On Cartesian Monoids

by

Rick Statman

April 1996

Introduction

We first learned about Cartesian monoids from Dana Scott and Peter Freyd. Their connection to the simply typed lambda calculus with surjective pairing and the domain equation D = DxD is rather transparent and forms the basis for [7]. In addition, since these monoids always contain a copy of the Freyd-Heller group (see below) there is a further connection to lambda calculus ([6],[10]) . Finally, such monoids come up in the study of type algebras especially concerning Curry's subject reduction theorem ([8],[9]). In short, Cartesian monoids are important for typed lambda calculus.

It is the purpose of this note to collect in one place our observations on Cartesian monoids especially on the free Cartesian monoid. In particular ,we shall solve the unification and matching problems negatively for this structure below. Our approach to treat the free Cartesian monoid as an algebraic structure of the sort we learned about in school. This is not to say that we have anything against the Category Theory approach; it is only to say that we are not competent to carry out such an approach.

0.Contents

1.Cartesian Monoids

A Cartesian monoid is a structure C = (M,*,I,L,R,<>) where (M,*,I) is a monoid with $L \varepsilon M$ and $R \varepsilon M$,
<> : M^2-->M, and
L*<x,y> = x
R*<x,y> = y
<x,y>*z = <x*z,y*z>
<L,R> = I.

Cartesian monoids were first introduced by Dana Scott in [5],and independently by J.Lambek in [3]. The free Cartesian monoid on zero generators is here denoted F. The members of F are denoted by expressions built up from I,L,and R by * and <>.

2.Normal Forms

Each expression can be re-written uniquely in a normal form consisting of a binary

tree,whose nodes correspond to applications of <>,with strings of L's and R's, joined by
*,at its leaves (here I counts as the empty string) and with no subexpression of the form
<L*x,R*x>. This is accomplished by considering the equivalent rewrite system
L*<x,y> --> x
R*<x,y> --> y
<x,y>*z --> <x*z,y*z>
<L*x,R*x> --> x
<L,R> --> I
I*x --> x
x*I --> x
modulo the associativity axioms.
This rewrite system is terminating because we can interpert it in the integers with
rewrites decreasing as follows
L=R=I=2
x*y=x multiplied by y
<x,y>= x+y+1.
The rewrite system is obviously weakly
Church-Rosser therefore it is Church-Rosser.
The binary tree of a given expression is called its Δ.

3.Homomorphisms

For any two members of F,f and g, define f^g:M-->M by f^g(x)= f*x*g (i.e.conjugation).
Given any two distinct normal forms h1 and h2 there exist f and g such that f^g(h1)=

L and f^g(h2)=R. This can be seen as follows. We can first assume that h1 and h2 have the same Δ by expansions of the- form x <-- <L*x,R*x> or I <--<L,R>. Indeed in this way normal expressions can be transformed into various shapes such as one where the binary tree is complete or all strings have the same length. Thus there is an f such that f*h1 =/= f*h2 ,and, both of these reduce to <>-free strings of L's and R's. We can also assume that neither of these strings is a suffix of the other since f could be replaced by either L*f or R*f without loss.Thus there are <>-free h3 and h4 and integers k and l such that

f*h1*<I,I>^k*<R,L>$^{\wedge l}$ = h3*L and

f*h2*<I,I>^k*<R,L>^l = h4*R and
there exist integers n and m such that
h3*L*<<I,I>^n*L,<I,I>^m*R> = L and
h4*R*<<I,I>^n*L,<I,I>^m*R> = R. Thus we can set
g = <I,I>^k*<R,L>^l*<<I,I>^n*L,<I,I>^m*R>.
We conclude that there are no non-trivial homomorphisms of F.

