Combinators;
or, a semantic argument for the extensional principle *

Peter Freyd
University of Pennsylvania
pjf@upenn.edu

January 27, 2017

Combinator Algebra has always struck me as a subject almost impossible to watch: it's a lousy spectator sport but has a reputation for being habit-forming as a participator sport. It is only with reluctance that I was talked into ever writing this up: I was particularly dubious about the newness of the proof of the extensional principle; it seemed too easy to have been overlooked.[1]

Let A be a set with a binary operation denoted by catenation. Every element $a \in A$ names a unary operation on A, to wit, the operation that sends x to ax. The first goal of combinatorial logic is to obtain a non-trivial A such that every unary operation is so nameable. In the usual foundations and with the usual interpretation of the word 'every' this is, of course, impossible. Let us, here, tinker with the notion of 'every'; we seek an A such that every unary operation *that can be described with a term* is nameable by an element of A. Before we get to a formal definition we'll consider a few special cases: A must have an element I that names the identity operation, that is, that satisfies the equation:

$$\mathsf{I}x = x$$

For each element $x \in A$ we will need another element x' that names the *constant operation* whose unique value is x. But this is itself a unary operation (from x to x') and it needs a name. Hence A must have an element K that satisfies the equation:

$$(\mathsf{K}x)y = x$$

We will henceforth use the standard combinator convention that in the absence of parentheses the binary operation is performed from left to right: $xyz = (xy)z$. Hence the above equation may be rewritten:

$$\mathsf{K}xy = x \quad [2]$$

For the next example note that for each element x we need an element that names the operation obtained by *evaluating at* x, and the unary operation that delivers this name is

* An earlier draft of this paper appeared in *Categories in computer science and logic* (Boulder, CO, 1987) Contemp. Math., 92, Amer. Math. Soc., Providence, RI, 1989

[1] J.P.Seldin in MathReviews said "Although the author expresses doubts about the newness of the proof of the extensional principle, the reviewer has never seen it and finds it interesting: if the proof is not new, it is an interesting presentation of an old proof." (But be warned: M.W.Bunder in zbMATH said "[T]he surprisingly short algebraic proof that [the extensional principle] does hold... not clear to the reviewer.")

[2] So we'll never start with "(" and we'll never have "((" anywhere.

itself named by an element E, that is, an element that satisfies the equation:

$$\mathsf{E}xy = yx$$

Our requirement for unary operations forces a requirement about binary operations. For example the *composition* operation: given two elements x and y let $x \circ y$ denote the operation defined by $(x \circ y)z = x(yz)$. For each x and y there must be an element that names the operation $x \circ y$. For each x there must be a unary operation that sends y to the element that names $x \circ y$. There must be an element that names the operation that sends x to the element that names the operation that sends each y to the element that names $x \circ y$. This last element is traditionally denoted B. All is summarized by the equation:

$$\mathsf{B}xyz = x(yz)$$

The binary operation that sends the pair x, y to $x \circ y$ is named by B since $\mathsf{B}xy = x \circ y$.

The standard notation for undoing this mess of words is, of course, Alonzo Church's λ-calculus: $x \circ y = \lambda z.x(yz)$, $\mathsf{B}x = \lambda y.x \circ y$, $\mathsf{B} = \lambda xyz.x(yz)$.

(It is worth noting that I, K and E are easily interpretable as binary operations: in reverse order, E names the binary operation that is the 'transpose' of the binary operation we started with; K names the binary operation usually called the *left projection* operation. I names the binary operation we started with. The *right projection* operation is named by KI (since $\mathsf{KI}xy = \mathsf{I}y = y$).

