Products in a Category with Only One Object
                                    by
                               Rick Statman
                           Carnegie Mellon University
                               May 2020

Abstract: We consider certain decision problems for the free model
of the theory of Cartesian monoids. We introduce a model of computation
based on the notion of a single stack one-way PDA due to Ginsburg,
Greibach and Harrison. This model allows us to solve problems such
as
(1) Given a finite set B of elements and an element F, is F a product
    of members of B?
(2) Is the submonoid generated by the finite set B infinite?
for certain fragments of the free Cartesian monoid. These fragments
include the submonoid of right invertible elements and so our results
apply to the Thompson-Higman groups.

(I) Introduction and preliminaries
     The notion of a Cartesian monoid has been rediscovered many times.
A list of those just known to me would include
Jonsson-Tarski Algebras (Jonsson & Tarski 1961 [6])
Vagabond Groups (Thompson 1971, Higman 1974 [17])
Cantor Algebras (Smirnov 1971 [10])
FP (Backus 1977 [1])
Cartesian Monoids (Lambek & Scott 1980 [7])
Freyd-Heller Monoids (Freyd & Heller 1993 [3])
TOPS (Statman 1996 [12])
Thompson-Higman Monoids (Birget 2007 [2])
CP Monoids (Gray & Pardue 2018 [5]).
The title of this paper has be pirated from Gray & Pardue; the reason for
this will become apparent presently.
     The notion of Cartesian Monoid axiomatizes the idea of a monoid of
functions on a set S supporting a surjective pairing function SxS -> S
lifted pointwise to (S->S) x (S->S) -> (S->S). So a Cartesian Monoid
                    C = (M,*,I,L,R,< >)
is a monoid (M,*,I) together with elements L,R:M and a map < >: MxM -> M
satisfying
L*<F,G> = F   (left projection)
R*<F,G> = G   (right projection)
<F,G>*H = <F*H,G*H>   (pointwise lifting).
<L,R>   = I   (surjectivity)
More generally, one could consider pairing functions which might not
be surjective. In this case, the fourth condition would be omitted. The
authors of the most recent rediscovery call such structures Categorical
Quasiproduct Monoids (CQP). We are happy to credit them with directing
our attention to them.
     Here it will be useful to review some properties of the free
Cartesian Monoid CM. The four conditions have the equivalent rewrite
system
(1) L * <X,Y>   >-> X
(2) R * <X,Y>   >-> Y
(3) <X,Y> * Z   >-> <X * Z, Y * Z>
(4) I * X   >-> X
(5) X * I   >-> X
(6) <L*X,R*X>   >-> X
(7) <L,R>   >-> I
modulo the associativity axioms. This rewrite system is equivalent to
the conditions in the sense that the smallest associative congruence
containing the rules as identities is the congruence generated by the
conditions (axioms). We denote the monotone, reflexive, transitive
closure of >-> by >->>. Here monotone means replacing subexpressions

by their rewrites.

Theorem 1 ([12]):
(1) Every pair of equivalent expressions have a common rewrite.
(2) Every sequence of rewrites eventually terminates in
     a unique normal form.
(3) All of the last three rewrites can be done at the end.
Corollary: The word problem for CM is solvable.
The normal forms of Cartesian monoid expressions have a pleasing
shape. They can be described a binary trees built up from <.>
with strings of L's and R's at the leaves. These strings are
built up by * with I = the empty string. We call these strings
the shifts of the normal form. Normal forms that use only the
rewrites (1)-(5) have a similar shape.

     Now in [12] we proved the following.
Theorem 2: If F and G have distinct normal forms then there exist
          H and K s.t. H*F*K = L and H*G*K = R in CM
This is the "simplicity" rediscoved in Birget [2].

(II) CQ is almost simple
     Let CQ be the free Categorical Quasiproduct Monoid. Clearly
there exists a homomorphism from CQ onto the free Cartesian Monoid.
First, we add to theorem 2 for the case of CQ

