



**HAL**  
open science

## Using groups for investigating rewrite systems

Patrick Dehornoy

► **To cite this version:**

| Patrick Dehornoy. Using groups for investigating rewrite systems. 2006. hal-00095690

**HAL Id: hal-00095690**

**<https://hal.science/hal-00095690>**

Preprint submitted on 18 Sep 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# USING GROUPS FOR INVESTIGATING REWRITE SYSTEMS

PATRICK DEHORNOY

ABSTRACT. We describe several technical tools that prove to be efficient for investigating the rewrite systems associated with a family of algebraic laws, and might be useful for more general rewrite systems. These tools consist in introducing a monoid of partial operators, listing the monoid relations expressing the possible local confluence of the rewrite system, then introducing the group presented by these relations, and finally replacing the initial rewrite system with a internal process entirely sitting in the latter group. When the approach can be completed, one typically obtains a practical method for constructing algebras satisfying prescribed laws and for solving the associated word problem.

## INTRODUCTION

Let  $\mathcal{L}$  be a family of algebraic laws, typically associativity, commutativity, distributivity, *etc.*, involving a certain signature  $\mathcal{F}$ , and let  $\mathcal{X}$  be an infinite set of variables. By associating with every law  $l = r$  of  $\mathcal{L}$  the two rules  $l \rightarrow r$  and  $r \rightarrow l$ , we obtain a rewrite system  $R_{\mathcal{L}}$  on the family of all  $\mathcal{F}$ -terms constructed on  $\mathcal{X}$ . The aim of this paper is to present a general method for investigating the rewrite systems of the form  $R_{\mathcal{L}}$  by introducing an associated monoid or group. This approach proved to be useful for various systems  $R_{\mathcal{L}}$ , and it might be relevant for more general rewrite systems, typically those arising in a context of algebra [6, 15].

The approach comprises three steps. The first one consists in associating with the rewrite system  $R_{\mathcal{L}}$  a certain inverse monoid  $\mathcal{Geom}_{\mathcal{L}}$  of partial operators by taking into account where and in which direction the rules are applied; this monoid is called the *geometry monoid* for  $\mathcal{L}$  as it captures several specific geometrical phenomena specific to  $\mathcal{L}$ .

The second step consists in replacing the inverse monoid  $\mathcal{Geom}_{\mathcal{L}}$  with a group  $\mathbf{Geom}_{\mathcal{L}}$  that resembles it: when the laws of  $\mathcal{L}$  are simple enough, typically when no variable is repeated more than once in each side of the laws, the group  $\mathbf{Geom}_{\mathcal{L}}$  can be defined to simply be the universal group of  $\mathcal{Geom}_{\mathcal{L}}$ ; in more complicated cases, a convenient group can be obtained by investigating the local confluence relations holding in  $\mathcal{Geom}_{\mathcal{L}}$  and defining  $\mathbf{Geom}_{\mathcal{L}}$  to be the group presented by these relations.

The third—and technically main—step is an internalization process that replaces the external action of  $\mathcal{Geom}_{\mathcal{L}}$  on terms by an internal multiplication in  $\mathbf{Geom}_{\mathcal{L}}$ .

When all three steps can be completed, various questions about the laws  $\mathcal{L}$  can be successfully addressed, typically constructing algebras that satisfy these laws, or solving the word problem of  $\mathcal{L}$ , *i.e.*, designing an algorithm that decides whether two terms are equivalent under the congruence generated by  $R_{\mathcal{L}}$ .

---

1991 *Mathematics Subject Classification.* 68Q01, 68Q42, 20N02.

*Key words and phrases.* algebraic law, term rewrite system, word problem, confluence.

In this paper, we shall present the approach in a general setting and mention some of the existing examples. However, in order to make this paper more than a survey, we shall put the emphasis on a new example for which the method had not been considered before, namely the case of the augmented left self-distributivity laws, or **ALD**-laws, defined to be the following three laws

$$(ALD) \quad \begin{cases} x * (y * z) = (x * y) * (x * z), \\ x * (y \circ z) = (x * y) \circ (x * z), \\ x * (y * z) = (x \circ y) * z. \end{cases}$$

These laws and the algebras that satisfy them, called **ALD-algebras** in this paper, appeared recently in several frameworks [12, 13, 14], and they had never been addressed from the point of view of the geometry monoid. We shall see that a number of technical questions remain open in this seemingly difficult case. Nevertheless, the method is sufficient to naturally lead to the construction of a (highly non-trivial) example of such an **ALD**-algebra. Technically, the main step is the construction of what is called a blueprint for the **ALD**-laws.

The leading principle underlying the approach is to use the geometry monoid to *guess* some formulas, and then to check the latter by a direct verification. Typically, concentrating on the possible confluence relations that hold in the geometry monoid  $\mathcal{Geom}_{\mathcal{L}}$  and using these relations for introducing a group  $\mathbf{Geom}_{\mathcal{L}}$  relies on a heuristic approach rather than on any uniform proved argument. Now the point is that this approach, which might seem loose, actually works in some definitely non-trivial cases: thus what legitimates the scheme is not some general a priori argument, but rather its a posteriori success.

The paper is organized as follows. The geometry monoid  $\mathcal{Geom}_{\mathcal{L}}$  is introduced in Section 1. The process for going from the monoid  $\mathcal{Geom}_{\mathcal{L}}$  to a group  $\mathbf{Geom}_{\mathcal{L}}$  is described in Section 2. The principle of internalizing terms in the geometry monoid/group is explained in Section 3. Next, we show in Section 4 how the general study developed in Sections 2 and 3 can be used, in good cases, to investigate a family of algebraic laws, with a special emphasis on the example of **ALD**. Finally, in Section 5, we briefly address the question of recognizing whether  $\mathbf{Geom}_{\mathcal{L}}$  is a group of fractions, which amounts to investigating whether a locally confluent system is actually confluent: we insist on some methods for doing that even in a non-noetherian framework, *i.e.*, when the standard methods fail.

## 1. THE GEOMETRY MONOID $\mathcal{Geom}_{\mathcal{L}}$

The first step in our study consists in analysing a rewrite system by means of a monoid of partial operators. The general idea is as follows. If  $R$  is a rewrite system on some set  $T$ , then, in the most general situation, several rules may be applied to a given element  $t$  of  $T$ , and, on the other hand, not every rule need to apply to  $t$ , so the action of  $R$  on  $T$  cannot be described by functional operators in a natural way. However, by fixing the value of enough parameters, we can always discard the lack of uniqueness and describe  $R$  in terms of partial functional operators.

Here we shall apply this scheme in the case when  $T$  is a set of terms and  $R$  consists of rules that can be applied to various subterms, more precisely when  $R$  is the rewrite system associated with a family of algebraic laws. In this case, we shall obtain uniqueness, hence functionality, whenever the rule and the position are fixed.

**1.1. Applying an algebraic law.** For  $\mathcal{F}$  a signature consisting of operation symbols exclusively, and  $\mathcal{X}$  a nonempty set of variables, we denote by  $\mathbf{Term}_{\mathcal{F},\mathcal{X}}$  the family of all terms built using operation symbols from  $\mathcal{F}$  and variables from  $\mathcal{X}$ . Practically, we shall assume that some infinite list of variables has been fixed, and drop the reference to  $\mathcal{X}$ , thus writing  $\mathbf{Term}_{\mathcal{F}}$  for  $\mathbf{Term}_{\mathcal{F},\mathcal{X}}$ . An algebraic law is a pair of terms  $\{l, r\}$ —usually denoted as  $l = r$ . We shall speak of an *ordered* algebraic law to insist that the pair  $(l, r)$  is ordered, *i.e.*, a distinguished orientation is chosen. For instance, there are two oriented versions for the standard associativity law  $x \cdot (y \cdot z) = (x \cdot y) \cdot z$ , namely the pair  $(x \cdot (y \cdot z), (x \cdot y) \cdot z)$ , and the symmetric pair. Note that, as the commutativity law is  $x \cdot y = y \cdot x$  is involutive, there is, up to a substitution of variables, only one oriented version.

In this context, the fundamental operation of *applying* an algebraic law to a term corresponds to a rewrite system: applying  $l = r$ —*i.e.*,  $\{l, r\}$ —to some term  $t$  means replacing some subterm  $t_0$  of  $t$  which happens to be a substitute of  $l$  with the corresponding substitute  $t'_0$  of  $r$ , or *vice versa*. This means that there exists a position  $\alpha$  and a substitution  $\sigma$ , *i.e.*, a mapping from  $\mathcal{X}$  into  $\mathbf{Term}_{\mathcal{F}}$ , such that the  $\alpha$ th subterm of  $t$  is  $l\sigma$  and  $t'$  is obtained from  $t$  by replacing the  $\alpha$ th subterm with  $r\sigma$ —or *vice versa* exchanging the roles of  $l$  and  $r$  (Figure 1). In other words, we apply one of the two rules  $l \rightarrow r, r \rightarrow l$ .

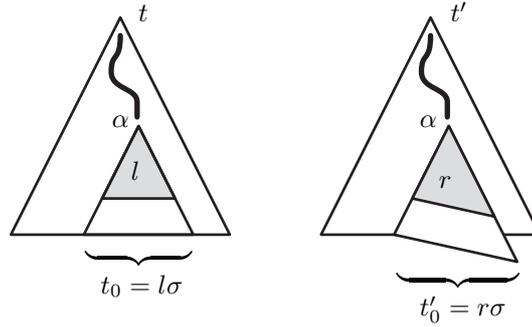


FIGURE 1. Applying the law  $l = r$  to a term  $t$ , here viewed as a rooted tree: one replaces some subterm  $t_0$  of  $t$  that is a substitute of  $l$  with the corresponding substitute of  $r$ —or *vice versa*.

**Definition 1.1.** For each family of algebraic laws  $\mathcal{L}$  involving the signature  $\mathcal{F}$ , we denote by  $R_{\mathcal{L}}$  the rewrite system on  $\mathbf{Term}_{\mathcal{F}}$  consisting of all rules  $l \rightarrow r$  and  $r \rightarrow l$  for  $l = r$  a law of  $\mathcal{L}$ .

Thus, by very definition, we have

**Proposition 1.2.** *Let  $\mathcal{L}$  be a family of algebraic laws involving the signature  $\mathcal{F}$ . Then two terms  $t, t'$  in  $\mathbf{Term}_{\mathcal{F}}$  are  $\mathcal{L}$ -equivalent if and only if one has  $t \leftrightarrow_{\mathcal{L}} t'$ , where  $\leftrightarrow_{\mathcal{L}}$  is the equivalence relation generated by  $R_{\mathcal{L}}$ .*

**1.2. Geometry monoid: the principle.** The rewrite systems  $R_{\mathcal{L}}$  are not functional: there may be many ways to apply one of its rules to a given term. However, these systems can be viewed as the union of a family of partial functions, each of which corresponds to choosing a law, an orientation, and a position.

For the sequel, it is convenient to fix an addressing system for the subterms of a term, *i.e.*, for the positions in a term. As is usual, we shall see terms as rooted trees

(*cf.* Figure 1), where the inner nodes are labeled using operation symbols, and the leaves are labeled using variables. Then a subterm of a term  $t$  is naturally specified by the node where its root lies, which is itself determined by the path that connects the root of the tree to that node. If all operation symbols are binary—which will be the case in the examples considered below—we can for instance use finite sequences of 0’s and 1’s to describe such a path, using 0 for “forking to the left” and 1 for “forking to the right”. We use  $\emptyset$  for the empty address, *i.e.*, the address of the root. If  $t$  is a term, and  $\alpha$  is an address, we denote by  $\text{sub}(t, \alpha)$  the  $\alpha$ -th subterm of  $t$ , *i.e.*, the subterm of  $t$  whose position is specified by  $\alpha$ . Note that  $\text{sub}(t, \alpha)$  exists only for  $\alpha$  short enough:  $\text{sub}(t, \emptyset)$  always exists and equals  $t$ , but, for instance,  $\text{sub}(t, 0)$  and  $\text{sub}(t, 1)$ , which are the left and the right subterms of  $t$  respectively, exist only if  $t$  is not a variable.