The first goal of combinatorial logic is now formalizable as follows: for any expression ϕ in the language and for any variable x we need an expression ψ that names the operation that carries x to ϕ. That is, $\psi x = \phi$ where x does not appear in ψ. (In the language of λ-calculus, we are requiring the set of expressions to be closed under λ-abstraction.) Moses Schönfinkel solved the existential side back in the early 1920s: define a **combinator algebra** as a set with a binary operation denoted with catenation and constants K and S satisfying the equations:

$$\mathsf{K}xy = x \qquad\qquad \mathsf{S}xyz = xz(yz)$$

That's all. Note first that if we define I as SKK then we may easily verify the equation $\mathsf{I}x = x$. The proof of **functional completeness**—as it is usually called—proceeds as follows: if x does not occur in a term ϕ then we may take ψ to be $\mathsf{K}\phi$, if x is ϕ then ψ may be taken to be I; in the remaining case ϕ is necessarily of the form $\phi'\phi''$ and by induction we may assume that ψ' and ψ'' are x-free terms such that $\psi' x = \phi'$ and $\psi'' x = \phi''$ so that we may take ψ to be $\mathsf{S}\psi'\psi''$.

(It's worth checking that E is definable as $\mathsf{S}(\mathsf{K}(\mathsf{SI}))\mathsf{K}$ and B as $\mathsf{S}(\mathsf{KS})\mathsf{K}$.[3] The adventurous might go on to find a variable-free term Y such that $\mathsf{Y}x = \mathsf{SII}(\mathsf{B}x(\mathsf{SII}))$. Then $\mathsf{Y}x = \mathsf{B}x(\mathsf{SII})(\mathsf{B}x(\mathsf{SII})) = x(\mathsf{SII}(\mathsf{B}x(\mathsf{SII}))) = x(\mathsf{Y}x)$, that is, Y names a function which delivers a fixed-point for any named unary function.[4])

Given a combinator algebra A, elements a and a' are **extensionally equivalent** if they name the same unary operation, that is, if for all $x \in A$ it is the case that $ax = a'x$. The **extensional principle** says that extensional equivalence implies equality. It is the uniqueness condition for functional completeness and it may be viewed as a cancellation condition:

[3] There's an appendix below devoted to subscoring. Take a look.

[4] If you want to find how anyone came up with Y try to use Cantor's "diagonal proof" to show that there must be an unnameable function and note how that proof critically uses the existence of a fixed-point-free function: using that $fx \neq x$ for all x Cantor obtains a function $gx = f(xx)$ which if it were already named would lead to a contradiction: $gg = f(gg) \neq gg$.

If ψ and ψ' are x-free terms such that $\psi x = \psi' x$ for all x then $\psi = \psi'$.

As just one example, note that our definition of I as SKK was a bit arbitrary: SKS would have done as well. The extensional principle easily implies that $\mathsf{SKK} = \mathsf{SKS}$.[5] We can regard the extensional principle as a rule of inference. It is a remarkable fact—first discovered by Haskell Curry—that, instead, one needs only a finite number of its equational consequences. (A combinator algebra satisfying the extensional principle is often called a **Curry algebra**.) There's an apocryphal story to the effect that combinator theorists needed to carry with them a copy of those finite number of equations.

Our approach starts as follows: let A be a subalgebra of B and b an element of B. *The subalgebra generated by A and b is constructable as $Ab = \{ab : a \in A\}$.* The proof: Ab is closed under the primitive binary operation because $(ab)(a'b) = (\mathsf{S}aa')b$; it contains the constants because $(\mathsf{KK})b = \mathsf{K}$ and $(\mathsf{KS})b = \mathsf{S}$; it contains A because $(\mathsf{K}a)b = a$; it contains b because $\mathsf{I}b = b$.

The set Ab is the image of the obvious function from A (hence this enlargement of A can be constructed as a quotient of A). We will specialize to the case that B is the "polynomial algebra" $A[X]$, the result of freely adjoining a generator X to A and we will take the element b to be X. The "obvious function" from A to $A[X]$ sends $a \in A$ to $aX \in A[X]$. *The extensional principle says that this onto function is one-to-one.* It says, therefore, that we can construct $A[X]$ using A as its underlying set. We may do so as follows:

Given a combinator algebra A define $\bullet A$ ("dot A") to be the combinator algebra obtained by taking the same set as that for A and defining the primitive binary operations $x \cdot y$ ("dot product") as $\mathsf{S}xy$; defining $\dot{\mathsf{K}}$ ("dot K") as KK; defining $\dot{\mathsf{S}}$ ("dot S") as KS. We need the equations:

EP1•) $\quad \dot{\mathsf{K}} \cdot x \cdot y = x$

EP2•) $\quad \dot{\mathsf{S}} \cdot x \cdot y \cdot z = x \cdot z \cdot (y \cdot z)$

The undotted versions:

EP1) $\quad \mathsf{S}\big(\mathsf{S}(\mathsf{KK})x\big)y = x$

EP2) $\quad \mathsf{S}\Big(\mathsf{S}\big(\mathsf{S}(\mathsf{KS})x\big)y\Big)z = \mathsf{S}(\mathsf{S}xz)(\mathsf{S}yz)$

These are the first two of the four equations we'll use to obtain the extensional principle. It is worth checking immediately that they are consequences of the extensional principle: for EP1) compute $\mathsf{S}(\mathsf{S}(\mathsf{KK})x)yt = \mathsf{S}(\mathsf{KK})xt(yt) = \mathsf{KK}t(xt)(yt) = \mathsf{K}(xt)(yt) = xt$; for EP2) $\mathsf{S}(\mathsf{S}(\mathsf{S}(\mathsf{KS})x)y)zt = \mathsf{S}(\mathsf{S}(\mathsf{KS})x)yt(zt) = \mathsf{S}(\mathsf{KS})xt(yt)(zt) = \mathsf{KS}t(xt)(yt)(zt) = \mathsf{S}(xt)(yt)(zt) = xt(zt)(yt(zt)) = \mathsf{S}xzt(\mathsf{S}yzt) = \mathsf{S}(\mathsf{S}xz)(\mathsf{S}yz)t$. (See? It's a lousy spectator sport.[6])

The inclusion map from A into $A[X]$ corresponds to what we will call the **K-map**, the function that sends $a \in A$ to $\mathsf{K}a \in \bullet A$. We will need an equation to say that this is a homomorphism of combinator algebras:

EP3•) $\quad \mathsf{K}(xy) = (\mathsf{K}x) \cdot (\mathsf{K}y)$

The undotted version is:

EP3) $\quad \mathsf{K}(xy) = \mathsf{S}(\mathsf{K}x)(\mathsf{K}y)$

[5] Indeed, it implies that $\mathsf{SK}y$ is independent of y. Another way of saying this is that both SK and KI name the right projection binary operation.

[6] Try the appendix for an effort to make the (mindless!) identity checking a bit more acceptable.

The verification of EP3 using the extensional principle is easier than it was for the previous cases: $\mathsf{K}(xy)t = xy = \mathsf{K}xt(\mathsf{K}yt) = \mathsf{S}(\mathsf{K}x)(\mathsf{K}y)t$. The fact that the K-map preserves the two constants is, of course, an immediate consequence of the definitions of $\dot{\mathsf{K}}$ and $\dot{\mathsf{S}}$. [7]

We will not need the freeness of $\bullet A$ [8] but it continues to serve as a discovery procedure. It yields our final equation:

EP4$^\bullet$) $\quad (\mathsf{K}x)\cdot\mathsf{I} = x$ [9]

The undotted version is:

EP4) $\quad \mathsf{S}(\mathsf{K}x)\mathsf{I} = x.$

The verification: $\mathsf{S}(\mathsf{K}x)\mathsf{I}t = \mathsf{K}xt(\mathsf{I}t) = xt.$ [10] [11]

The semantic argument is now easy: suppose that ψ and ψ' are x-free terms such that $\psi x = \psi' x$ in any combinator algebra. Then for any combinator algebra A satisfying EP1-4 the extensional equivalence of ψ and ψ' says that in $\bullet A$ it is the case that $\mathsf{K}\psi\cdot\mathsf{I} = \mathsf{K}\psi'\cdot\mathsf{I}$ and EP4 implies $\psi = \psi'$.

