

# Products in a Category with Only One Object

Rick Statman

Carnegie Mellon University

rs31@webmail.math.cmu.edu

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.

## 1 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 [6], Vagabond Groups [12], Cantor algebras [9], FP [1], Cartesian Monoids [7], Freyd-Heller Monoids [3], TOPS [11], Thompson-Higman Monoids [2], and CP Monoids [5]. The title of this paper has been 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  $S \times S \rightarrow S$  lifted pointwise to  $(S \rightarrow S) \times (S \rightarrow S) \rightarrow (S \rightarrow S)$ . So a Cartesian Monoid  $C = (M, *, I, L, R, \langle - \rangle)$  is a monoid  $(M, *, I)$  together with elements  $L, R : M$  and a map  $\langle - \rangle : M \times M \rightarrow M$  satisfying:

$$\begin{aligned} L * \langle F, G \rangle &= F && \text{(left projection)} \\ R * \langle F, G \rangle &= G && \text{(right projection)} \\ \langle F, G \rangle * H &= \langle F * H, G * H \rangle && \text{(pointwise lifting)} \\ \langle L, R \rangle &= I && \text{(surjectivity)} \end{aligned}$$

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) [4, Section 2.4]. 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)  $L * \langle X, Y \rangle \rightarrow X$
- (2)  $R * \langle X, Y \rangle \rightarrow Y$
- (3)  $\langle X, Y \rangle * Z \rightarrow \langle X * Z, Y * Z \rangle$
- (4)  $I * X \rightarrow X$
- (5)  $X * I \rightarrow X$

$$(6) \langle L * X, R * X \rangle \rightsquigarrow X$$

$$(7) \langle L, R \rangle \rightsquigarrow I$$

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  $\rightsquigarrow$  by  $\rightsquigarrow^*$ . Here monotone means replacing subexpressions by their rewrites.

**Theorem 1** (See [11]). (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 as binary trees built up from  $\langle - \rangle$  with strings of  $L$ 's and  $R$ 's at the leaves. These strings are built up by  $*$  with  $I$  as 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 [11] 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” rediscovered in Birget [2].

## 2 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  $H, K$  such that for  $i = 0$  or  $i = 1$  we have:

$$\begin{aligned} H * U_i * K &\rightsquigarrow_{5,6,7}^* I \\ H * U_{1-i} * K &\rightsquigarrow_{5,6,7}^* \langle L, R \rangle \end{aligned}$$

*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  $U_{1-i} = \langle U, V \rangle$ . Now  $\langle U, V \rangle \rightsquigarrow_{5,6,7}^* I$ . Thus  $U \rightsquigarrow_{5,6,7}^* L$  and  $V \rightsquigarrow_{5,6,7}^* R$ , and we are done, or  $U \rightsquigarrow_{5,6,7}^* LX$  and  $V \rightsquigarrow_{5,6,7}^* RX$  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  $\langle - \rangle$ . Let  $U_0 = \langle V_0, V_1 \rangle$  and  $U_1 = \langle W_0, W_1 \rangle$ . Then there exists a  $j$  s.t.  $V_j \neq W_j$ . In case  $j = 0$  we have  $L * U_0 \rightsquigarrow (1) \rightsquigarrow V_0$  and  $L * U_1 \rightsquigarrow (1) \rightsquigarrow W_0$  and the induction hypothesis can be applied to  $V_0$  and  $W_0$ . The case  $j = 1$  is similar.  $\square$

**Proposition 1.** 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.:

$$\begin{aligned} H * F * H &\gg_{1,2,3,4} U \gg_{5,6,7} L \\ H * G * H &\gg_{1,2,3,4} V \gg_{5,6,7} R \end{aligned}$$