**Definition 1.3.** (i) Assume that  $L$  is an oriented algebraic law. For each address  $\alpha$  and each orientation  $e$  (namely  $+$  or  $-$ ), we denote by  $L_\alpha^e$  the (partial) operator on terms corresponding to applying  $L$  at position  $\alpha$  in direction  $e$ .

(ii) For  $\mathcal{L}$  a family of oriented laws, we define the *geometry monoid of  $\mathcal{L}$* , denoted  $\mathcal{Geom}_{\mathcal{L}}$ , to be the monoid generated, using composition, by all operators  $L_\alpha^e$  when  $L$  ranges over  $\mathcal{L}$ ,  $\alpha$  ranges over all addresses, and  $e$  ranges over  $\{+, -\}$ .

We shall always think of the operators  $L_\alpha^e$  as acting on the right, thus writing  $t \cdot f$  rather than  $f(t)$  for the image of the term  $t$  under the operator  $f$ . To be coherent, we use the reversed composition, denoted  $\bullet$ , in the geometry monoid. Thus,  $f \bullet g$  means “ $f$  then  $g$ ”.

**Remark 1.4.** When a term  $t_0$  is a substitute of the term  $l$ , the substitution  $\sigma$  such that  $t_0 = l\sigma$  is not unique, as the value of  $x\sigma$  is uniquely determined only for those variables  $x$  that occur in  $l$ . Hence, for  $L = (l, r)$ , the operator  $L_\alpha^+$  is functional only if the same variables occur in  $l$  and  $r$ . The laws that satisfy this condition will be called *balanced*. Although this is not necessary, we shall always restrict to balanced laws in the sequel. Thus we discard laws like  $x = x * y$ ; note that the algebras obeying such laws are trivial.

By construction, every element in a geometry monoid  $\mathcal{Geom}_{\mathcal{L}}$  is a finite product of operators  $L_\alpha^e$  with  $L$  in  $\mathcal{L}$ . It will be convenient in the sequel to fix the following notation:

**Notation 1.5.** For  $\mathcal{L}$  a family of (oriented, balanced) laws, we denote by  $\mathcal{W}_{\mathcal{L}}$  the free monoid generated by all letters  $L_\alpha^e$  with  $L \in \mathcal{L}$ ,  $\alpha \in \{0, 1\}^*$ , and  $e \in \{+, -\}$ ; we denote by *eval* the canonical evaluation morphism

$$\text{eval} : \mathcal{W}_{\mathcal{L}} \rightarrow \mathcal{Geom}_{\mathcal{L}}.$$

Thus  $\mathcal{W}_{\mathcal{L}}$  consists of all abstract products of letters  $L_\alpha^e$ , while  $\mathcal{Geom}_{\mathcal{L}}$  consists of actual operators on terms. We shall see soon that, in general (and as can be expected), the evaluation mapping is far from injective, *i.e.*, the geometry monoid  $\mathcal{Geom}_{\mathcal{L}}$  is far from free.

**1.3. Geometry monoid: the example of ALD.** To illustrate our approach, we consider the family (ALD) consisting of the following three algebraic laws:

$$\begin{aligned} \text{(LD)} \quad & x * (y * z) = (x * z) * (x * z), \\ \text{(ALD}_1\text{)} \quad & x * (y \circ z) = (x * z) \circ (x * z), \\ \text{(ALD}_2\text{)} \quad & x * (y * z) = (x \circ y) * z. \end{aligned}$$

The specific interest of this choice is that, contrary to (LD) alone, the above mixed laws, collectively denoted (ALD)—“Augmented Left self-Distributivity”—in the sequel, have never been investigated from the viewpoint of the geometry monoid and, so, the results we shall obtain below are new.

Here the signature consists of two binary operation symbols  $*$ ,  $\circ$ . The law (LD) is the left self-distributivity law, which was extensively studied in [9]. The additional laws (ALD<sub>1</sub>) and (ALD<sub>2</sub>) express that  $*$  is left distributive with respect to  $\circ$  and that  $\circ$  behaves like a sort of composition relative to  $*$ . Many examples of LD-algebras, *i.e.*, of structures satisfying (LD), happen to be equipped with a second operation that satisfies the mixed laws (ALD<sub>1</sub>) and (ALD<sub>2</sub>). This is in particular the case for every group equipped with the LD-operation  $x * y := xyx^{-1}$ : in this case, defining the second operation to be the multiplication  $x \circ y := xy$  yields an ALD-algebra—and more, actually, namely an LD-monoid in the sense of [9], Chapter XI.

By definition, the geometry monoid  $\mathcal{Geom}_{\text{ALD}}$  is generated by three families of operators, corresponding to the three laws. In the specific case, we observe that the actions of the operators  $\text{LD}_\alpha^+$  and  $(\text{ALD}_1)_\alpha^+$  are parallel, as both consist in distributing the left subterm to the two halves of the right subterm. However, their domains are disjoint, as  $\text{LD}_\alpha^+$  applies only when the symbol at  $\alpha 1$  is  $*$ , while  $(\text{ALD}_1)_\alpha^+$  applies only when the symbol at  $\alpha 1$  is  $\circ$ . Hence, instead of considering two operators separately, we shall introduce their union, which is still functional, and denote it by  $\Sigma_\alpha^+$ . In particular, we have

$$(1.1) \quad \Sigma_\emptyset^+ : t_1 * (t_2 \square t_3) \rightarrow (t_1 * t_2) \square (t_1 * t_3),$$

*i.e.*,  $\Sigma_\emptyset^+$ , which will be also simply denoted  $\Sigma^+$ , is the operator that maps every term of the form  $t_1 * (t_2 \square t_3)$  to the corresponding term  $(t_1 * t_2) \square (t_1 * t_3)$ , where  $\square$  stands for either  $*$  or  $\circ$ . Similarly, we have

$$(1.2) \quad A_\emptyset^+ : t_1 * (t_2 * t_3) \rightarrow (t_1 \circ t_2) * t_3,$$

*i.e.*,  $A_\emptyset^+$ , also denoted  $A^+$ , maps every term of the form  $t_1 * (t_2 * t_3)$  to  $(t_1 \circ t_2) * t_3$ . By definition, the geometry monoid  $\mathcal{Geom}_{\text{ALD}}$  is the monoid generated by all partial operators  $\Sigma_\alpha^+$ ,  $\Sigma_\alpha^-$ ,  $A_\alpha^+$ , and  $A_\alpha^-$  with  $\alpha$  ranging over  $\{0, 1\}^*$ .

As displayed in Figure 2, a given term may belong to the domain and the range of several operators  $\Sigma_\alpha^\pm$  and  $A_\alpha^\pm$ .

**Remark 1.6.** For a signature  $\mathcal{F}$  comprising more than one operation symbol, as is the case with ALD, only taking into account the address where a law is applied does not exhaust all information. Indeed, when an operator  $L_\alpha^e$  is applied to a term  $t$ , the complete list of the operation symbols that occur in  $t$  above  $\alpha$  may be important, typically in terms of the relations possibly connecting various operators. In order to include such data, we can use positions instead of addresses to index the operators of the geometry monoid, a *position* being defined to be a finite sequence of even length consisting of alternating operation symbols and forking directions: for instance,  $(*, 1)$  and  $(*, 1, \circ, 0)$  are positions that refine the addresses 1 and 10

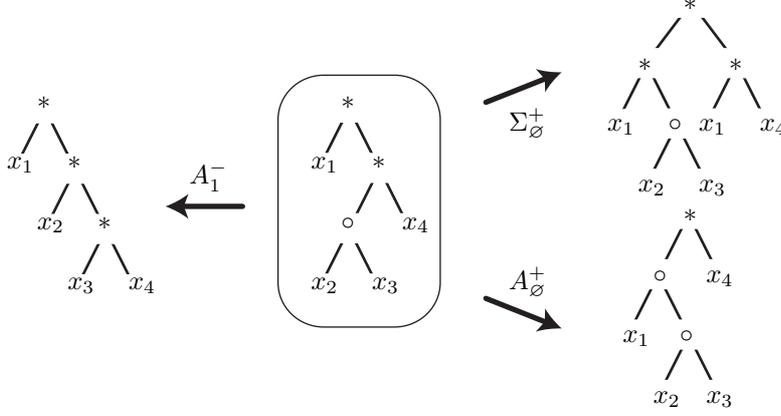


FIGURE 2. Case of ALD: Two positive operators apply to the term  $x_1 * ((x_2 \circ x_3) * x_4)$ , namely  $\Sigma_{\emptyset}^+$  and  $A_{\emptyset}^+$ , while one negative operator applies to it, namely  $A_1^-$ , *i.e.*, the copy of  $A_{\emptyset}^-$  acting at position 1.

respectively. In this framework, the operator  $A_1^-$  mentioned in Figure 2 would become  $A_{(*,1)}^-$ , *i.e.*,  $A_1^-$  applied to a term with  $*$  at the root. Observe that an operator  $L_{\alpha}^e$  as defined in Definition 1.3 is, as a set of pairs of terms, just the disjoint union of all operators  $L_p^e$  with  $p$  a position that projects on the address  $\alpha$  when the operation symbols are forgotten. Although these refinements may be necessary in some cases, they are not in the examples considered in this paper, in particular ALD, and there will be no need to split the operators  $L_{\alpha}^e$  into more elementary components.

**1.4. Connection between  $\mathcal{Geom}_{\mathcal{L}}$  and  $\mathcal{L}$ -equivalence.** We saw in Proposition 1.2 that  $\mathcal{L}$ -equivalence is directly connected with the equivalence relation  $\leftrightarrow_{\mathcal{L}}$  associated with the rewrite system  $R_{\mathcal{L}}$ . As a consequence, it is also connected with the geometry monoid  $\mathcal{Geom}_{\mathcal{L}}$ :

**Proposition 1.7.** *Let  $\mathcal{L}$  be any family of algebraic laws involving the signature  $\mathcal{F}$ . Then two terms  $t, t'$  in  $\mathbf{Term}_{\mathcal{F}}$  are  $\mathcal{L}$ -equivalent if and only if some element of the geometry monoid  $\mathcal{Geom}_{\mathcal{L}}$  maps  $t$  to  $t'$ .*

*Proof.* Let us write  $t \equiv t'$  if some operator of  $\mathcal{Geom}_{\mathcal{L}}$  maps  $t$  to  $t'$ . First  $t \equiv t'$  implies  $t \leftrightarrow_{\mathcal{L}} t'$ . Indeed, by construction,  $t' = t \cdot L_{\alpha}^e$  implies  $t \leftrightarrow_{\mathcal{L}} t'$ . As  $\leftrightarrow_{\mathcal{L}}$  is transitive, the same is true when several operators  $L_{\alpha}^e$  are composed.