But there is a little problem. The last paragraph works for canceling one variable but not two. Suppose ψ, ψ' are both x- and y-free terms such that $\psi xy = \psi' xy$. Then, yes, $\psi x = \psi' x$ holds in any combinator algebra satisfying EP1-4 but we have not shown that it holds in any combinator algebra and there's no reason to believe that $\mathsf{K}\psi\cdot\mathsf{I} = \mathsf{K}\psi'\cdot\mathsf{I}$ holds in $\bullet A$.

Fortunately there's an easy solution. We can insure that EP1–4 hold in $\bullet A$. The K- and S-rules are enough to replace any axiom on n variables with a (stronger) axiom of the form $\psi x_1 x_2 \dots x_n = \psi' x_1 x_2 \dots x_n$, where ψ, ψ' are variable-free. If the axiom $\psi = \psi'$ holds in A then it holds in $\bullet A$. [12]

$$\begin{array}{lrcl}
\text{EP}1^\star) & \mathsf{BS}(\mathsf{S}(\mathsf{KK})) & = & \mathsf{K} \\
\text{EP}2^\star) & \mathsf{B}(\mathsf{BS})(\mathsf{BS}(\mathsf{S}(\mathsf{KS}))) & = & \mathsf{S}(\mathsf{BB}(\mathsf{BS}(\mathsf{B}(\mathsf{BS})\mathsf{S})))(\mathsf{KS}) \\
\text{EP}3^\star) & \mathsf{BK} & = & \mathsf{S}(\mathsf{BB}(\mathsf{BSK}))(\mathsf{KK}) \\
\text{EP}4^\star) & \mathsf{S}(\mathsf{BSK})(\mathsf{KI}) & = & \mathsf{I}
\end{array}$$

Appendix: A few subscorings.[13] (It is said that "subscoring" is short for "substitution underscoring.")

---

[7] For any non-trivial A it is comforting to check that I is not in the image of the K-map: if $a \in A$ were such that $\mathsf{I} = \mathsf{K}a$ then for any $a' \in A$ it would be the case that $a' = \mathsf{I}a' = \mathsf{K}aa' = a$.

[8] Given a homomorphism $g : A \to B$ and an element of $b \in B$ we seek a homomorphism $f\colon \bullet A \to B$ so that $f(\mathsf{K}a) = ga$ for all $a \in A$ and $f\mathsf{I} = b$. Define it by $fx = gxb$. f is a homomorphism because $f(x\cdot y) = g(\mathsf{S}xyb) = \mathsf{S}(gx)(gy)b = gxb(gyb) = fx(fy)$, $f\dot{\mathsf{K}} = g(\mathsf{KK})b = \mathsf{KK}b = \mathsf{K}$, $f\dot{\mathsf{S}} = g(\mathsf{KS})b = \mathsf{KS}b = \mathsf{S}$. And, further, $f\mathsf{I} = g\mathsf{I}b = \mathsf{I}\,b = b$.

[9] EP4 is a special case of the uniqueness condition for the induced maps resulting from freeness: it says that if one chooses B to be $\bullet A$ and g to be the K-map and b to be I, then using the previous footnote f has to be the identity function. Conversely any map $f\colon \bullet A \to B$ is necessarily definable as $fx = gxb$ where $g : A \to B$ is defined by $gx = f(\mathsf{K}x)$ and b is defined as $f\mathsf{I}$. The proof is immediate: $fx = f((\mathsf{K}x)\cdot\mathsf{I}) = f(\mathsf{K}x)(f\mathsf{I}) = f(gxb)$. (Actually there's an entirely soft proof using the fact that the $\bullet A$-construction is functorial.)

[10] Any finite number of equations can be replaced with just one, e.g. the pair $\phi = \phi'$, $\psi = \psi'$ is equivalent to $t\phi\psi = t\phi'\psi'$ for a fresh variable t (because for $t = \mathsf{K}$ we obtain $\phi = \phi'$ and for $t = \mathsf{KI}$ we get $\psi = \psi'$).