Now if  $X * I$  occur in either  $U$  or  $V$  it can be eliminated by  $X * I \rightarrow X$ , since  $X$  cannot contain  $\langle - \rangle$ . Now, neither  $U$  nor  $V$  can contain  $\langle X, I \rangle$ ,  $\langle I, X \rangle$  or  $\langle L, R \rangle$  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  $I = L * \langle I, I \rangle = U * \langle I, I \rangle = X$  and  $I = R * \langle I, I \rangle = V * \langle I, I \rangle = Y$ . So if  $F$  and  $G$  are identified by a homomorphism of CQ then so are  $X$  and  $Y$ . But then by Lemma 1 the homomorphism identifies  $I$  and  $\langle L, R \rangle$ . Thus the homomorphism is a homomorphism of CM. This is impossible by [11, Section 3].  $\square$

### 3 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 [11, Section 9]. 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. notion [4] of a one-way single stack pushdown automata. These are non-deterministic PDAs which can scan the current stack by a two way N DFA 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 given finite set, and input from left to right.

When the initial contents of the stack are the string  $S$ , with top-to-bottom corresponding to 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  $F_1 * \dots * F_t$ . 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. PDAs 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 is in CQ normal form. First, let  $S_1, \dots, S_n$  be the complete list of shifts s.t.  $S_i * F$  equals a shift  $S_i$ , but no initial segment of  $S_i$  (right to left) does.

We construct deterministic G.G.H. machines  $M_i, N_i$ , for  $i = 1, \dots, n$ , so that  $M_i$  accepts all inputs  $F_1, \dots, F_k$  s.t.  $S_i * F_1 * \dots * F_k = S_i$ . For each  $i = 1, \dots, n$  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  $F_1, \dots, F_k$  s.t. the CQ normal form of  $S_i'' * F_1 * \dots * F_k$  is not a shift. Here we use closure of deterministic machines under complements. Now by the work of G.G.H. [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  $i = 1, \dots, n$ . Now the decision theorems of G.G.H. do not apply to non-deterministic G.G.H. PDAs. However, the method of La Torre [13] does apply. The machine  $M$  can be represented in Rabin's theory WS2S [8] and tested for emptiness.  $\square$

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.

## 4 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 equipped 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 1.** Let  $B$  be a finite set of CQ normal forms. We say that  $B$  covers Cantor space if there exists an infinite sequence  $F_1, \dots, F_n, \dots$  of members of  $B$  s.t. for each shift  $S$  there exists an  $n$  s.t. the normal form of  $S * F_1 * \dots * F_n$  is not a shift.

As an example, the set  $\{ \langle I, I \rangle \}$  covers Cantor space but  $\{ \langle R, L \rangle \}$  does not.

**Definition 2.** A shift  $S$  is said to be *bad* for  $B$  if for any sequence  $F_1, \dots, F_n$  of members of  $B$  the CQ normal form of  $S * F_1 * \dots * F_n$  is a shift.

For example,  $R * R$  is bad for  $\{ \langle L, R * R \rangle \}$ . 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  $F_1, \dots, F_k$  in  $B$  s.t. the CQ normal form of  $S * F_1 * \dots * F_k$  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  $F_1 \dots F_k L \dots L$  (with  $n$  occurrences of  $L$ ), and where after reading  $F_1 \dots F_k$  the stack has length at least  $n$ .

Now the set of words  $L \dots L$  s.t.  $F_1, \dots, F_k L \dots L$  is accepted by the machine is the result of applying a sequential transducer to the set of words  $F_1, \dots, F_k L \dots L$  accepted. Thus by [4, Theorem 2.4] 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. nondeterministic 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  $@_pS'$  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  $F_1, \dots, F_k@_m$  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  $F_1, \dots, F_n, \dots$  of members of  $B$  must have an initial segments  $F_1, \dots, F_k$  s.t. all the shifts of the CQ normal form of  $F_1 * \dots * F_k$  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  $F_1 * \dots * F_k$  has a bad shift  $S$  which is extenuative. Now  $F_1, \dots, F_k$  has an initial segment in  $T$  say  $F_1, \dots, F_p$ . So  $F_1 * \dots * F_p$  has a bad shift with the same property. Now the submonoid generated by  $B$  will be finite if and only if no such extenuative shift exists. Thus it suffices to test all the shifts of products in the tree for extenuativeness.  $\square$