Conversely,  $\leftrightarrow_{\mathcal{L}}$  is the congruence generated by the instances of the laws of  $\mathcal{L}$ , so, in order to prove that  $t \leftrightarrow_{\mathcal{L}} t'$  implies  $t \equiv t'$ , it is enough to prove that  $\equiv$  is a congruence on terms, and that it contains all instances of the laws of  $\mathcal{L}$ . Now,  $\equiv$  is an equivalence relation because  $\mathcal{Geom}_{\mathcal{L}}$  is closed under composition and contains the identity mapping, and it is a congruence, *i.e.*, it is compatible with all operations of  $\mathcal{F}$ . Indeed, assume that  $L_{\alpha}^+$  maps  $t$  to  $t'$ . Let  $t_0$  be any term and  $\square$  be any operation in  $\mathcal{F}$ . Then—assuming that all operations of  $\mathcal{F}$  are binary and 0, 1 addresses are used—the operator  $L_{0\alpha}^+$  maps  $t_0 \square t$  to  $t_0 \square t'$ , while  $L_{1\alpha}^+$  maps  $t \square t_0$  to  $t' \square t_0$ . Finally, assume that  $(t, t')$  is an instance of some law  $L$  of  $\mathcal{L}$ . This means that there exists a substitution  $\sigma$  such that, assuming that  $L$  is  $l = r$ , one has  $t = l\sigma$  and  $t' = r\sigma$ . Now, by definition,  $L_{\emptyset}^+$  maps  $t$  to  $t'$ , so  $t \equiv t'$  holds.  $\square$

Our aim in this paper is not to develop a general theory of the geometry monoid. However, we mention two results, namely one about the structure of  $\mathcal{G}eom_{\mathcal{L}}$  and one about its dependence on  $\mathcal{L}$ .

First, we observe that, by construction, each operator in the geometry monoid  $\mathcal{G}eom_{\mathcal{L}}$  is close to admitting an inverse. Indeed, the operator  $L_{\alpha}^{-}$  is, as a set of pairs, the inverse of the operator  $L_{\alpha}^{+}$ , *i.e.*,  $L_{\alpha}^{+}$  maps  $t$  to  $t'$  if and only if  $L_{\alpha}^{-}$  maps  $t'$  to  $t$ . This is not enough to make  $\mathcal{G}eom_{\mathcal{L}}$  into a group, as the composition  $L_{\alpha}^{+} \bullet L_{\alpha}^{-}$  is the identity operator of its domain only, and *not* the identity operator of  $\mathbf{Term}_{\mathcal{F}}$  in general, but we have:

**Proposition 1.8.** *Assume that  $\mathcal{L}$  is a family of balanced algebraic laws. Then  $\mathcal{G}eom_{\mathcal{L}}$  is an inverse monoid.*

*Proof.* Assume that  $f$  is a nonempty operator in  $\mathbf{Geom}_{\mathcal{L}}$ . Then, by construction, there exists a word  $w$  in  $\mathcal{W}_{\mathcal{L}}$  such that  $f$  equals  $eval(w)$ . Let  $w^{-1}$  be the formal inverse of  $w$ , *i.e.*, the word obtained from  $w$  by exchanging  $L_{\alpha}^{+}$  and  $L_{\alpha}^{-}$  everywhere in  $w$  and reversing the order of the letters, and let  $g := eval(w^{-1})$ . An immediate induction shows that a pair of terms  $(t, t')$  belongs to  $f$  if and only if the pair  $(t', t)$  belongs to  $g$ , and we deduce

$$f \bullet g \bullet f = f \quad \text{and} \quad g \bullet f \bullet g = g,$$

so every element in  $\mathbf{Geom}_{\mathcal{L}}$  possesses an inverse as required for an inverse monoid—this applies in particular to  $f = g = \emptyset$ .  $\square$

As for the second result, the *equational variety* associated with a family  $\mathcal{L}$  of algebraic laws involving the signature  $\mathcal{F}$  is, by definition, the collection of all  $\mathcal{F}$ -structures that satisfy the laws of  $\mathcal{L}$ . Different families of laws may give rise to the same variety: for instance, when appended to the commutativity law  $xy = yx$ , the associativity law  $x(yz) = (xy)z$  and the law  $x(yz) = z(yx)$  define the same variety.

**Proposition 1.9.** [7] *Up to isomorphism, the monoid  $\mathcal{G}eom_{\mathcal{L}}$  only depends on the equational variety defined by  $\mathcal{L}$ : if  $\mathcal{L}$  and  $\mathcal{L}'$  define the same variety, then the monoids  $\mathcal{G}eom_{\mathcal{L}}$  and  $\mathcal{G}eom_{\mathcal{L}'}$  are isomorphic.*

*Sketch of proof.* It suffices to show that, when we add to a family  $\mathcal{L}$  a new law  $L$  that is a consequence of the laws of  $\mathcal{L}$ , then the geometry monoid is not changed. Assume  $L = (l, r)$ . By Proposition 1.7, the hypothesis that  $L$  is a consequence of  $\mathcal{L}$  implies that some operator  $f$  in  $\mathcal{G}eom_{\mathcal{L}}$  maps  $l$  to  $r$ . The geometry monoid  $\mathcal{G}eom_{\mathcal{L} \cup \{L\}}$  is obtained from  $\mathcal{G}eom_{\mathcal{L}}$  by adding a new generator that is a product of the canonical generators of  $\mathcal{G}eom_{\mathcal{L}}$ , and, therefore, it is isomorphic to  $\mathcal{G}eom_{\mathcal{L}}$ .  $\square$

## 2. REPLACING THE GEOMETRY MONOID WITH A GROUP

The first specific step that proves to be often useful in investigating the geometry monoids consists in replacing the monoid  $\mathcal{G}eom_{\mathcal{L}}$  with a *group*  $\mathbf{Geom}_{\mathcal{L}}$  that resembles it. The benefit of the replacement is the possibility of freely computing with inverses and avoiding the problems possibly connected with the empty operator. However, there is no universal recipe for going from  $\mathcal{G}eom_{\mathcal{L}}$  to a (non-trivial) group, and we shall complete the approach in specific cases only, by guessing confluence relations in  $\mathcal{G}eom_{\mathcal{L}}$  and introducing the abstract group defined by these relations.

**2.1. The linear case.** For each inverse monoid  $M$ , there exists a biggest quotient of  $M$  that is a group, namely the *universal group*  $U(M)$  of  $M$  obtained by collapsing all idempotents to 1 [21]. When the geometry monoid  $\mathcal{Geom}_{\mathcal{L}}$  does not contain the empty operator, the idempotent elements are those operators that are identity on their domain, and the associated group keeps enough information to be non-trivial. We shall first briefly mention one case when this favourable situation occurs, namely the case of *linear* laws.

**Definition 2.1.** A term  $t$  is said to be *injective* if no variable occurs twice in  $t$ . An algebraic law  $l = r$  is said to be *linear* if it is balanced and, in addition, the terms  $l$  and  $r$  are injective.

For instance, associativity and commutativity are linear laws, while self-distributivity is not, as the variable  $x$  is repeated twice in the right hand term of  $x(yz) = (xy)(xz)$ .

In order to describe the case of linear laws, we start with a general result about the elements of a geometry monoid.

**Definition 2.2.** Assume that  $f$  is an operator on  $\mathbf{Term}_{\mathcal{F}}$ . We say that a pair of terms  $(l, r)$  is a *seed* for  $f$  if  $f$ , as a set of pairs, consists of all instances of  $(l, r)$ , *i.e.*, consists of all pairs  $(l\sigma, r\sigma)$  with  $\sigma$  a  $\mathbf{Term}_{\mathcal{F}}$ -valued substitution.

If  $L$  is the law  $l = r$ , then  $L_{\emptyset}^+$  consists of all instances of  $(l, r)$ , so  $(l, r)$  is a seed for  $L_{\emptyset}^+$ . More generally we have:

**Lemma 2.3.** [9] *Assume that  $\mathcal{L}$  is a family of balanced algebraic laws involving a single binary operation symbol. Then each nonempty operator  $f$  in  $\mathcal{Geom}_{\mathcal{L}}$  admits a seed.*

*Sketch of proof.* First, every operator  $L_{\alpha}^+$  admits a seed, because, if  $(l, r)$  is a seed for  $L_{\emptyset}^+$  and  $x$  is a variable not occurring in  $l$  and  $r$ , then, assuming that  $*$  is the operation symbol,  $(x * l, x * r)$  is a seed for  $L_1^+$ ,  $(l * x, r * x)$  is a seed for  $L_0^+$ , and an easy induction gives the result for every  $L_{\alpha}^+$ .

Then, by construction, every operator in  $\mathcal{Geom}_{\mathcal{L}}$  is a finite composition of operators  $L_{\alpha}^e$  with  $L$  in  $\mathcal{L}$ , and, by definition, the latter admit seeds. Hence, the point is to show that the composition of two operators admitting a seed still admits a seed provided it is nonempty. So, assume that  $f$  and  $g$  consist of all instances of  $(l_1, r_1)$  and  $(l_2, r_2)$ , respectively. Two cases may occur. Either there exists no term that is a substitute both of  $r_1$  and  $l_2$ , and in this case  $f \bullet g$  is empty. Or there exists such a term. Then, as is well known [22], there exists a most general unifier (MGU) of the terms  $r_1$  and  $l_2$ , *i.e.*, there exist two substitutions  $\sigma, \tau$  satisfying  $r_1\sigma = l_2\tau$  and such that every common substitute of  $r_1$  and  $l_2$  is a substitute of  $r_1\sigma$ . In this case, it is easy to check that  $(l_1\sigma, r_2\tau)$  is a seed for  $f \bullet g$ .  $\square$

**Remark 2.4.** The restriction on the signature in Lemma 2.3 can be dropped at the expense of splitting the operators  $L_{\alpha}^+$  according to positions as explained in Remark 1.6, or, alternatively, of considering a more general notion of terms and instances in which substitution is possible not only for variables but also for some specific operation symbols considered as variables of a new type. Once again, there is no need to go into details here, as we shall not use such notions.

**Lemma 2.5.** *Assume that  $\mathcal{L}$  is a family of linear laws involving a single binary operation symbol. Then the seed of every operator in  $\mathcal{Geom}_{\mathcal{L}}$  is a pair of injective terms. Moreover,  $\mathcal{Geom}_{\mathcal{L}}$  does not contain the empty operator.*

*Proof.* The MGU of two injective terms always exists and it is an injective term: indeed, the only reason that may cause a unifier not to exist is a variable clash, and this cannot happen with injective terms. So the result about the seed follows from an induction. As for the empty operator, it cannot appear as the needed unifier always exists.  $\square$

**Definition 2.6.** Assume that  $f, g$  are partial mappings on some set  $S$ . We say that  $f \sim g$  (*resp.*  $f \approx g$ ) holds if there exists at least one element  $x$  in  $S$  such that  $f(x) = g(x)$  holds (*resp.* if  $f(x) = g(x)$  holds for every  $x$  in the intersection of the domains of  $f$  and  $g$ ).

When the geometry monoid  $\mathcal{G}eom_{\mathcal{L}}$  contains the empty operator, the relations  $\sim$  and  $\approx$  need not be transitive. This however cannot happen in the linear case:

**Proposition 2.7.** [12] *Assume that  $\mathcal{L}$  is a family of linear laws. Then the relations  $\sim$  and  $\approx$  coincide, and they are congruences on the monoid  $\mathcal{G}eom_{\mathcal{L}}$ . Furthermore, the quotient-monoid  $\mathcal{G}eom_{\mathcal{L}}/\approx$  is a group, and it is the universal group of  $\mathcal{G}eom_{\mathcal{L}}$ .*

Under the previous hypotheses, we shall denote by  $\mathbf{G}eom_{\mathcal{L}}$  the group  $\mathcal{G}eom_{\mathcal{L}}/\approx$ . So, in this case, we have a scheme of the form

$$(2.1) \quad \mathcal{W}_{\mathcal{L}} \xrightarrow{\text{onto}} \mathcal{G}eom_{\mathcal{L}} \xrightarrow{\text{onto}} \mathbf{G}eom_{\mathcal{L}}.$$

Moreover, the partial action of  $\mathcal{G}eom_{\mathcal{L}}$  on terms induces a well-defined action of  $\mathbf{G}eom_{\mathcal{L}}$  as, by definition, all operators in a  $\approx$ -class agree on the terms that lie in their domain. Furthermore, no information is lost when we replace  $\mathcal{G}eom_{\mathcal{L}}$  with  $\mathbf{G}eom_{\mathcal{L}}$  in the precise sense that the counterpart of Proposition 1.7 is true: two terms  $t, t'$  are  $\mathcal{L}$ -equivalent if and only if we have  $t' = t \cdot g$  for some  $g$  in  $\mathbf{G}eom_{\mathcal{L}}$ . So, in this case, one can replace the monoid  $\mathcal{G}eom_{\mathcal{L}}$  with the group  $\mathbf{G}eom_{\mathcal{L}}$  in all further uses, and it is then natural to call the latter the *geometry group* of  $\mathcal{L}$ .