[11] If we call the $\bullet A$-structure the "derived structure" on A note that we needn't stop with just one derivation. We obtain an infinite sequence of combinator-algebra structures, in particular, an infinite sequence of monoid structures on A (beginning with I and B) each of which distributes over the all the further structures. See Rick Statman's paper "Freyd's Hierarchy of Combinator Monoids" in the *Sixth Annual IEEE Symposium on Logic in Computer Science* 1991.

[12] We're used to the fact for that any onto homomorphism the equational axioms holding in its source continue to hold in its target. Combinator algebras satisfying the extensional principle thus have the special feature that the word "onto" can be removed.

[13] For other examples check out the ending pages of `www.math.upenn.edu/~pjf/amplifications.pdf` and the last section of `www.math.upenn.edu/~pjf/analysis.pdf`

I, E and B:

$$\begin{array}{c} \mathsf{I}x \\ = \\ \mathsf{SKK}x \\ \hline\hline \mathsf{K}x(\mathsf{K}x) \\ \hline\hline x \end{array} \qquad \begin{array}{c} \mathsf{E}xy \\ = \\ \mathsf{S}(\mathsf{K}(\mathsf{SI}))\mathsf{K}xy \\ \hline\hline\hline \mathsf{K}(\mathsf{SI})x(\mathsf{K}x)y \\ \hline\hline\hline \mathsf{SI}(\mathsf{K}x)y \\ \hline\hline\hline \mathsf{I}y(\mathsf{K}xy) \\ \hline\hline\hline\hline yx \end{array} \qquad \begin{array}{c} \mathsf{B}xyz \\ = \\ \mathsf{S}(\mathsf{KS})\mathsf{K}xyz \\ \hline\hline\hline \mathsf{KS}x(\mathsf{K}x)yz \\ \hline\hline\hline\hline \mathsf{S}(\mathsf{K}x)yz \\ \hline\hline\hline\hline \mathsf{K}xz(yz) \\ \hline\hline\hline\hline x(yz) \end{array}$$

Y:

$$\begin{array}{c} \mathsf{S}(\mathsf{B}(\mathsf{B}(\mathsf{SII}))\mathsf{B})(\mathsf{K}(\mathsf{SII}))x \\ \hline\hline\hline\hline \mathsf{B}(\mathsf{B}(\mathsf{SII}))\mathsf{B}x(\mathsf{K}(\mathsf{SII})x) \\ \hline\hline\hline\hline\hline\hline \mathsf{B}(\mathsf{SII})(\mathsf{B}x)(\mathsf{SII}) \\ \hline\hline\hline\hline\hline\hline\hline \mathsf{SII}(\mathsf{B}x(\mathsf{SII})) \\ \hline\hline\hline\hline\hline\hline\hline\hline \mathsf{Y}x \end{array} \qquad \begin{array}{c} \mathsf{SII}(\mathsf{B}x(\mathsf{SII})) \\ \hline\hline\hline\hline \mathsf{I}(\mathsf{B}x(\mathsf{SII}))(\mathsf{I}(\mathsf{B}x(\mathsf{SII}))) \\ \hline\hline\hline\hline\hline\hline \mathsf{B}x(\mathsf{SII})(\mathsf{B}x(\mathsf{SII})) \\ \hline\hline\hline\hline\hline\hline\hline \mathsf{x}(\mathsf{SII}(\mathsf{B}x(\mathsf{SII}))) \\ \hline\hline\hline\hline\hline\hline\hline\hline x(\mathsf{Y}x) \end{array}$$