Lemma 1;If $F_{0} = F_{1}$ in CM but the (1)-(5) normal forms of
        F_{0} and F_{1} are distinct, say U_{0},U_{1}
        resp.,then there exist H,K s.t. for i= 0 or 1
        $H*U_{i}*K$ >->>(5)-(7)>->> I,   and
        $H*U_{1-i}*K$ >->>(5)-(7)>->> <L,R>.
Proof; By induction on the sum of the lengths of U_{0} and U_{1}.
Basis; since the U_{i} are both in (1)-(5) normal form they cannot both
be shifts. Thus the shortest case has the form U_{i} = I and U_{1-i} =
<U,V>. Now <U,V> >->>(5)-(7) >->> I. Thus U >->>(5)-(7)>->> L and
V >->>(5)-(7)>->> R, and we are done, or U >->>(5)-(7)>->> LX and
V >->>(5)-(7)>->> RX where X must be a shift since U_{1-i} is normal
w.r.t.(1)-(5). But then X = I and we are done again.
Induction Step. In case that one of the U_{i} is a shift is as in the
basis case. Thus we can assume both U_{i} begin with <,>. Let U_{0}=
$<V_{0},V_{1}>$ and U_{1} = $<W_{0},W_{1}>$. Then there exists a j s.t.
V_{j} =/= W_{j}. In case j = 0 we have $L*U_{0}$ >->>(1)>->>V_{0} and
$L*U_{1}$ >->>(1)>->>W_{0} and the induction hypothesis can be applied
to V_{0} and W_{0}.The case j = 1 is similar. End of proof.

Proposition: The homomorphism from CQ onto the free Cartesian
             Monoid is unique and the only non-trivial homomorphism
             of CQ.
Proof; If F and G are not equivalent in CM then there exist H and K
as in theorem 2. By theorem 1 there exists a common rewrite which can
be obtained by rewriting with all of the (5)-(7) rewrites at the end.
Thus there exist U,V s.t.
H*F*H >->> without (5)-(7) >->> U ->>> only (5)-(7) >->> L
H*G*H >->> without (5)-(7) >->> V ->>> only (5)-(7) >->> R
Now if X*I occurs in either U or V it can be elimminated by
X*I >-> X since X cannot contain <,>. Now, neither U nor V can
contain <X,I>, <I,X>, or <L,R> since this would prevent rewriting
to L or R. This holds for any rewrite of U or V by (6). Thus only
the rewrite (6) is used. Hence there exist X,Y such that U = X*L and
V = Y*R (reverse rewrite (3) multiple times).But then I = L*<I,I> =
U*<I,I> = X and I = R*<I,I> = V*<I,I> = Y. So if F an G are identified
by a homomorphism of CQ then so are X and Y.But then by lemma 1 the
homomorphism identifies I and <L,R>.Thus the homomorphism is a
homomorphism of CM. This is impossible by [12] section 3. End of Proof.

(III)G.G.H. PDA's
      Now we would like to introduce a model for computing with CQ
expressions which will allow us to prove many questions about CQ
multiplication decidable. We cannot expect too much since it is
obvious that the existential theory of CQ is undecidable (section 9
of [12]). For what follows we shall refer to (1)-(5) normal form as
CQ normal form.
                      If F is in CQ normal form then the shift S which
when read from left to right describes the position of the leaf of
the tree of F where the shift S' resides then the CQ normal form of
S*F is S' but for no initial segment S" of S is the normal form of
S"*F a shift. Our model is the G.G.H.[4] notion of a one way single
stack pushdown automata. These are non-deterministic PDA's which can
scan the current stack by a two way NDFA before reading the top most
stack symbol, executing a stack operation,changing state and reading
the next input.The input alphabet consists of CQ expressions F in
normal form taken from a give finite set, and input from left to right.
When the initial contents of the stack is the string S, with
                top to bottom = right to left,
and the input is $F_{1},...,F_{t}$ then we want the contents of the
stack to be the normal form of $S*F_{1}*...*F_{t}$.However this may not
be a shift. In this case we terminate the computation in failure.
The PDA operates as follows
(1) It reads the input F
(2) It reads the stack to determine if it has the form S'*S where the
    normal form of S*F is a shift S". If not the computation fails.
(3) If (2) succeeds it pops S from the stack and pushes S" onto the stack.
Here we note that in (2) there are only finitely many S to check so
this can be implemented in a deterministic G.G.H. . Now G.G.H proved
that the sets accepted by G.G.H PDA;'s are closed under intersection
and union but not complement. The deterministic one are closed under
complement but not intersection.