**Example 2.8.** Let  $\mathbf{A}$  denote the associativity law  $x(yz) = (xy)z$ . Then the corresponding group  $\mathbf{G}eom_{\mathbf{A}}$  is a well known group, namely R. Thompson's group  $F$  [23]. In the case of associativity together with commutativity, the corresponding group is R. Thompson's group  $V$ —*cf.* [12].

**Remark 2.9.** Even in the smooth case of linear laws, the action of the group  $\mathbf{G}eom_{\mathcal{L}}$  on terms is a partial action: for  $t \cdot g$  to be defined, it is necessary that  $t$  be large enough. However, we can obtain an everywhere defined action by considering infinite trees; equivalently, this amounts to defining an action on the boundary of the family of finite terms, which is a Cantor set in the case of terms involving one binary operation symbol and one variable.

**2.2. Confluence relations: the principle.** Whenever the empty operator occurs in the geometry monoid, the previous approach badly fails:

**Lemma 2.10.** *Assume that  $\mathcal{G}eom_{\mathcal{L}}$  contains the empty operator. Then the universal group  $U(\mathcal{G}eom_{\mathcal{L}})$  is trivial, i.e., it reduces to  $\{1\}$ .*

*Proof.* Assume that  $\pi$  is a homomorphism of  $\mathcal{G}eom_{\mathcal{L}}$  to a group. Then, for each  $f$  in  $\mathcal{G}eom_{\mathcal{L}}$ , we have  $\emptyset \bullet f = \emptyset$ , hence  $\pi(f)\pi(\emptyset) = \pi(\emptyset)$ , whence  $\pi(f) = 1$ .  $\square$

**Example 2.11.** The case when there is no unifier frequently occurs. For instance, in the case of the self-distributivity law (LD), every term  $t$  belonging to the range

of  $\Sigma_{\emptyset}^+$  satisfies  $\mathbf{sub}(t, 00) = \mathbf{sub}(t, 10)$ ; it follows that every term  $t$  in the range of  $\Sigma_{\emptyset}^+ \bullet \Sigma_1^+$  satisfies  $\mathbf{sub}(t, 00) = \mathbf{sub}(t, 100)$ , and therefore  $\mathbf{sub}(t, 00) \neq \mathbf{sub}(t, 10)$ . Hence no term in the image of  $\Sigma_{\emptyset}^+ \bullet \Sigma_1^+$  may belong to the domain of  $\Sigma_{\emptyset}^-$ : in other words, the composition of the operators  $\Sigma_{\emptyset}^+$ ,  $\Sigma_1^+$ , and  $\Sigma_{\emptyset}^-$  is empty, *i.e.*, the operator  $\Sigma^+ \bullet \Sigma_1^+ \bullet \Sigma^-$ , as a set of ordered pairs, is the empty set, and, in particular, its domain of definition is empty. In other words, we have in  $\mathcal{Geom}_{\text{LD}}$ —as well as in  $\mathcal{Geom}_{\text{ALD}}$ —the relation

$$\text{eval}(\Sigma_{\emptyset}^+ \Sigma_1^+ \Sigma_{\emptyset}^-) = \emptyset.$$

This implies that any word  $w$  in  $\mathcal{W}_{\text{LD}}$  containing  $\Sigma_{\emptyset}^+ \Sigma_1^+ \Sigma_{\emptyset}^-$  as a factor evaluates in  $\mathcal{Geom}_{\text{LD}}$  into the empty operator. This is in particular the case for the symmetrized word  $\Sigma_{\emptyset}^+ \Sigma_1^+ \Sigma_{\emptyset}^- \Sigma_{\emptyset}^+ \Sigma_1^- \Sigma_{\emptyset}^-$ , although the latter appears as freely reducing to the empty word and one might therefore expect the associated operator to be close to the identity.

The above situation almost always occurs when non-linear laws are involved. As the example of  $(xy)(xz) = (xz)(xy)$  shows, it is not readily true that the presence of at least one non-linear law in  $\mathcal{L}$  forces the empty operator to belong to  $\mathcal{Geom}_{\mathcal{L}}$ , but, for instance, it is sufficient for implementing the argument used in Example 2.11 that  $\mathcal{L}$  contains a law  $l = r$  such that, for some variable  $x$ , the sets  $\{\alpha_1, \dots, \alpha_p\}$  and  $\{\beta_1, \dots, \beta_q\}$  where  $x$  respectively occurs in  $l$  and  $r$  are distinct and at least one of them is not a singleton.

In such cases, the universal group  $U(\mathcal{Geom}_{\mathcal{L}})$  is of no use, and we have to look for another method. A misleading attempt would be to try to modify the construction of the geometry monoid so as to artificially discard the empty operator. We doubt that anything interesting can occur by doing so—see [9] for a more thorough discussion. Instead, we shall now develop a completely different method for associating with  $\mathcal{Geom}_{\mathcal{L}}$  a group that keeps the meaningful information of  $\mathcal{Geom}_{\mathcal{L}}$ , namely finding a presentation of  $\mathcal{Geom}_{\mathcal{L}}$  and introducing the group that admits this presentation—whatever its connection with  $\mathcal{Geom}_{\mathcal{L}}$  is.

Actually, finding a presentation of  $\mathcal{Geom}_{\mathcal{L}}$  is in general out of reach, at least by a direct approach. So, once again, we use an indirect approach consisting in isolating *some* relations satisfied in  $\mathcal{Geom}_{\mathcal{L}}$  using a uniform scheme, but not trying to prove that these relations make a presentation: the latter will possibly come at the very end when further constructions have been performed.

So, at this point, the problem is to find relations connecting the various operators  $L_{\alpha}^e$  of  $\mathcal{Geom}_{\mathcal{L}}$ . In the sequel, we shall concentrate on *positive* relations, *i.e.*, on relations that involve the operators  $L_{\alpha}^+$  but not their inverses.

**Definition 2.12.** For  $\mathcal{L}$  a family of oriented algebraic laws, we denote by  $R_{\mathcal{L}}^+$  the rewrite system comprising the rules  $l \rightarrow r$  for  $(l, r)$  in  $\mathcal{L}$ ; the *positive geometry monoid*  $\mathcal{Geom}_{\mathcal{L}}^+$  of  $\mathcal{L}$  is defined to be the submonoid of  $\mathcal{Geom}_{\mathcal{L}}$  generated by all operators  $L_{\alpha}^+$  with  $L$  in  $\mathcal{L}$  and  $\alpha$  in  $\{0, 1\}^*$ .

The connection between the rewrite system  $R_{\mathcal{L}}^+$  and the positive geometry monoid  $\mathcal{Geom}_{\mathcal{L}}^+$  is similar to the connection between the rewrite system  $R_{\mathcal{L}}$  and the geometry monoid  $\mathcal{Geom}_{\mathcal{L}}$ . We shall be looking for relations in  $\mathcal{Geom}_{\mathcal{L}}^+$ , *i.e.*, for relations connecting the operators  $L_{\alpha}^+$  in  $\mathcal{Geom}_{\mathcal{L}}^+$ . The principle will be to investigate the relations that possibly arise when two operators are applied to one and the same term  $t$ , which amounts to investigating the local confluence of the rewrite system  $R_{\mathcal{L}}^+$ . This means that, for all laws  $L, L'$  in  $\mathcal{L}$  and all addresses  $\alpha, \beta$ , we look

for relations of the generic form

$$(2.2) \quad L_\alpha^+ \bullet \dots = L_\beta^+ \bullet \dots$$

In other words, we look for common right multiples in  $\mathcal{Geom}_\mathcal{L}^+$ .

**Definition 2.13.** For  $f$  a (partial) mapping on  $\mathbf{Term}_\mathcal{F}$  and  $\alpha$  an address, we denote by  $sh_\alpha(f)$  the  $\alpha$ -*shift* of  $f$ , defined to be the partial operator on  $\mathbf{Term}_\mathcal{F}$  that consists in applying  $f$  to the  $\alpha$ th subterm of its argument:  $t \cdot sh_\alpha(f)$  is defined if and only if  $\mathbf{sub}(t, \alpha)$  exists and  $\mathbf{sub}(t, \alpha) \cdot f$  is defined, and, in this case,  $t \cdot sh_\alpha(f)$  is obtained from  $t$  by replacing the  $\alpha$ th subterm by its image under  $f$ .

So, for instance, we have  $L_\alpha^+ = sh_\alpha(L_\emptyset^+)$  for each law  $L$  and each address  $\alpha$ , and, more generally,  $L_{\alpha\beta}^+ = sh_\alpha(L_\beta^+)$  for all  $\alpha, \beta$ .

As for confluence relations in the monoid  $\mathcal{Geom}_\mathcal{L}^+$ , two general schemes will be involved. The first one relies on the following general principle:

**Lemma 2.14.** *Assume that  $\alpha$  and  $\beta$  are incomparable addresses, i.e., there exists  $\gamma$  such that  $\gamma 0$  is a prefix of  $\alpha$  and  $\gamma 1$  is a prefix of  $\beta$ , or vice versa. Then, for all partial operators  $f, g$ , we have*

$$(2.3) \quad sh_\alpha(f) \bullet sh_\beta(g) = sh_\beta(g) \bullet sh_\alpha(f).$$

*Proof.* (Figure 3) The operators  $sh_\alpha(f)$  and  $sh_\beta(g)$  act on disjoint subtrees, and therefore they commute.  $\square$

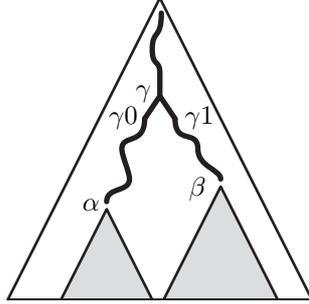


FIGURE 3. Incomparable addresses: The  $\alpha$ th and  $\beta$ th subterms are disjoint, and therefore operators acting on them commute.

In particular, we deduce:

**Proposition 2.15.** *Assume that  $\alpha$  and  $\beta$  are incomparable addresses, i.e., there exists  $\gamma$  such that  $\gamma 0$  is a prefix of  $\alpha$  and  $\gamma 1$  is a prefix of  $\beta$ , or conversely. Then we have*

$$(2.4) \quad L_\alpha^+ \bullet L_\beta^+ = L_\beta^+ \bullet L_\alpha^+.$$

The second general scheme appears in connection with what can be called a *geometric inheritance* phenomenon.