4.Finite Generation of F

The monoid F is finitely generated. We see this as follows. Let Σ = {<X*L,<Y*L*R,Z*R*R>> : X,Y,Z ε {L,R,I}} U {<I,<I,I>>}. Now for any f,g,h <>-free strings of L's and R's the element <f,<g,h>> can be generated from Σ by a simple recursive proceedure. Now say that f is a derivation if it has the form $<<\ldots<f1,f2>\ldots>,fn>$ for $n>2$ such that

$f1=L$

$f2=R$

$f3=I$ and for $j>3$

$fj= <fk,fl>$ for some $k,l < j$ or

$= L*fk$ for some $k<j$ or

$= R*fk$ for some $k<j$.

It is easy to see that every derivation can be generated from Σ using the previous observation. It follows that all of F is generated from Σ.

Any Cartesian monoid which is finitely generated by $f1,\ldots,fn$ is generated by two elements $e0 = <R,<f1,<\ldots<fn,R>\ldots>>>$ and $e1 = L$. For F we denote e0 by E.

5.The Group

Let H be the submonoid of right invertible elements and let G be the group of (doubly) invertible elements of F. Clearly L and R belong to H. If we begin with f in normal form then it is easy to see that

$f\varepsilon H$ <=> f can be expanded so that all of its strings at the leaves have the same length and none occurs more than once

f has left inverse <=> f can be expanded so that all of its strings have the same length n and each of

2^n strings of this length
actually occur

fεG <=> f can be expanded so that all of its
strings have the same length n and
each of the 2^n strings of this length
occurs exactly once.

It follows that H = L*G = R*G. Let

Bn=<L,<...<L*R^n-1,<L*L*R^n,<R*L*R^n,R^n+1>>>...>

C0=<R,L>

Cn+1=<L,<...<L*R^n-1<L*R^n+1,<L*R^n,R^n+2>>>...>>

Clearly, both Bn and Cn are invertible and the set of all of them generate G. Indeed observe that if n<m then

Bn*Bm= Bm+1*Bn.

The group generated by the Bn alone is (anti)isomorphic to the Freyd-Heller group [1]. It is generated by B0 and B1 as a group. Thus it is easy to see that G is generated by B0,B1,C0, and C1 as a group.

6.A Wreath Product

Let J be the monoid of all number theoretic functions of finite support so that s:N -> N belongs to J if there exists n such that for m>n s(m) = m. Suppose that t:N -> F so that for m>k t(m) = I and s and n are as above; let l=max{n,k}, then the pair (t,s) can be represented by

<t(0)*L*R^s(0),<...<t(l)*L*R^s(l),R^l+1>...>>.

This representation gives an embedding of the wreath product of F with J into F (this should be compared to [1]T6).

7.Representation

In [1] the authors give a faithful representation of the Freyd-Heller group in the continuous order preserving permutations of the real numbers. Here we will generalize a modified such representation to F. Let CS be Cantor space ,here construed as the product of {0,1}, endowed with the discrete topology, along N. The properties of CS are very well known; in particular, CS is a totally disconnected compact Hausdorf space. Among the continuous open mappings A : CS -> CS are the shift operators Z and O defined by

Z(f)(0) = 0 O(f)(0) = 1

Z(f)(n+1) = f(n) O(f)(n+1) = f(n)

We simply write 0f for Z(f) and 1f for O(f). If **C** is a collection of mappings A : CS -> CS we let piecewise **C** be the closure of **C** under the following kind of definition of A by cases from A' and A"

A(0f) = A'(f)

A(1f) = A"(f).

Indeed if all **C** mappings are continuous and open then so are all piecewise **C** mappings. The piecewise shift operators A can be explicitly characterized by the following condition: Whenever A(f) = g there exists basic open neighborhoods (f(0),...,f(r)) and (g(0),...,g(s)) containing resp f and g such that for any t>s g(t) = f(t-s+r). We define a Cartesian monoid structure on the piecewise shift operators as follows;

I = I

L = Z

R = O

x*y = the composition z |-> y(x(z))

$$<x,y>(f) = \begin{matrix} x(f) \text{ if } f(0) = 0 \\ y(f) \text{ if } f(0) = 1. \end{matrix}$$

It is not difficult to see that this Çartesian monoid is isomorphic to F. Now let us order the members of CS lexicographically and let G+ be the order preserving members of G (under this isomorphism). The G+ is precisely the Freyd-Heller group.