Theorem 3. Let B be a finite set of CQ expressions. Then it is decidable
          if F is a product of members of B.
Proof. We assume that all expression are in CQ normal form. First,
let $S_{1},..., S_{n}$ be the complete list of shifts s.t. $S_{i}*F$
= a shift S'_{i} but no initial segment of S_{i}(right to left) does.
We construct a deterministic G.G.H. machines M_{i},N_{i}, for i=1,...,n,
so that M_{i} accepts all inputs $F_{1}...F_{k}$ s.t. $S_{i}*F_{1}*...*F_{k}$
= S'_{i}. For each i = 1,...,n let $S"_{i}$ be S_{i} with its last L or
R (from right to left) deleted. Then N_{i} is constructed to accept all
inputs $F_{1}...F_{k}$ s.t. the CQ normal form of $S"_{i}*F_{1}*...*F_{k}$
is not a shift . Here we use closure of deterministic machines under
complements. Now by [4] there exists a non-deterministic G.G.H. machine
M which accepts the intersection of the sets of inputs accepted by the
all machines M_{i} and N_{i} for i=1,...,n. Now the decision theorems
of G.G.H. do not apply to non-deterministic G.G.H. PDA's. However, the
method of La'Torre [8] does apply. The machine M can be representsed in
Rabin's theory WS2S [9] and tested for emptiness. End of proof.
      Many CQ decision problems can be solved with this method. For
example, it is decidable whether the set of CQ distinct products
of members of B is infinite. This will be seen in the next section.

(IV) More applications of G.G.H. PDA's
      There is a well known duality between rooted binary trees with
n leaves and triangulations of an n-gon. In the CQ case the leaves
come equiped with shifts. These shifts control the results of further
composition. The nice geometrical questions about tiling the plane
with n-gons are analogous to questions about covering Cantor Space
by normal CQ expressions.

Definition: Let B be a finite set of CQ normal forms. We say that B
          covers Cantor space if there exists an infinite sequence

F_{1},...,F_{n},... of members of B s.t. for each shift
              S there exists an n s.t. the normal form of
                           S*F_{1}*...*F_{n}
              is not a shift.
Example: {<I,I>} covers Cantor space but {< R,L>} does not.

Definition: A shift S is said to be bad for B if for any sequence
F_{1},...,F_{n} of members of B the CQ normal form of
                      S*F_{1}*...*F_{n}
is a shift.

          For example R*R is bad for { <L,R*R> }. First we observe
that the set of bad B shifts is recursive uniformly in B. This can be
seen by constructing a deterministic G.G.H. one way stack machine
which, if started with S in its stack, accepts all inputs if and only
if S is bad. The construction of the machine is uniform in S, so that
the set of S which are bad is definable in Rabin's theory WS2S by
LaTorre's method. To recall, the input alphabet is the set B and the
machine runs as follows.With input symbol F the machine reads the top
of the current stack looking for a minimal S s.t. the normal form of
S*F is a shift. This can be done by a DFSA. Having found such an S the
machine pops S and pushes the normal form of S*F. Otherwise, the machine
rejects the entire input.

     Before considering coverings we settle an algebraic question.
Theorem 4 : It is decidable whether the submonoid generated by B is
             infinite.
Proof: Of course, if B contains a shift then the submonoid generated
by B is infinite, so we can assume this is not the case. First, we
say that an S bad for B is extenuative if for each n there exists
F_{1},...,F_{k} in B s.t. the CQ normal form of S*F_{1}*...*F_{k}
has length at least n. Now it is decidable whether S is extenuative.
First decide whether S is bad for B. Now consider the set of natural
numbers encoded as strings of L's and the input language B + {L}.
Design a deterministic G.G.H. PDA with S initially in its stack and
which accepts an input W if and only if W has the form
                 F_{1}...F_{k}L...L
                                -----
                             n occurrences
and where after reading F_{1}...F_{k} the stack has length at least n.
Now the set of words L...L s.t. F_{1}...F_{k}L...L is accepted by
the machine is the result of applying a sequential transducer to
the set of words F_{1}...F_{k}L...L accepted. Thus by theorem 2.4
of [4] we can construct a non-deterministic G.G.H PDA which accepts
exactly this set of words L...L. Thus by La'Torre's method we can
decide if this set is all strings of L's. Next pick an initial
member F of B and one of its shifts S. We construct a G.G.H. non-
deterministic PDA as follows. The input alphabet consists of B plus
a new letter @ (we could have used L as above). On inputs of the
form F_{1}...F_{k}@^{m} the machine proceeds as above except when
the current stack is @^{p}S' and the input is F_{i} s.t. the CQ
normal form of S'*F_{i} is not a shift. In this case, the machine
guesses a minimal length shift S" s.t. the normal form S"*S'*F_{i}
is a shift, say S+, and updates the stack to @^{p+1}S+. The machine
accepts if in the final stack the number of @ is not less than m.
By G.G.H. theorem 2.4 the set of @^{m} s.t. there is a
                      F_{1}...F_{k}@^{m}
accepted by the machine is accepted by a non-deterministic G.G.H
PDA. Now it is decidable if this set is all @^{m}. We distinguish
two cases.
Case 1. The set is all @^{m}. Then there are arbitrarily large
CQ normal form generated by B and submonoid generated by B is
infinite.
Case 2 The set of all @^{m} is finite for all F in B and S. Then