**Lemma 2.16.** *Assume that  $L$  is the oriented law  $(l, r)$  and that some variable  $x$  occurs at addresses  $\beta_1, \dots, \beta_p$  in  $l$  and at  $\gamma_1, \dots, \gamma_q$  in  $r$ . Then, for each (partial) operator  $f$  acting on terms, we have*

$$(2.5) \quad sh_{\beta_1}(f) \bullet \dots \bullet sh_{\beta_p}(f) \bullet L_\emptyset^+ = L_\emptyset^+ \bullet sh_{\gamma_1}(f) \bullet \dots \bullet sh_{\gamma_q}(f).$$

More generally, for each address  $\alpha$ , we have

$$(2.6) \quad sh_{\alpha\beta_1}(f) \bullet \dots \bullet sh_{\alpha\beta_p}(f) \bullet L_\alpha^+ = L_\alpha^+ \bullet sh_{\alpha\gamma_1}(f) \bullet \dots \bullet sh_{\alpha\gamma_q}(f).$$

*Proof.* (Figure 4) Assume that  $L_\alpha^+$  maps  $t$  to  $t'$ . This means that there exists a substitution  $\sigma$  such that  $t$  is  $l\sigma$  and  $t'$  is  $r\sigma$ . Let  $\sigma_1$  be the substitution defined by  $y\sigma_1 = y\sigma$  for  $y \neq x$ , and  $x\sigma_1 = x\sigma f$ . Let  $t_1 := l\sigma_1$  and  $t'_1 := r\sigma_1$ . Then, by construction,  $L^+$  maps  $t_1$  to  $t'_1$ . Now,  $t_1$  is obtained from  $t$  by replacing the subterms at addresses  $\beta_1, \dots, \beta_p$  with their image under  $f$ , so we have

$$t_1 = t \cdot (sh_{\beta_1}(f) \bullet \dots \bullet sh_{\beta_p}(f)).$$

Similarly,  $t'_1$  is obtained from  $t'$  by replacing the subterms at addresses  $\gamma_1, \dots, \gamma_q$  with their image under  $f$ , so we have

$$t'_1 = t' \cdot (sh_{\gamma_1}(f) \bullet \dots \bullet sh_{\gamma_q}(f)),$$

and (2.5) follows.

Relation (2.6) is deduced by applying  $sh_\alpha$  to both terms of (2.5).  $\square$

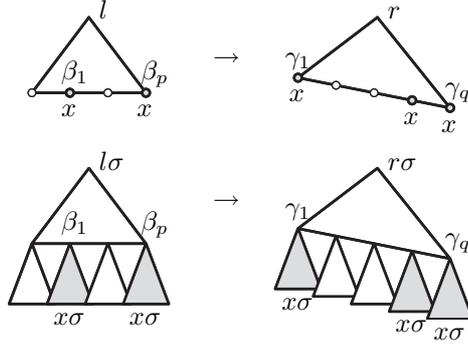


FIGURE 4. Inheritance phenomenon: If the variable  $x$  occurs at  $\beta_1, \dots, \beta_p$  in  $l$  and at  $\gamma_1, \dots, \gamma_q$  in  $r$  and nowhere else, then, for each term  $l\sigma$ , the subterm  $x\sigma$  occurs at  $\beta_1, \dots, \beta_p$  in  $l\sigma$ , and at  $\gamma_1, \dots, \gamma_q$  in  $r\sigma$ , and applying  $f$  in each  $x\sigma$  before or after applying  $l \rightarrow r$  leads to the same result.

By applying Lemma 2.16 to the case when the operator  $f$  has the form  $L'_\delta$ , we obtain:

**Proposition 2.17.** *Assume that  $L$  is the oriented law  $(l, r)$  and that some variable  $x$  occurs at addresses  $\beta_1, \dots, \beta_p$  in  $l$  and at  $\gamma_1, \dots, \gamma_q$  in  $r$ . Then, for all addresses  $\alpha, \delta$  and each law  $L'$ , we have*

$$(2.7) \quad L_\alpha^+ \bullet L_{\alpha\gamma_1\delta}^+ \bullet \dots \bullet L_{\alpha\gamma_q\delta}^+ = L'_{\alpha\beta_1\delta} \bullet \dots \bullet L'_{\alpha\beta_p\delta} \bullet L_\alpha^+.$$

In general, the previous two general schemes do not exhaust all possible confluence relations in  $\mathcal{Geom}_{\mathcal{L}}$ . Typically, no relation is obtained for pairs of the form  $(\alpha, \alpha\beta)$  where  $\beta$  is so short that it falls inside the terms  $l$  or  $r$  associated with the considered law. In this case, no general scheme is known, and one has to look at the specific terms in order to find possible confluence relations.

**2.3. Confluence relations: the example of ALD.** We illustrate the previous scheme on the case of ALD. The first family of confluence relations comprises the trivial commutation relations, here all relations

$$(2.8) \quad L_\alpha \bullet L'_\beta = L'_\beta \bullet L_\alpha \quad \text{for } L, L' = \Sigma, A \text{ and } \alpha, \beta \text{ incomparable.}$$

The second family of confluence relations comprises the geometric inheritance relations. As for the laws LD and  $\text{ALD}_1$ , *i.e.*, for the operator  $\Sigma$ , three variables are involved in the rule

$$(2.9) \quad x * (y \square z) \rightarrow (x * y) \square (x * z).$$

The variable  $x$  occurs at address 0 on the LHS of (2.9), while it occurs at 00 and 10 on the RHS. So Relation (2.7) is here

$$(2.10) \quad \Sigma_\alpha^+ \bullet L_{\alpha 0\delta}^+ = L_{\alpha 00\delta}^+ \bullet L_{\alpha 10\delta}^+ \bullet \Sigma_\alpha^+$$

for  $L = \Sigma, A$ . Similarly, the variable  $y$  occurs at 10 on the LHS of (2.9), and at 01 on the RHS, while the variable  $z$  occurs at 11 on both sides. Thus Relations (2.7) become

$$(2.11) \quad \Sigma_\alpha^+ \bullet L_{\alpha 10\delta}^+ = L_{\alpha 01\delta}^+ \bullet \Sigma_\alpha^+,$$

$$(2.12) \quad \Sigma_\alpha^+ \bullet L_{\alpha 11\delta}^+ = L_{\alpha 11\delta}^+ \bullet \Sigma_\alpha^+ \quad \text{for } L = \Sigma_\emptyset, A_\emptyset.$$

The treatment is analogous for the law  $\text{ALD}_2$ , *i.e.*, for the operator  $A$ . Three variables occur in the rule

$$(2.13) \quad x * (y * z) \rightarrow (x \circ y) * z.$$

The variable  $x$  occurs at 0 on the LHS of (2.13), and at 00 on the RHS;  $y$  occurs at 10 and at 01 respectively; finally,  $z$  occurs at 11 and at 1. The corresponding Relations (2.7) are

$$(2.14) \quad A_\alpha^+ \bullet L_{\alpha 0\delta}^+ = L_{\alpha 00\delta}^+ \bullet A_\alpha^+,$$

$$(2.15) \quad A_\alpha^+ \bullet L_{\alpha 10\delta}^+ = L_{\alpha 01\delta}^+ \bullet A_\alpha^+,$$

$$(2.16) \quad A_\alpha^+ \bullet L_{\alpha 11\delta}^+ = L_{\alpha 1\delta}^+ \bullet A_\alpha^+,$$

again for  $L = \Sigma, A$ .

With the previous approach, we succeeded in finding one relation of type (2.2) for all pairs  $\alpha, \beta$ , with the exception of the pairs  $\{A_\alpha^+, \Sigma_\alpha^+\}$ , as well as all pairs  $\{L_\alpha^+, L'_{\alpha 1}^+\}$ . So the question is whether we can find in  $\text{Geom}_{\text{ALD}}^+$  relations of the form

$$L_\alpha \bullet \dots = L'_\alpha \bullet \dots \quad \text{and} \quad L_\alpha \bullet \dots = L'_{\alpha 1} \bullet \dots$$

when  $(L, L')$  ranges over the various combinations of  $\Sigma$  and  $A$ . As shown in Figure 5, there exist such relations, namely

$$(2.17) \quad \Sigma_\alpha^+ \bullet \Sigma_{\alpha 1}^+ \bullet \Sigma_\alpha^+ = \Sigma_{\alpha 1}^+ \bullet \Sigma_\alpha^+ \bullet \Sigma_{\alpha 1}^+ \bullet \Sigma_{\alpha 0}^+,$$

$$(2.18) \quad \Sigma_\alpha^+ \bullet \Sigma_{\alpha 1}^+ \bullet A_\alpha^+ = A_{\alpha 1}^+ \bullet \Sigma_\alpha^+ \bullet \Sigma_{\alpha 0},$$

$$(2.19) \quad A_\alpha^+ \bullet \Sigma_\alpha^+ = \Sigma_{\alpha 1}^+ \bullet \Sigma_\alpha^+ \bullet A_{\alpha 1}^+ \bullet A_{\alpha 0}^+.$$

Notice that all cases are not covered: there is *no* relation  $A_\alpha^+ \dots = \Sigma_\alpha^+ \dots$ , or  $A_\alpha^+ \dots = A_{\alpha 1}^+ \dots$  in the list above.

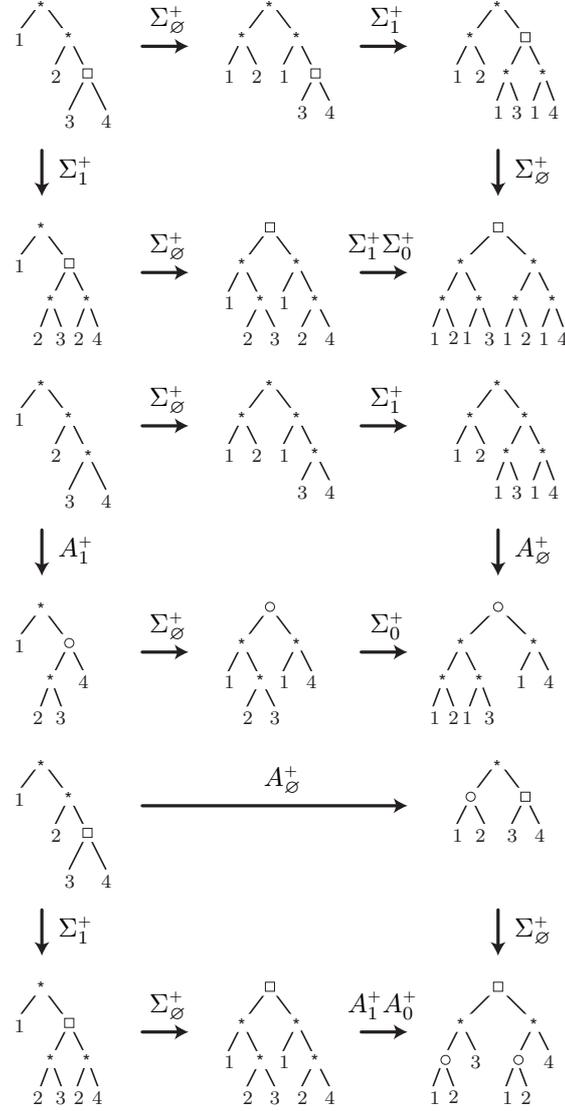


FIGURE 5. Three more types of confluence relations in  $\mathcal{Geom}_{\text{MLD}}$ : here  $\square$  stands for both  $*$  or  $\circ$ , the numbers stand for the indices of the variables.

**2.4. The geometry group.** According to the principle considered above, we introduce the abstract group for which the confluence relations of  $\mathcal{Geom}_{\mathcal{L}}$  make a presentation.

**Definition 2.18.** For  $\mathcal{L}$  a family of oriented algebraic laws, we define the *geometry group*  $\mathbf{Geom}_{\mathcal{L}}$  of  $\mathcal{L}$  to be the group generated by formal copies of the operators  $L_{\alpha}^+$  for  $L$  in  $\mathcal{L}$ , subject to all confluence relations connecting these operators in the monoid  $\mathcal{Geom}_{\mathcal{L}}^+$ . We denote by  $\mathbf{eval}$  the canonical evaluation morphism of  $\mathcal{W}_{\mathcal{L}}$  onto  $\mathbf{Geom}_{\mathcal{L}}$ .