8.The Polynomial Monoid F[x]

All of the principal results mentioned above for F hold as well for F[x]. More generally, if f(x) & g(x) are distinct normal expressions then there exists an hεF such that f(h) =/= g(h). Indeed, if f(x1,...,xn) and g(x1,...,xn) are distinct normal expressions in F[x1,...,xn] then we shall find h1,...,hn such that f(h1,...,hn) =/= g(h1,...,hn). The construction takes two steps.In the first step n may be increased.We remove subexpressions of the form L*xi*h and R*xi*h (for h possibly empty) by making substitutions xi <- <y,z> and re-normalizing.It is easy to see that this process terminates and that the original f and g are recoverable by the substitutions y <- L*xi and z <- R*xi. Thus we can assume that the first step is completed and f and g are normal,distinct, and have no subexpressions of the above forms. Indeed expressions like this can be recursively generated as a string of xi's followed by a string of L's and R's or a string of xi's followed by a single < > of expressions of the same form. Given such an expression e, if we evaluate each xi,L, and R as 1,<> as max, and * as + ,then the result is a positive integer #e (the "length of the longest path in e").Let m=max{#f,#g}+1,and k=m(m+n+1).For each positive integer i set hi=

```
<<R^k,<...<R^k,I>...>>,R^k>
         \---------/
            m+i
```

We shall show that both f(x1,...,xn) and g(x1,...,xn) are reconstructible from the normal forms of f(h1,...,hn) and g(h1,...,hn) resp. and thus f(h1,...,hn)=/=g(h1,...,hn). Toward this end note that if t is a normal expression for a member of F and #t<k then hi*t = e =

```
                                  df
       <<t',<...<t',t>...>t'>
```

```
\-----/
  m+i
```

where t' is <>-free and #e<#t + m+n+2.Now consider either f(h1,...,hn) or g(h1,..,hn).The normal form of this expression can be computed recursively bottom-up as in the computation of e from hi*t above.Observe that no subexpression of the form <L*h,R*h> is introduced since each t' begins with R. In order to reconstruct, say, f(x1,...,xn) proceed top-down to find subterms e as above with t' <>-free. By choice of m such a subterm is not the "trace" ([2]pg 18) of a subterm of f(h1,...,hn) disjoint from the hi. Such subterms cannot overlap because their left components have <>. Finally, consider any of the pairs <> in e. Such a pair cannot be the trace of a pair in f(h1,...,hn) disjoint from the hi since the left component of hi contains <>. Thus e = hi*t as above.

9.Integers in F

Let Int = { R^n : n=0,1,... } with n = R^n.

(i) fεInt <=> f*R = R*f

Indeed if f*R=R*f then, taking f in normal form ,f cannot have a non trivial Δ. Thus f is a string of L's and R's.

(ii) fεF*L <=> f*<L,L> = f

fεF*R <=> f*<R,R> = f

These can be proved by induction on normal forms.

(iii)We say that f is an n-sequence if f has the form <f0*L,<f1*L,<...<fn-1*L,R>...>>>. For fεH we have f is an n-sequence <=> R^n*f = f. This can be easily seen from paragraph (5).

(iv)Define

Copy(f,n) =
<f*L,<f*R*L,<...<f*R^n-1*L,R>...>>>
Iterate(f,n) =
<f^n*L,<f^n-1*L*R,<...<f*L*R^n-1,R^n>...>>>.

These are related in the following way.

g=Copy(f,n) <=> g is an n-sequence &
g = R*g*<<L,L>,f*R^n-1*L,R>>
g=Iterate(f,n) <=> g = Copy(f*L)*<I,R^n>*R*g*<I,<
f*L*R^n-1,R^n>> .