**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  $F_1 \dots F_n : B$  the normal form of  $S * F_1 * \dots * F_n$  is a shift. This is decidable by previous argument. Now if no bad shift exists then for every shift  $S$  there exists  $F_1 \dots F_n : B$  s.t. the normal form of  $S * F_1 * \dots * F_n$  is not a shift. Thus  $B$  can cover Cantor space by repeatedly applying the  $F_1 * \dots * F_n$  as shifts  $S$  appear in the normal form of previous applications.  $\square$

We conclude with an amusing observation.

## 5 GIGSAW PUZZLES

A gigsaw puzzle is a pattern 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  $x \vee x \vee y$  as distinct from  $x \vee y$ ). We suppose that the variables are  $x_1, \dots, x_n$ . For each variable  $x_i$  we construct two gadgets  $G_i$  and  $H_i$  by

$$G_i = \underbrace{\langle \langle I, I \rangle, \langle \langle I, I \rangle, \langle \dots \langle L, \langle \langle I, I \rangle, \langle \dots \langle \langle I, I \rangle, I \rangle \dots \rangle \rangle \dots \rangle \rangle \rangle}_{i-1 \text{ occurrences}} \quad \underbrace{\langle \dots \langle \langle I, I \rangle, \langle \dots \langle \langle I, I \rangle, I \rangle \dots \rangle \rangle \dots \rangle}_{n-i+1 \text{ occurrences}}$$

$$H_i = \underbrace{\langle \langle I, I \rangle, \langle \langle I, I \rangle, \langle \dots \langle R, \langle \langle I, I \rangle, \langle \dots \langle \langle I, I \rangle, I \rangle \dots \rangle \rangle \dots \rangle \rangle}_{i-1 \text{ occurrences}} \quad \underbrace{\langle \dots \langle \langle I, I \rangle, \langle \dots \langle \langle I, I \rangle, I \rangle \dots \rangle \rangle \dots \rangle}_{n-i+1 \text{ occurrences}}$$

Now suppose that we have a conjunct  $C$  of the form

$$x_{a(1)} \bigvee \dots \bigvee x_{a(k)} \bigvee \neg x_{b(1)} \bigvee \dots \bigvee \neg x_{b(m)}$$

We replace each of the  $k + m$  variable occurrences by a new variable  $y_i$  and we construct a product  $C\#$  of terms

$$\begin{aligned} L * R^{a(i)} * y_i * \langle I, \langle I, I \rangle \rangle & \quad \text{if } i < k + 1 \\ L * R^{b(k-i)} * y_i * \langle \langle I, I \rangle, I \rangle & \quad \text{if } k < i \end{aligned}$$

and the identity  $C\# = I$ . Now assume that  $x_i$  occurs  $m$  times. We introduce  $m$  new variables  $z_1, \dots, z_m$  and define the term  $B$  in  $m$  stages as follows

$$\begin{aligned} B_1 &= L * z_1 \\ B_{j+1} &= B_j * z_j * \langle \langle L, \langle I, I \rangle \rangle, \langle \langle I, I \rangle, R \rangle \rangle \\ B &= B_m \end{aligned}$$

and we add the identity  $B * \langle I, I \rangle = I$ . 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 if and only if the original conjunction is satisfiable.

## 6 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-deterministic G.G.H. machines do not seem to be answerable in any straightforward manner. However, we can make some direct applications. Let  $\text{RI}$  be the submonoid of right invertible elements of  $\text{CM}$ . Here, we let  $B$  be a finite subset of  $\text{RI}$  and  $F$  an element of  $\text{RI}$ , all in (1)–(7) normal form.

**Lemma 2.** *If  $F$  is in the submonoid generated by  $B$  then there exists  $F_1, \dots, F_n$  in  $B$  s.t.*

$$F_1 * \dots * F_n \rightsquigarrow_{1,2,3,4,5} F$$

except for  $F = I$ .