**Example 2.19.** In the case of the ALD-laws, the group  $\mathbf{Geom}_{\text{ALD}}$  is, by definition, a group generated by two infinite series of generators  $\Sigma_\alpha$  and  $A_\alpha$  indexed by finite sequences of 0's and 1's, subject to the relations (2.8), (2.10), (2.11), (2.12), (2.17), (2.18), and (2.19).

**Example 2.20.** Let us come back to the (linear) associativity law A. One easily checks that, for each pair of addresses  $\alpha, \beta$ , there exists a natural confluence relation of the form  $A_\alpha^+ \bullet \dots = A_\beta^+ \bullet \dots$  in the positive monoid  $\mathcal{Geom}_A^+$ . Indeed, besides the quasi-commutativity inheritance relations

$$A_\alpha^+ \bullet A_{\alpha 0 \delta}^+ = A_{\alpha 0 0 \delta}^+ \bullet A_\alpha^+, \quad A_\alpha^+ \bullet A_{\alpha 1 0 \delta}^+ = A_{\alpha 0 1 \delta}^+ \bullet A_\alpha^+, \quad A_\alpha^+ \bullet A_{\alpha 1 1 \delta}^+ = A_{\alpha 1 \delta}^+ \bullet A_\alpha^+,$$

the only missing pairs are the pairs  $\alpha, \alpha 1$ , and the famous MacLane–Stasheff pentagon equality ??

$$A_\alpha^+ \bullet A_\alpha^+ = A_{\alpha 1}^+ \bullet A_\alpha^+ \bullet A_{\alpha 1}^+$$

completes the picture. Then the group  $\mathbf{Geom}_A$  presented by the above relations coincides with the group  $\mathcal{Geom}_A / \approx$  of Proposition 2.7, *i.e.*, with R. Thompson's group  $F$  [8]: so, in this case, the method based on presentation by confluence relations subsumes that of Section 2.1.

When confluence relations are used to introduce the group  $\mathbf{Geom}_\mathcal{L}$ , the connection between the monoid  $\mathcal{Geom}_\mathcal{L}$  and the group  $\mathbf{Geom}_\mathcal{L}$  is not obvious. Indeed, instead of the sequence of (2.1), the scheme is now

$$(2.20) \quad \begin{array}{ccc} \mathcal{W}_\mathcal{L} & \xrightarrow[\text{eval}]{\text{onto}} & \mathcal{Geom}_\mathcal{L} \\ \text{onto} \downarrow \text{eval} & & \\ \mathbf{Geom}_\mathcal{L} & & \end{array}$$

and it is not clear whether any factorization connects  $\mathcal{Geom}_\mathcal{L}$  and  $\mathbf{Geom}_\mathcal{L}$ . In order to prove that there exists a morphism from  $\mathcal{Geom}_\mathcal{L}$  to  $\mathbf{Geom}_\mathcal{L}$ , we ought to know that the selected confluence relations exhaust all relations holding in  $\mathcal{Geom}_\mathcal{L}$ —which cannot be the case if the empty operator occurs. And, in order to prove that there exists a morphism from  $\mathbf{Geom}_\mathcal{L}$  to  $\mathcal{Geom}_\mathcal{L}$ , we ought to be able to control the pairs  $L_\alpha^+ \bullet L_\alpha^-$  in  $\mathcal{Geom}_\mathcal{L}$ . Precisely, assume that  $w, w'$  are words in  $\mathcal{W}_\mathcal{L}$  that satisfy  $\text{eval}(w) = \text{eval}(w')$ , *i.e.*, that represent the same element of  $\mathbf{Geom}_\mathcal{L}$ . This means that there exists a finite sequence of words  $w = w_0, w_1, \dots, w_n = w'$  such that each  $w_i$  is obtained from the previous one either by applying one of the confluence relation, or by deleting some subfactor  $L_\alpha L_\alpha^{-1}$  or  $L_\alpha^{-1} L_\alpha$ , or by inserting such a subfactor. In the first two cases, the associated operators are equal, but, in the third case, the associated operators need not be equal, and, in particular, the empty operator may appear. So we cannot deduce from the fact that  $w, w'$  represent the same element in  $\mathbf{Geom}_\mathcal{L}$  the fact that they represent the same element in  $\mathcal{Geom}_\mathcal{L}$ . However, and this is an important point, even if we cannot directly compare  $\mathcal{Geom}_\mathcal{L}$  and  $\mathbf{Geom}_\mathcal{L}$ , we can use the properties of  $\mathcal{Geom}_\mathcal{L}$  as a sort of oracle for guessing properties of  $\mathbf{Geom}_\mathcal{L}$ , and then try to find a direct proof for the results so conjectured.

### 3. INTERNALIZATION OF TERMS

At this point, we have have associated a monoid  $\mathcal{Geom}_\mathcal{L}$ , and in some cases a group  $\mathbf{Geom}_\mathcal{L}$ , with each family of algebraic laws  $\mathcal{L}$ . By construction, there is a partial action of the monoid  $\mathcal{Geom}_\mathcal{L}$  on terms, and similarly, at least in the

case of linear laws, a partial action of the group  $\mathbf{Geom}_{\mathcal{L}}$  on terms. The next step consists in trying to carry terms inside the monoid  $\mathcal{Geom}_{\mathcal{L}}$  and/or the group  $\mathbf{Geom}_{\mathcal{L}}$ , so as to replace the external action of the monoid or the group on terms by an internal operation, typically a multiplication in good cases. Once again, we have no general solution, but we explain how to complete the construction in good cases, and specifically in the case of the ALD-laws.

**3.1. Blueprint of a term: the principle.** A priori, terms and operators of the geometry monoid live in disjoint worlds: the only connection is that operators act on terms. The principle we shall apply in the sequel—and which turns out to be efficient in good cases—consists in building inside the geometry monoid  $\mathcal{Geom}_{\mathcal{L}}$ , and simultaneously inside the geometry group  $\mathbf{Geom}_{\mathcal{L}}$  that mimicks its algebraic properties, a copy of each term, so that the action of operators on terms translates into a simple operation inside  $\mathcal{Geom}_{\mathcal{L}}$  and  $\mathbf{Geom}_{\mathcal{L}}$ , typically a multiplication. This copy of a term  $t$  in  $\mathbf{Geom}_{\mathcal{L}}$  will be called the *blueprint* of  $t$ .

We recall that, for  $\mathcal{F}$  a list of operation symbols,  $\mathbf{Term}_{\mathcal{F}}$  denotes the family of all terms constructed using the operations of  $\mathcal{F}$  and variables from an infinite list  $\mathcal{X}$ . In the sequel, we assume that  $x$  is a fixed element of  $\mathcal{X}$ , and use  $\mathbf{Term}_{\mathcal{F},x}$  for the family of all terms constructed using the operations of  $\mathcal{F}$  and the single variable  $x$ .

The general principle is as follows. Assume that  $\mathcal{L}$  is a family of algebraic laws involving the signature  $\mathcal{F}$ , and we have an injective map (“carbon copy”)

$$(3.1) \quad \mathcal{C}c : \mathbf{Term}_{\mathcal{F},x} \rightarrow \mathcal{Geom}_{\mathcal{L}},$$

*i.e.*, a representation of terms inside the geometry monoid  $\mathcal{Geom}_{\mathcal{L}}$ . If  $t, t'$  are  $\mathcal{L}$ -equivalent terms, *i.e.*, by Proposition 1.7, if some element  $f$  of  $\mathcal{Geom}_{\mathcal{L}}$  maps  $t$  to  $t'$ , there must exist some relation between the copies  $\mathcal{C}c(t)$  and  $\mathcal{C}c(t')$ . We shall be interested in the case when this relation takes the form

$$(3.2) \quad \mathcal{C}c(t \cdot f) \sim \mathcal{C}c(t) \bullet \varphi(f)$$

where  $\varphi$  is some endomorphism of  $\mathcal{Geom}_{\mathcal{L}}$ , *i.e.*, when the action of  $\mathcal{Geom}_{\mathcal{L}}$  on terms becomes a right multiplication twisted by endomorphism at the level of the copies—optimally, we might hope for an equality in (3.2), but the problem of the empty operator makes a true equality impossible in most cases: this is precisely why we shall subsequently resort to the group  $\mathbf{Geom}_{\mathcal{L}}$ . We recall that  $\mathcal{W}_{\mathcal{L}}$  denotes the family of all abstract words in the letters  $L_{\alpha}^{\pm 1}$  for  $L$  in  $\mathcal{L}$  and  $\alpha$  an address. By construction, every operator in  $\mathcal{Geom}_{\mathcal{L}}$  is a finite product of elementary operators  $L_{\alpha}^{\pm 1}$ , so it can be expressed as  $eval(w)$  where  $w$  is a word in  $\mathcal{W}_{\mathcal{L}}$ , and (3.2) can be restated as

$$(3.3) \quad \mathcal{C}c(t \cdot eval(w)) \sim \mathcal{C}c(t) \bullet \varphi(eval(w)),$$

for  $t$  in  $\mathbf{Term}_{\mathcal{F},x}$  and  $w$  in  $\mathcal{W}_{\mathcal{L}}$  such that  $t \cdot eval(w)$  is defined.

Now, according to the general principle of Section 2, we wish to replace the monoid  $\mathcal{Geom}_{\mathcal{L}}$  with the group  $\mathbf{Geom}_{\mathcal{L}}$ . If the construction of the mapping  $\mathcal{C}c$  is explicit enough, we can mimick it in the group  $\mathbf{Geom}_{\mathcal{L}}$ , thus defining a similar (hopefully injective) mapping

$$(3.4) \quad \mathbf{C}c : \mathbf{Term}_{\mathcal{F},x} \rightarrow \mathbf{Geom}_{\mathcal{L}}$$

and use  $\mathbf{C}c(t)$  as a copy of  $t$  inside  $\mathbf{Geom}_{\mathcal{L}}$ . If  $\mathbf{Geom}_{\mathcal{L}}$  resembles  $\mathcal{Geom}_{\mathcal{L}}$  enough, we may hope that the counterpart of (3.3) holds in  $\mathbf{Geom}_{\mathcal{L}}$ , so that we can carry the action of  $\mathcal{Geom}_{\mathcal{L}}$  on terms inside the group  $\mathbf{Geom}_{\mathcal{L}}$ . We are thus led to the following notion:

**Definition 3.1.** Assume that  $\mathcal{L}$  is a family of oriented algebraic laws involving the signature  $\mathcal{F}$ , and that  $\phi$  is an endomorphism of  $\mathbf{Geom}_{\mathcal{L}}$ . A mapping  $\mathbf{Cc} : \mathbf{Term}_{\mathcal{F},x} \rightarrow \mathbf{Geom}_{\mathcal{L}}$  is said to be a  $\phi$ -*blueprint* if the relation

$$(3.5) \quad \mathbf{Cc}(t \cdot \mathit{eval}(w)) = \mathbf{Cc}(t) \cdot \phi(\mathit{eval}(w))$$

holds in  $\mathbf{Geom}_{\mathcal{L}}$  for every term  $t$  and every word  $w$  in  $\mathcal{W}_{\mathcal{L}}$  such that  $t \cdot \mathit{eval}(w)$  is defined.