Moreover, if fεInt then Copy(f*L,n)εH and Iterate(f,n)ε H.Finally,

g=f^n <=> g = L*Iterate(f,n)*<I,I>.

Now it follows from paragraph (5) and (i),(ii), and (iii) above that for any Diophantine set S of intgers there exists an F polynomial f(x,y) and gεF such that

nεS <=> there exists hεF such that f(n,h)=g.

Briefly, this is because for f,gεH

f=g <=> there exists h,t such that
<f,h>*t = I & t*<f,h> = I &
<g,h>*t = I & t*<g,h> = I.

Thus by the famous theorem of Matiyasevich

([4]) the matching problem for F is unsolvable. With a bit more work it can be shown that every RE subset of H is the set of projections of such a matching problem. We do not believe that this extends to the whole of F. In particular, we conjecture that the set of simplicies { <I,I>^n | n a natural no. } is not the projection of a matching problem. It is not hard to see that if this set is such a projection then every RE subset of F is as well.

10.Finitely Generated Submonoids

We shall next show that any finitely generated submonoid of F is the set of projections of an F unification problem.This requires some definitions. First we want to characterize n- sequences for f not in H. Let Copy(n) = Copy(L,n), then

f = Copy(n) <=> R^n*f = R & f = R*f*<<L,L>,<L*R^n-1*L,
R>>

and

f is an n-sequence <=> R^n*f = R &
Copy(n)*<I,L*R^n-1>*f = Copy(n)*<I,L*R^n-1>*f*<L,L>

Consider the first biconditional.

The direction <= can be seen as follows.If R^n*f = R we can write f = <f1,<f2,<...<fn,R>...>>, and we can compute

R*f*<<L,L>,<L*R^n-1*L,R>>=
<f2*<<L,L>,<L*R^n-1*L,R>>,<...<fn*<<L,L>,<L*R^n-1*L,R>>
,<L*R^n-1*L,R>>...>>

If this = f then for i=1,...,n-1 fi = fi+1*<<L,L>,<L* R^n-1*L,R>> and fn= L*R^n-1*L. Thus fi= L*R^i-1*L and f = Copy(n). The direction => is obvious.

The second biconditional is proved similarly.As above

g = f^n <=> g = L*Iterate(f,n)*<I,I>.

If s:N --> N let Copy(f,s,n) =

<f*R^s(0)*L,<...<f*R^s(n-1)*L,R>...>>

so Copy(f,n) = Copy(f,identity,n).In addition, let Comp(f,s,n) =

<L*f*R^s(0)*L,<...<L*R^n-1*f*R^s(n-1)*L,R>...>>.

The point of these definitions is that Comp can be expressed in terms of Copy and Comp effects multi-ary compositions. Indeed for f = <f0,<...< fn-1,R>...>> and g = <g0,<...<gn-1,R>...>> define f#g = <fo*g0,<...<fn-1*gn-1,R>...>>. Then f#g= Comp(f*L,identity,n)*<I,R^n>*g.

(i) There exists s such that g = Copy(I,s,n) iff
g is an n-sequence & g*<R,R> = Copy(R*L,identity, n)*<I,R^n>*g

For <=, if g = <g0*L,<...<gn-1*L,R>...>> we compute Copy(R*L,identity,n)*<I,R^n>*g = <R*g0,<...<R*gn-1,R>. ..>>

g*<R,R> = <g0*R,<...<gn-1*R,R>...>>

and if these are equal we have for each i=0,...,n-1 R*gi=gi*R.Thus by paragraph 8 (i) there is an s such that gi = R^s(i).