EP1–2:

$$\begin{array}{c} \mathsf{S}(\mathsf{S}(\mathsf{KK})x)yt \\ \hline\hline\hline\hline \mathsf{S}(\mathsf{KK})xt(yt) \\ \hline\hline\hline\hline \mathsf{KK}t(xt)(yt) \\ \hline\hline\hline \mathsf{K}(xt)(yt) \qquad \mathsf{K}xyt \\ \hline\hline\hline\hline\hline\hline\hline\hline\hline xt \end{array} \qquad \begin{array}{c} \mathsf{S}(\mathsf{S}(\mathsf{S}(\mathsf{KS})x)y)zt \\ \hline\hline\hline\hline\hline\hline\hline\hline\hline \mathsf{S}(\mathsf{S}(\mathsf{KS})x)yt(zt) \\ \hline\hline\hline\hline\hline\hline\hline\hline \mathsf{S}(\mathsf{KS})xt(yt)(zt) \\ \hline\hline\hline\hline\hline\hline\hline\hline \mathsf{KS}t(xt)(yt)(zt) \qquad \mathsf{S}(\mathsf{S}xz)(\mathsf{S}yz)t \\ \hline\hline\hline\hline \mathsf{S}(xt)(yt)(zt) \qquad \mathsf{S}xzt(\mathsf{S}yzt) \\ \hline xt(zt)(yt(zt)) \end{array}$$

EP3–4:

$$\begin{array}{c} \mathsf{K}(xy)t \\ \hline\hline\hline\hline\hline xy \end{array} \qquad \begin{array}{c} \mathsf{S}(\mathsf{K}x)(\mathsf{K}y)t \\ \hline\hline\hline\hline\hline\hline\hline \mathsf{K}xt(\mathsf{K}yt) \\ \hline\hline\hline\hline\hline\hline\hline\hline\hline xy \end{array} \qquad \begin{array}{c} \mathsf{S}(\mathsf{K}x)\mathsf{I}t \\ \hline\hline\hline\hline\hline\hline\hline \mathsf{K}xt(\mathsf{I}t) \\ \hline\hline\hline\hline\hline\hline\hline\hline\hline xt \end{array} \qquad \begin{array}{c} \mathsf{I}xt \\ \hline\hline\hline\hline\hline\hline\hline\hline\hline xt \end{array}$$

EP1*:

$$\dfrac{\mathsf{BS(S(KK))}xy}{\mathsf{S(S(KK)}x)y}$$

EP2*:

$$\dfrac{\dfrac{\dfrac{\mathsf{B(BS)(BS(S(KS)))}xyz}{\mathsf{BS(BS(S(KS))}x)yz}}{\mathsf{S(BS(S(KS))}xy)z}}{\mathsf{S(S(S(KS)}x)y)z} \qquad \dfrac{\dfrac{\dfrac{\mathsf{B(BS)(BS(S(KS)))}xyz}{\mathsf{BS(BS(S(KS))}x)yz}}{\mathsf{BS(S(S(KS)}x))y)z}}{\mathsf{S(S(S(KS)}x)y)z} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathsf{S(BB(BS(B(BS)S)))(KS)}xyz}{\mathsf{BB(BS(B(BS)S))}x(\mathsf{KS}x)yz}}{\mathsf{B(BS(B(BS)S)}x)\mathsf{S}yz}}{\mathsf{B(S(B(BS)S}x))\mathsf{S}yz}}{\mathsf{B(S(BS(S}x)))\mathsf{S}yz}}{\mathsf{S(BS(S}x))(\mathsf{S}y)z}}{\mathsf{BS(S}x)z(\mathsf{S}yz)}}{\mathsf{S(S}xz)(\mathsf{S}yz)}$$

EP3*–4*:

$$\dfrac{\mathsf{BK}xy}{\mathsf{K}(xy)} \qquad \dfrac{\dfrac{\dfrac{\mathsf{S(BB(BSK))(KK)}xy}{\mathsf{BB(BSK)}x(\mathsf{KK}x)y}}{\mathsf{B(BSK}x)\mathsf{K}y}}{\mathsf{BSK}x(\mathsf{K}y)}}{\mathsf{S(K}x)(\mathsf{K}y)} \qquad \dfrac{\dfrac{\mathsf{S(BSK)(KI)}x}{\mathsf{BSK}x(\mathsf{KI}x)}}{\mathsf{S(K}x)\mathsf{I}}$$

$$\int$$

Available at
`http://www.math.upenn.edu/~pjf/combinators.pdf`