Thus, a  $\phi$ -blueprint transforms the operation of applying the laws of  $\mathcal{L}$  into a multiplication twisted by  $\phi$  in the group  $\mathbf{Geom}_{\mathcal{L}}$ . The interest of a  $\phi$ -blueprint will be discussed in Section 4 below. The general idea is that it allows for carrying the problems about  $\mathcal{L}$ -equivalence inside the presented group  $\mathbf{Geom}_{\mathcal{L}}$ , and therefore may lead to solutions when the latter is under control. Typically, carrying the relation  $\leftrightarrow_{\mathcal{L}}$  to  $\mathbf{Geom}_{\mathcal{L}}$  using the map  $\mathbf{Cc}$  supposed to be injective yields a new equivalence relation inside  $\mathbf{Geom}_{\mathcal{L}}$  that may be more easily controlled than  $\leftrightarrow_{\mathcal{L}}$  itself.

**Remark 3.2.** A constant mapping provides a blueprint, certainly a trivial and uninteresting one. In the sequel, we are mainly interested in blueprints that are injective or close to, but we shall see that, in certain cases like that of ALD, even a blueprint that is not proved to be injective may be useful. That is why we do not require injectivity in Definition 3.1.

**3.2. Blueprint of a term: the example of ALD.** In order to realize the approach sketched above, *i.e.*, to construct a blueprint, in the case of the ALD-laws, the first step consists in representing terms in  $\mathcal{Geom}_{\mathcal{L}}$ , *i.e.*, in selecting for each term  $t$  a certain operator  $\mathbf{Cc}(t)$  in the geometry monoid  $\mathcal{Geom}_{\mathcal{L}}$  that in some sense characterizes  $t$ . A general idea is to choose an operator that *constructs*  $t$  starting from some fixed absolute startpoint. This procedure heavily depends on the specific algebraic laws we are investigating, here ALD. We use  $\mathbf{Term}_{*,\circ,x}$  for the family of all well-formed terms involving two binary operators  $*$ ,  $\circ$  and the single variable  $x$ .

The solution we develop relies on some specific property of the ALD-laws, namely the existence of an *absorption* phenomenon. To describe this phenomenon, let us define *right vines* to be those terms of  $\mathbf{Term}_{*,\circ,x}$  inductively specified by

$$(3.6) \quad x^{[n]} := \begin{cases} x & \text{for } n = 1, \\ x * x^{[n-1]} & \text{for } n \geq 2. \end{cases}$$

The associated trees are “all to the right” trees. The following result expresses that, in presence of the laws of ALD, every term  $t$  of  $\mathbf{Term}_{*,\circ,x}$  is absorbed by all sufficiently large right vines.

**Lemma 3.3** (absorption lemma). *For each term  $t$  in  $\mathbf{Term}_{*,\circ,x}$ , there exists a positive integer  $p$  such that*

$$(3.7) \quad x^{[n]} \leftrightarrow_{\text{ALD}} t * x^{[n-p]} \quad \text{holds for } n \text{ large enough.}$$

*Proof.* The property is true with  $p = 1$  for  $t = x$  and  $n \geq 2$  (in which case the equivalence is an equality), so, in order to establish it for every term in  $\mathbf{Term}_{*,\circ,x}$ , it is enough to show that, if (3.7) holds for  $t_1$  and  $t_2$ , then it holds for  $t_1 * t_2$  and for  $t_1 \circ t_2$  as well. So assume  $x^{[n]} \leftrightarrow_{\text{ALD}} t_1 * x^{[n-p_1]}$  for  $n \geq m_1$  and  $x^{[n]} \leftrightarrow_{\text{ALD}} t_2 * x^{[n-p_2]}$

for  $n \geq m_2$ . We obtain for  $n \geq \max(m_1 + p_2, m_2 + p_1)$

$$\begin{aligned}
x^{[n]} &\leftrightarrow_{\text{ALD}} t_1 * x^{[n-p_1]} && \text{by hypothesis,} \\
&\leftrightarrow_{\text{ALD}} t_1 * (t_2 * x^{[n-p_1-p_2]}) && \text{by hypothesis,} \\
&\leftrightarrow_{\text{ALD}} (t_1 * t_2) * (t_1 * x^{[n-p_1-p_2]}) && \text{by LD,} \\
&\leftrightarrow_{\text{ALD}} (t_1 * t_2) * x^{[n-p_2]} && \text{by hypothesis.}
\end{aligned}$$

Similarly, we have for  $n \geq \max(m_1, m_2 + p_1)$

$$\begin{aligned}
x^{[n]} &\leftrightarrow_{\text{ALD}} t_1 * x^{[n-p_1]} && \text{by hypothesis,} \\
&\leftrightarrow_{\text{ALD}} t_1 * (t_2 * x^{[n-p_1-p_2]}) && \text{by hypothesis,} \\
&\leftrightarrow_{\text{ALD}} (t_1 \circ t_2) * x^{[n-p_1-p_2]} && \text{by ALD}_2,
\end{aligned}$$

so the induction is completed.  $\square$

By Proposition 1.7, the equivalence of (3.7) must be witnessed for by some operator of the geometry monoid  $\mathcal{G}eom_{\text{ALD}}$ : for each term  $t$  in  $\text{Term}_{*,\circ,x}$  and every integer  $n$ , there must exist an operator  $\mathcal{C}c(t)$  in  $\mathcal{G}eom_{\text{ALD}}$ , hence a composition of operators  $\Sigma_\alpha^\pm$  and  $A_\alpha^\pm$ , that maps  $x^{[n]}$  to  $t * x^{[n-p]}$ . Actually, the inductive proof of Lemma 3.3 gives not only the existence of such a witness, but only an inductive construction for such a witness.

**Lemma 3.4.** *For  $t$  in  $\text{Term}_{*,\circ,x}$ , inductively define  $\mathcal{C}c(t)$  in  $\mathcal{G}eom_{\text{ALD}}$  by*

$$(3.8) \quad \mathcal{C}c(t) = \begin{cases} id & \text{for } t = x, \\ \mathcal{C}c(t_1) \bullet sh_1(\mathcal{C}c(t_2)) \bullet \Sigma_\varnothing^+ \bullet sh_1(\mathcal{C}c(t_1))^{-1} & \text{for } t = t_1 * t_2, \\ \mathcal{C}c(t_1) \bullet sh_1(\mathcal{C}c(t_2)) \bullet A_\varnothing^+ & \text{for } t = t_1 \circ t_2. \end{cases}$$

*Then, for every term  $t$  in  $\text{Term}_{*,\circ,x}$ , there exists  $p$  such that, for every  $n$  large enough, we have*

$$(3.9) \quad \mathcal{C}c(t) : x^{[n]} \mapsto t * x^{[n-p]}.$$

*Proof.* The formulas of (4.7) are a mere translation of the successive equivalence steps in the proof of Lemma 3.3, and the result is then a straightforward verification.  $\square$

Note that (3.9) guarantees that the mapping  $\mathcal{C}c$  is injective, since the term  $t$  can be recovered from the operator  $\mathcal{C}c(t)$ . So it is coherent to use the operator  $\mathcal{C}c(t)$  as a counterpart of the term  $t$  inside the geometry monoid  $\mathcal{G}eom_{\text{ALD}}$ . According to the general scheme of Section 3.1, we shall now analyse the counterpart of the action of  $\mathcal{G}eom_{\mathcal{L}}$  on terms.

**Lemma 3.5.** *For each term  $t$  in  $\text{Term}_{*,\circ,x}$  and each operator  $f$  in  $\mathcal{G}eom_{\text{ALD}}$  such that  $t \cdot f$  is defined, we have*

$$(3.10) \quad \mathcal{C}c(t \cdot f) \sim \mathcal{C}c(t) \bullet sh_0(f).$$

*Proof.* Let  $t' = t \cdot f$ . Then, for each term  $t_1$ , the operator  $sh_0(f)$  maps the term  $t * t_1$  to  $t' * t_1$ —and, similarly,  $t \circ t_1$  to  $(t \cdot f) \circ t_1$ . So, in particular,  $sh_0(f)$  maps  $t * x^{[n]}$  to  $t' * x^{[n]}$  for each  $n$ . Now, by Lemma 3.4 (and for  $n$  large enough), the operator  $\mathcal{C}c(t)$  maps  $x^{[n]}$  to  $t * x^{[n-p]}$ , while  $\mathcal{C}c(t')$  maps  $x^{[n]}$  to  $t' * x^{[n-p']}$  for some  $p'$ . This means that both  $\mathcal{C}c(t) \bullet sh_0(f)$  and  $\mathcal{C}c(t')$  map  $x^{[n]}$  to  $t' * x^{[n-p']}$ . Hence, in the

monoid  $\mathcal{G}eom_{\text{ALD}}$ , the two operators  $\mathcal{C}c(t) \bullet \mathbf{sh}_0(f)$  and  $\mathcal{C}c(t')$  agree on at least one term, namely  $t$ , which, by definition, means that (3.10) holds.  $\square$

We thus obtained in the case of ALD a relation of the form (3.2), the involved endomorphism of  $\mathcal{G}eom_{\text{ALD}}$  being  $sh_0$ . Expressed on the shape of (3.3), the relation reads

$$(3.11) \quad \mathcal{C}c(t \cdot \text{eval}(w)) \sim \mathcal{C}c(t) \bullet sh_0(\text{eval}(w))$$

for  $t$  a term in  $\text{Term}_{*,o,x}$  and  $w$  a word in  $\mathcal{W}_{\text{ALD}}$  such that  $t \cdot \text{eval}(w)$  is defined.

Following the general scheme of Section 3.1 again, we now mimick the construction of  $\mathcal{C}c$  inside the group  $\text{Geom}_{\text{ALD}}$ . Using  $\mathbf{sh}_\alpha$  to denote the endomorphism of  $\text{Geom}_{\text{ALD}}$  that maps  $\Sigma_\alpha$  to  $\Sigma_{1\alpha}$  and  $A_\alpha$  to  $A_{1\alpha}$  for each address  $\alpha$ , this amounts to setting:

**Definition 3.6.** We inductively associate with every term  $t$  in  $\text{Term}_{*,o,x}$  an element  $\mathcal{C}c(t)$  of  $\text{Geom}_{\text{ALD}}$ , also denoted  $\widehat{t}$ , by

$$(3.12) \quad \mathcal{C}c(t) = \begin{cases} 1 & \text{for } t = x, \\ \mathcal{C}c(t_1) \cdot \mathbf{sh}_1(\mathcal{C}c(t_2)) \cdot \Sigma_\emptyset \cdot \mathbf{sh}_1(\mathcal{C}c(t_1))^{-1} & \text{for } t = t_1 * t_2, \\ \mathcal{C}c(t_1) \cdot \mathbf{sh}_1(\mathcal{C}c(t_2)) \cdot A_\emptyset & \text{for } t = t_1 \circ t_2. \end{cases}$$

**Example 3.7.** Let  $t$  be  $x * ((x \circ x) * x)$ . Starting from  $\mathcal{C}c(x) = 1$ , we first find

$$\mathcal{C}c(x \circ x) = 1 \cdot \mathbf{sh}_1(1) \cdot A_\emptyset = A_\emptyset,$$

then

$$\mathcal{C}c((x \circ x) * x) = A_\emptyset \cdot \mathbf{sh}_1(1) \cdot \Sigma_\emptyset \cdot \mathbf{sh}_1(A_\emptyset)^{-1} = A_\emptyset \Sigma_\emptyset A_1^{-1},$$

and, finally,

$$\mathcal{C}c(t) = 1 \cdot \mathbf{sh}_1(A_\emptyset \Sigma_\emptyset A_1^{-1}) \cdot \Sigma_\emptyset \cdot \mathbf{sh}_1(1)^{-1} = A_1 \Sigma_1 A_{11}^{-1} \Sigma_\emptyset.$$

If our intuition is correct, *i.e.*, if the confluence relations defining the group  $\text{Geom}_{\text{ALD}}$  capture enough of the geometry of the laws ALD, Relation (3.11) should follow from the lattice relations in  $\mathcal{G}eom_{\text{ALD}}$ , and, therefore, it should induce an equality in the group  $\text{Geom}_{\text{ALD}}$ , *i.e.*, we *should* have the relation

$$(3.13) \quad \mathcal{C}c(t \cdot \text{eval}(w)) = \mathcal{C}c(t) \bullet \mathbf{sh}_0(\text{eval}(w)),$$

*i.e.*, with our former definition, the mapping  $\mathcal{C}c$  should be an  $\mathbf{sh}_0$ -blueprint. This is indeed the case—and this is the key point for our current analysis of the ALD-laws. The nice feature is that, if the result is true—and it is—its proof must be a simple verification.