(ii)g = Comp(f,identity,n) iff there exist h1,h2,h3 such that

a) h1 is an n-sequence

b) h2 is an n^2-sequence
c) there exists an s such that h3 = Copy(I,s,n)
d) f = h1*<I,R^n*f>
e) h2 = R^n*h2*<<L,L>,h1*<R^n-1*L,R>>
f) h3 = R*h3*<<I,I>^n+1*L,<R^(n^2-1)*L,R>>
g) g = Copy(L*L,identity,n)*<L,R^n>*h3*<h2,R>

For => suppose that f= <f0,<...<fn-1,fn>...>>.Let h1= <f0*L,<...<fn-1*L,R>...>> so a) and d) are satisfied. Let h2 = <f0*L,<...<fn-1*L,<f0*R*L,<...<f^n-1*R*L,<... <f0*R^n-1*L,<...<fn-1*R^n-1*L,R>...>>...>>...>>>...>> so b) and e) are satisfied. Finally, we put h3= <L,<R^n+1*L,<R^2n+2*L,<...<R^(n^2-1)*L,R>...>>>> so c) is satisfied by the function s defined by s(i)= i(n+1),and f) and g) are satisfied as well.
For <= it is easy to argue that h1,h2,and h3 satisfying a)-g) must be as above in =>.

(iii)There exists an s such that g=Comp(f,s,n) iff
there exists s such that g = Comp(f,identity,n)*
<Copy(I,s,n),R>.

Now suppose that f1,...,fk are given. We will express membership in the submonoid generated by f1,..., fk. Let Fit(n) = {f : f = <fs(1)*L,<...<fs(n)*L,R>...>> for some s:[1,n] --> [1,k] }. We say that f is an n-permutation if f = Copy(L,s,n) for some permutation s:[0,n-1] --> [0,n-1]. It should be clear that

(iv)f is an n-permutation iff there exists s and m
such that f = Copy(L,s,n) and (f*<I,R^n>)^m = I.

(v) fεFit(n) <=> there exist integers m1,...,mk such
that m1+...+mk=n and there exists g
such that g is an n-permutation and

f = g*<I,R^n>*Copy(f1,zero,m1)*...*
Copy(fk,zero,mk)

Finally we conclude that f belongs to the submonoid generated by f1,...,fk if and only if there is an n such that there exists an n+1-sequence h and gε Fit(n) with f = L*h*<I,R> & h=((g*<I,R>)#(R*h))* <L,<I*L,R>>.In particular the members of the submonoid generated by f1,...,fk are the projections of solutions to the above unification problem. When f1 = L and f2 = R we write Bit(n) for Fit(n), and "n-string" for "a <>-free string of L's and R's of length n."

11.Godel Numbering

We define Binary(f,g) <=> for some m, f=R^m and g is a <>-free string of L's and R's such that if bi is defined by

1 if the ith element of g is L
bi =
0 if the ith element of g is R

when g is read from right to left and i=0,1,...,n-1 then
m= (bn-1)2^n-1+...+(b1)2+b0.

We assign to each member of F a non-unique Godel number as follows. Let f = e(i1)*...*e(ik) as in the last sentence of pargraph 2 and let m be as above (in binary) such that bj = 0 <=> ij = 0 and bj = 1 <=> ij = 1; then m is a Godel number of f provided bn-1=1. We can do the same with F[x]. Every member of F(F[x]) has a Godel number since L*<I,I> = I. The following are the key facts.

(i)Binary(f,g) <=> there are integers m and n and h1,h2,h3,h4,h5 such that

(1)h1εBit(n)

(2)h2 is an n+1-sequence

(3)h3 is an n-sequence

(4)h4 is an n-sequence

(5)h5 is an n+1-sequence

(6)g = L*h2*<I,R>

(7)h2 = ((h1*<I,R>)#(R*h2))*<L,<I*L,R>>

(8)L*R^n-1*h3 = R*L

(9)h3 = ((R*h3*<I,R>)#h3)*<L,<R*L,R>>

(10)h3 = Copy(L*L,identity,n)*<I,R^n>*h4

(11)Copy(I,zero,n) = Copy(R*L,identity,n)*<I,R^n>*h4

(12)h5 = ((((h3*<I,R>)#h4)*<I,R>)#(R*h5))*<L,<I*L,R>>

(13)f = L*h5*<I,I>

(ii)f is the Godel number of g <=> there are integers n,m and elements g1,h1,h2,h3 such that