every infinite sequence F_{1},...,F_{n},...of members of B must
have an initial segments F_{1},...,F_{k} s.t. all the shifts of
the CQ normal form of F_{1}*...*F_{k} are bad. So by Konig's
lemma there exists a finite tree of such finite sequences s.t.
every path has an initial segment in this tree. Now search until
such a finite tree T is found. Now suppose that the CQ normal
form of F_{1}*...*F_{k} has a bad shift S which is extenuative.
Now F_{1},...,F_{k} has an initial segment in T say F_{1},...,F_{p}.
So F_{1}*...*F_{p} has a bad shift with the same property. Now the
submonoid generated by B will be finite <=> no such extenuative
shift exists. Thus it suffices to test all the shifts of products
in the tree for extenuativeness.
End of Proof.

Theorem 5 : It is decidable whether B covers Cantor Space
Proof; Given B it is impossible to cover Cantor Space if there
is a bad shift i.e. a shift s.t. for all F_{1}...F_{n} : B the
normal form of S*F_{1}*...*F_{n} is a shift. This is decidable
by previous arguement. Now if no bad shift exists then for every
shift S there exits F_{1}...F_{n} : B s.t. the normal form of
S*F_{1}*...*F_{n} is not a shift. Thus B can cover Cantor Space
by repeatedly applying the F_{1}*...*F_{n} as shifts S appear
in the normal form of previous applications. End of Proof.
We conclude with an amusing observation.
(V) Gigsaw puzzles
     A gigsaw puzzle is a patern matching problem where each variable
occurs at most once and solutions come from a fixed set of CQ expressions
all of which must be used.Here we show the problem is NP complete.
     We encode the satisfiability problem. We assume
that we are given a conjunctive normal form (conjunction of disjunctions;
we regard x v x v y as distinct from x v y). We supposethat the variables
are x_{1} ... x_{n}. For each variable x_{i} we construct two gadgets
G_{i} and H_{i} by

G_{i} = <<I,I>,<<I,I>,< ... <L,<<I,I>,< ... <<I,I>,I> ...>>> ... >>>
            ------------------   -----------------
             i-1 occurrences   n-i+1 occurrences
H_{i} = <<I,I>,<<I,I>,< ... <R,<<I,I>,< ... <<I,I>,I> ...>>> ... >>>
            ------------------   ----------------
             i-1 occurrences   n-i+1 occurrences

Now suppose that we have a conjunct C of the form
        x_{a(1)} V .. V x_{a(k)} V ~x_{b(1)} V ... V ~x_{b(m)}
We replace each of the k+m variable occurrences by a new variable y_{i}
and we construct a product C# of terms
L*R^{a(i)} * y_{i} * <I,<I,I>>   if i<k+1
L*R^{b(k-i)} * y_{i} * <<I,I>,I>   if k<i
and the identity C# = I. Now assume that x_{i} occurs m times. We
introduce m new variables z_{1} ... z_{m} and define the term B
in m stages as follows
B_{1}   = L*z_{1}
B_{j+1} = B_{j}*z_{j}*<<L,<I,I>>,<<I,I>,R>>
B = B_{m}
and we add the identity B*<I,I> = I. This is a total number of identities
equal to the number of conjuncts plus the number of variables. Then
the identities are solvable using the gadgets <=> the original conjunction
is satisfiable.