*Proof.* First recall the characterization of the elements in  $\text{RI}$  given in [11].  $F$  is in  $\text{RI}$  if and only if no shift of  $F$  is a final segment (left to right) of another shift of  $F$ . Now suppose that  $F_1 * \dots * F_n \rightsquigarrow_{1,2,3,4,5,6,7} F$ . By Theorem 1 (3) there exists  $G$  s.t.  $F_1 * \dots * F_n \rightsquigarrow_{1,2,3,4,5} G \rightsquigarrow_{5,6,7} F$ . Note that in the rewrite from  $G$  to  $F$  all the applications of (6) are to cancel a shift. Now let  $J$  be the kernel of the unique homomorphism of  $\text{CQ}$  on  $\text{CM}$ .  $J$  is precisely the set of all  $H$  in  $\text{CQ}$  s.t.  $H = I$  in  $\text{CM}$ . 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  $\text{CQ}$  normal form of a member of  $J * S$ . Note that such an  $S$  is not  $I$  since  $F$  is in  $\text{RI}$ , 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  $S'' * H$  equals a shift in  $\text{CQ}$ . 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  $S''$ , then  $S' * S'' * G = S' * S$  in CQ. Now, for each  $i = 1, \dots, n$ ,  $S' * S' * F_1 * \dots * F_i$  equals 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 not trivial then there are at least two such. Thus  $H$  is trivial. Hence  $G = F$ .  $\square$

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

## References

- [1] J. Backus (1978): *Can programming be liberated from the von Neumann style?* *Communications of the ACM* 21(21), doi:10.1007/978-3-662-09507-2\_10.
- [2] J.-C. Birget (2009): *Monoid generalization of the Richard Thompson groups.* *Journal of Pure and Applied Algebra* 13, pp. 264–278, doi:10.1016/j.jpaa.2008.06.012.
- [3] P. Freyd & A. Heller (1993): *Splitting homotopy idempotents II.* *Journal of Pure and Applied Algebra* 89, pp. 93–195, doi:10.1016/0022-4049(93)90088-b.
- [4] S. Ginsburg, S. Greibach & M. Harrison (1967): *One-way stack automata.* *Journal of the ACM* 14, pp. 389–418, doi:10.1145/321386.321403.
- [5] A. Gray & K. Pardue (2016): *Products in a category with one object.* arXiv:1604.03999.
- [6] B. Jonsson & A. Tarski (1961): *On two properties of free algebras.* *Mathematica Scandinavica* 9, pp. 95–101, doi:10.7146/math.scand.a-10627.
- [7] J. Lambek (1980): *H. B. Curry, Essays in Combinatory Logic.* Academic Press.
- [8] M. Rabin (1969): *Decidability of second order theories and automata on finite trees.* *Transactions of the American Mathematical Society* 141, pp. 1–35, doi:10.2307/2272788.
- [9] D. Smirnov (1971): *Cantor algebras with one generator.* *Algebra and Logic* 10, pp. 40–49, doi:10.1007/bf02217801.
- [10] R. Statman (1992): *Simply typed lambda calculus with surjective pairing.* *CMU Department of Mathematics Research Report*, pp. 92–164.
- [11] R. Statman (1996): *On Cartesian monoids.* *Springer Lecture Notes in Computer Science* 1258, pp. 446–459, doi:10.1007/3-540-63172-0\_55.
- [12] R. Thompson (1980): *Word Problems*, chapter Embeddings into finitely generated simple groups which preserve the word problem, pp. 401–444. North Holland, doi:10.1016/S0049-237X(08)71348-X.
- [13] S. La Torre, P. Manhusadan & G. Parlato (2007): *A robust class of context sensitive languages.* *Proceedings of 22nd IEEE Symposium on Logic in Computer Science*, doi:10.1109/lics.2007.9.