(1)f = R^m

(2)g1 is an n-string

(3)h1εBit(n)

(4)h2 is an n+1-string

(5)g1 = L*h2*<I,R>

(6)h2 = ((h1*<I,R>)#(R*h2))*<L,<I*L,R>>

(7)h3 = ((h1*<<L,E>,R>)#(R*h3))*<L,<I*L,R>>

(8)g = L*h3*<I,I>

Let us prove fact (i) first.<=.Suppose that h1,h2,h3,h4,and h5 are as in (1)-(13).Then h1= <Xn-1*L,<....<X0*L,R>...>> for Xiε{L,R}, i=0...n-1 by(1) and h2 = <Xn-1*...*X0*L,<...<X0*L,R>...>> by (2) and (7).Thus g = Xn-1*...*X0 by (6).Now h3 = <rn-1*L,<...<r1*L,<R*L,R>>...>> by (3) and (8). By (9) ri+1=ri*ri for i=0...n-2. Thus h3 = <R^(2^(n-1))*L,<...<R^2*L,<R*L,R>>...>>. Hence by (4) and (10) h4 =

<<R^(2^(n-1)),I>*L,<...<<R,I>*L,R>...>>

so (h3*<I,R>)#h4 = <sn-1*L,<...<s0*L,R>...>> where

$$si = \begin{cases} R^{\wedge}(2^{\wedge}i) \text{ if } ri = L \\ R^{\wedge}0 \qquad \text{if } ri = R. \end{cases}$$

Now by (5) and (12) h5 =

<sn-1*...*s0*L,<...<s0*L,R>...>>

Thus by (13) f = sn-1*...*s0=

$$R^{(bn-1)2^{\wedge}(n-1)+...+(b1)2+b0}$$

where the bi are as above.This completes the proof of <=.For => use the h1,h2,h3,h4,and h5 as in <=.Finally (ii) is proved like (i).

We conclude that the set of pairs (f,g) such that f is the Godel number of g is the projection of the set of solutions to a (3 variable) unification problem. A similar result holds for F[x]. Combining this with paragraph 8 gives the following theorem:

Every RE subset of F is the projection of the set of solutions to a unification problem.A similar result holds for F[x].

12.References

[1]Freyd&Heller
Splitting homotopy invariants
unpublished manuscript
to appear in the festschrift for
Alex Heller

[2]Klop
Combinatory Reduction
Systems
Mathematical Centre Tracts
127
Math. Centrum Amsterdam
1980

[3]Lambek
From lambda calculus to
Cartesian closed categories
in Seldin&Hindley eds.
To H.B.Curry: Essays on
Combinatory Logic, Lambda
Calculus and Formalism
Academic Press 1980
(Curry festschrift)

[4]Matiyasevich
Diophantine representation of
recursively enumerable

predicates
in Fenstad ed.
Proceedings of the Second
Scandanavian Logic Symposium
North Holland 1971

[5]Scott

Relating theories of the
lambda calculus
in the Curry festschrift

[6]Statman

Freyd's heirarchy of
combinator monoids
in
Proceedings Symposium on
Logic in Computer Science
IEEE 1991

[7]

Simply Typed Lambda
Calculus with Surjective
Pairing
CMU Dept. of Math. Research
Report 92-146

[8]

Recursive types and the
subject reduction theorem
CMU Dept. of Math Research
Report 94-164

[9]

A local translation of untyped

lambda calculus into simply typed lambda calculus
CMU Dept. of Math. Research Report 91-134

[10]

Combinators and the theory of partitions
CMU Dept. of Math. Research Report 88-31