairson. Linear lambda calculus and PTIME-completeness. JFP, 14:6 (2004).

[2] Grigorii E. Mints. Closed categories and the theory of proofs. Zapiski Nauchnykh Seminarov LOMI im. V.A. Steklova AN SSSR, 68 (1977). Translation in *Journal of Soviet Mathematics*, 15 (1981), republished in Mints, *Selected Papers in Proof Theory*, Bibliopolis (1992).

[3] Frank Pfenning. Refinement Types for Logical Frameworks. TYPES 1993.

[4] Noam Zeilberger and Alain Giorgetti. A correspondence between rooted planar maps and normal planar lambda terms. To appear in *Logical Methods in Computer Science*.