(VI) What next?
     It is interesting to see if these methods can be extended to
the free Cartesian monoid. The questions one wants to ask about the
corresponding non-detreministic G.G.H. machines do not seem to be
answerable in any straightforward manner.However, we can make some
direct applications. Let Let RI be the submonoid of right invertible

elements of CM. Here, we let B be a finite subset of RI and F an element of RI, all in (1)-(7) normal form.
Lemma 2: If F is in the submonoid generated by B then there exists
$F_{1},...,F_{n}$ in B s.t.
$$F_{1}*...*F_{n} \text{ >->> } (1)-(5) \text{ >->> } F$$
except for F = I.
Proof; first recall the characterization of the elements in RI given in [12]. F is in RI <=> no shift of F is a final segment (left to right) of another shift of F. Now suppose that
$$F_{1}*...*F_{n} \text{ >->> } (1)-(7) \text{ >->> } F.$$
By theorem 1 (3) there exists G s.t.
$$F_{1}*...*F_{n} \text{ >->> } (1)-(5) \text{ >->> } G \text{ >->> } (5)-(7) \text{ >->> } F.$$
   Note that in the rewrite from G to F all the applications of (6) are to X a shift. Now let J be the kernel of the unique homomorphism of CQ on CM. J is precisely the set of all H in CQ s.t. H = I in CM. Now consider the shape of G. The binary tree of G begins with the tree of F but where a shift S of F would occur is the CQ normal form of a member of J * S. Note that such an S is not I since F is in RI, but there is the trivial case when the member of J is I. Now each member H of J has the following structure. Each leaf of the binary tree of H can be described by a sequence of L's and R's read from right to left. The corresponding shift S' has the property that the normal form of S'* H is precisely S' and no final segment (left to right) S" of S' has the property that S"*H = a shift in CQ. Now let H be the member of J for S a shift of F. We have for each shift S' of H s.t. if S is in position S", S'*S"*G = S'*S in CQ.
   Now, for each i=1,...,n, S"*S'*F_{1}*...*F_{i} = a shift in CQ. However, since all the F_{j} are in RI there is a unique shortest S'*S with this property s.t. S'*S"*G = S'*S in CQ. But if H is non trivial then there are at least two such. Thus H is trivial. Hence G = F. End of proof.

Corollary; Theorems 3,4,5 apply to RI, and thus to the Thompson-Higman groups.

(VII) References

[1]Backus,J. Can programming be liberated from the von
           Neumann style?
           Communications of the ACM,
           Vol 21, No 8, August 1978
[2] Birget J-C. Monoid generalization of the Richard Thompsom groups
            Journal of Pure and Applied Algebra
            13, 2009 pps 264-278
[3]Freyd P. & Heller A. Spliting homotopy invariants II
                  Journal of Pure and Applied Algebra
                  89,1993.pps 93-195
[4]Ginsburg S.,Greibach S., Harrison M. One way stack automata
                  J.A.C.M. Vol 14, No. 2, April 1967
                  pps 389-418
[5]Gray A. & Pardue K. Products in a category with one object
                  https://arxiv.org/abs/1604.03999
[6]Jonsson B. & Tarski A. On two properties of free algebras
                   Math. Scand. 9, 1961, pps 95-101
[7]Lambek J. From lambda calculus to cartesian closed categories
           To H.B. Curry; Essays in Combinatory Logic...
           Seldin J. & Hindley R. eds, Academic Press 1980
[8] LaTorre S., Manhusadan P.,& Parlato G. A robust class of
                                      context sensitive
                                      languages
              22nd IEEE Symposium on Logic in Computer Science
              July 2007, Wroclaw, Poland,
[9]Rabin M. Decidability of second order theories and automata on
           infinite trees
           Transactions of the American Mathematical Society

                    Vol. 141 , July 1969, pps. 1-35
[10]Smirnov D.M. Cantor algebras with one generator
                    Algebra and Logic 10, 1971, pps 40-49
[11]Statman R.
                    Simply typed lambda calculus with surjective
                    pairing,
                    CMU Department of Mathematics Research Report
                    92-164, 1992
[12]   On Cartesian monoids,
                    Proceedings of Computer Science Logic 1996.
                    Springer Lecture Notes in Computer Sciences
                    1258, Springer-Verlag, 1996, pps 446-459
[13]   Surjective pairing revisited
                    Liber Amicorum for Roel deVrijer,
                    Klop J.W., van Oostrom J., van Raamsdonk F. eds,
                    University of Amsterdam, 2009
[14]   Cartesian monoids
                    Electronic Notes in Theoretical Computer Science
                    265, Elsevier 2010 pps 437-451
[15]   Representation of Cartesian monoids in and around
                    the piecewise shift operators
                    to appear in
                    Mathematical Structures in Computer Science
[16]   Minimal Backus FP is Turing complete
                    Higher Dimensional Rewriting and Algebra
                    July 7 2018, Oxford
[17]Thompson R., Embeddings into finitely generated simple groups
                    which preserve the word problem
                    in Adian S.,Boone W. & Higman G. (eds)
                    Word Problems
                    North Holland, 1980, pps 401-444