**Lemma 3.8.** *The mapping  $\mathcal{C}c$  is a  $\mathbf{sh}_0$ -blueprint for the ALD-laws.*

*Proof.* For an induction on the length of the word  $w$ , it is enough to prove the result when  $w$  consists of a single letter, *i.e.*, it is one of  $A_\alpha^{\pm 1}$ ,  $\Sigma_\alpha^{\pm 1}$ . Moreover, the cases of  $A_\alpha^{-1}$  and  $\Sigma_\alpha^{-1}$  immediately follow from the cases of  $A_\alpha$  and  $\Sigma_\alpha$ , respectively. So, the point is to establish the equalities

$$(3.14) \quad \mathcal{C}c(t \cdot A_\alpha) = \mathcal{C}c(t) \cdot A_{0\alpha}, \quad \mathcal{C}c(t \cdot \Sigma_\alpha) = \mathcal{C}c(t) \cdot \Sigma_{0\alpha},$$

whenever the involved terms are defined.

We prove (3.14) using induction on the length of the address  $\alpha$ . Let us begin with  $\alpha = \emptyset$  and the case of  $A_\alpha$ , *i.e.*, of  $A_\emptyset$ . Saying that  $t \cdot A_\emptyset$  (*i.e.*,  $t \cdot A_\emptyset$ ) is defined, means that  $t$  can be decomposed as  $t = t_1 * (t_2 * t_3)$ , and, then, we

have  $t' = (t_1 \circ t_2) * t_3$ . Using the commutation and quasi-commutation relations of  $\text{Geom}_{\text{ALD}}$ , we find

$$\begin{aligned}\hat{t}' &= \hat{t}_1 \cdot \text{sh}_1(\hat{t}_2) \cdot A_\emptyset \cdot \text{sh}_1(\hat{t}_3) \cdot \Sigma_\emptyset \cdot A_1^{-1} \cdot \text{sh}_{11}(\hat{t}_2)^{-1} \cdot \text{sh}_1(\hat{t}_1)^{-1}, \\ &= \hat{t}_1 \cdot \text{sh}_1(\hat{t}_2) \cdot \text{sh}_{11}(\hat{t}_3) \cdot A_\emptyset \cdot \Sigma_\emptyset \cdot A_1^{-1} \cdot \text{sh}_{11}(\hat{t}_2)^{-1} \cdot \text{sh}_1(\hat{t}_1)^{-1}, \\ \hat{t} \cdot A_{0\alpha} &= \hat{t}_1 \cdot \text{sh}_1(\hat{t}_2) \cdot \text{sh}_{11}(\hat{t}_3) \cdot \Sigma_1 \cdot \text{sh}_{11}(\hat{t}_2)^{-1} \cdot \Sigma_\emptyset \cdot \text{sh}_1(\hat{t}_1)^{-1} \cdot A_0, \\ &= \hat{t}_1 \cdot \text{sh}_1(\hat{t}_2) \cdot \text{sh}_{11}(\hat{t}_3) \cdot \Sigma_1 \cdot \Sigma_\emptyset \cdot A_0 \cdot \text{sh}_{11}(\hat{t}_2)^{-1} \cdot \text{sh}_1(\hat{t}_1)^{-1},\end{aligned}$$

and the equality follows from (2.19), which gives  $A_\emptyset \cdot \Sigma_\emptyset \cdot A_1^{-1} = \Sigma_1 \cdot \Sigma_\emptyset \cdot A_0$ .

We consider now the case of  $\Sigma_\emptyset$ . The hypothesis that  $t \cdot \Sigma_\emptyset$  is defined means that  $t$  can be decomposed as  $t_1 * (t_2 \square t_3)$ , and, then,  $t'$  is  $(t_1 * t_2) \square (t_1 * t_3)$ . Assume first  $\square = *$  (this is the most complicated case). Using the commutation and quasi-commutation relations of  $\text{Geom}_{\text{ALD}}$ , we find

$$\begin{aligned}\hat{t}' &= \hat{t}_1 \cdot \text{sh}_1(\hat{t}_2) \cdot \Sigma_\emptyset \cdot \text{sh}_1(\hat{t}_1)^{-1} \cdot \text{sh}_1(\hat{t}_1) \cdot \text{sh}_{11}(\hat{t}_3) \cdot \Sigma_1 \cdot \text{sh}_{11}(\hat{t}_1)^{-1} \\ &\quad \cdot \Sigma_\emptyset \cdot \text{sh}_{11}(\hat{t}_1) \cdot \Sigma_1^{-1} \cdot \text{sh}_{11}(\hat{t}_2)^{-1} \cdot \text{sh}_1(\hat{t}_1)^{-1} \\ &= \hat{t}_1 \cdot \text{sh}_1(\hat{t}_2) \cdot \text{sh}_{11}(\hat{t}_3) \cdot \Sigma_\emptyset \cdot \Sigma_1 \cdot \Sigma_\emptyset \cdot \Sigma_1^{-1} \cdot \text{sh}_{11}(\hat{t}_2)^{-1} \cdot \text{sh}_1(\hat{t}_1)^{-1}, \\ \hat{t} \cdot \Sigma_{0\alpha} &= \hat{t}_1 \cdot \text{sh}_1(\hat{t}_2) \cdot \text{sh}_{11}(\hat{t}_3) \cdot \Sigma_1 \cdot \text{sh}_{11}(\hat{t}_2)^{-1} \cdot \Sigma_\emptyset \cdot \text{sh}_1(\hat{t}_1)^{-1} \cdot \Sigma_0 \\ &= \hat{t}_1 \cdot \text{sh}_1(\hat{t}_2) \cdot \text{sh}_{11}(\hat{t}_3) \cdot \Sigma_1 \cdot \Sigma_\emptyset \cdot \Sigma_0 \cdot \text{sh}_{11}(\hat{t}_2)^{-1} \cdot \text{sh}_1(\hat{t}_1)^{-1},\end{aligned}$$

and the equality follows from the (2.17) relation  $\Sigma_1 \cdot \Sigma_\emptyset \cdot \Sigma_1 \cdot \Sigma_0 = \Sigma_\emptyset \cdot \Sigma_1 \cdot \Sigma_\emptyset$ . Assume now  $\square = \circ$ . One finds

$$\begin{aligned}\hat{t}' &= \hat{t}_1 \cdot \text{sh}_1(\hat{t}_2) \cdot \Sigma_\emptyset \cdot \text{sh}_1(\hat{t}_1)^{-1} \cdot \text{sh}_1(\hat{t}_1) \cdot \text{sh}_{11}(\hat{t}_3) \cdot \Sigma_1 \cdot \text{sh}_{11}(\hat{t}_1)^{-1} \cdot A_\emptyset \\ &= \hat{t}_1 \cdot \text{sh}_1(\hat{t}_2) \cdot \text{sh}_{11}(\hat{t}_3) \cdot \Sigma_\emptyset \cdot \Sigma_1 \cdot A_\emptyset \cdot \text{sh}_1(\hat{t}_1)^{-1}, \\ \hat{t} \cdot \Sigma_{0\alpha} &= \hat{t}_1 \cdot \text{sh}_1(\hat{t}_2) \cdot \text{sh}_{11}(\hat{t}_3) \cdot A_1 \cdot \Sigma_\emptyset \cdot \text{sh}_1(\hat{t}_1)^{-1} \cdot \Sigma_0 \\ &= \hat{t}_1 \cdot \text{sh}_1(\hat{t}_2) \cdot \text{sh}_{11}(\hat{t}_3) \cdot A_1 \cdot \Sigma_\emptyset \cdot \Sigma_0 \cdot \text{sh}_1(\hat{t}_1)^{-1},\end{aligned}$$

and the equality follows from the (2.18) relation  $A_1 \cdot \Sigma_\emptyset \cdot \Sigma_0 = \Sigma_\emptyset \cdot \Sigma_1 \cdot A_\emptyset$ .

The case  $\alpha = \emptyset$  is completed. From now on, we shall treat the cases of  $A_\alpha$  and  $\Sigma_\alpha$  simultaneously, using  $L_\alpha$  as a common notation. Assume first that the address  $\alpha$  is  $0\beta$  for some  $\beta$ . The hypothesis that  $t \cdot L_\alpha$  is defined implies that  $t$  can be decomposed as  $t_1 \square t_2$ , and, then,  $t'$  is  $t'_1 \square t_2$ , with  $t'_1 = t_1 \cdot L_\beta$ . Assume first  $\square = *$ . The induction hypothesis implies  $\hat{t}'_1 = \hat{t}_1 \cdot L_{0\beta}$ . We find

$$\begin{aligned}\hat{t}' &= \hat{t}_1 \cdot L_{0\beta} \cdot \text{sh}_1(\hat{t}_2) \cdot \Sigma_\emptyset \cdot L_{10\beta}^{-1} \cdot \text{sh}_1(\hat{t}_1)^{-1}, \\ &= \hat{t}_1 \cdot \text{sh}_1(\hat{t}_2) \cdot L_{0\beta} \cdot \Sigma_\emptyset \cdot L_{10\beta}^{-1} \cdot \text{sh}_1(\hat{t}_1)^{-1}, \\ \hat{t} \cdot L_{0\alpha} &= \hat{t}_1 \cdot \text{sh}_1(\hat{t}_2) \cdot \Sigma_\emptyset \cdot \text{sh}_1(\hat{t}_1)^{-1} \cdot L_{00\beta}, \\ &= \hat{t}_1 \cdot \text{sh}_1(\hat{t}_2) \cdot \Sigma_\emptyset \cdot L_{00\beta} \cdot \text{sh}_1(\hat{t}_1)^{-1},\end{aligned}$$

and the equality follows from the (2.10) relation  $L_{0\beta} \cdot \Sigma_\emptyset = \Sigma_\emptyset \cdot L_{00\beta} \cdot L_{10\beta}$ . Similarly, for  $\square = \circ$ , we find

$$\begin{aligned}\hat{t}' &= \hat{t}_1 \cdot L_{0\beta} \cdot \text{sh}_1(\hat{t}_2) \cdot A_\emptyset = \hat{t}_1 \cdot \text{sh}_1(\hat{t}_2) \cdot L_{0\beta} \cdot A_\emptyset, \\ \hat{t} \cdot L_{0\alpha} &= \hat{t}_1 \cdot \text{sh}_1(\hat{t}_2) \cdot A_\emptyset \cdot L_{00\beta},\end{aligned}$$

and the equality follows from the (2.14) relation  $L_{0\beta} \cdot A_\emptyset = A_\emptyset \cdot L_{00\beta}$ .

The argument is similar when the address  $\alpha$  is  $1\beta$  for some  $\beta$ . The hypothesis that  $t \cdot L_\alpha$  is defined implies that  $t$  can be decomposed as  $t_1 \square t_2$ , and, then,  $t'$  is  $t_1 \square t'_2$ , with  $t'_2 = t_2 \cdot L_\beta$ . Assume first  $\square = *$ . The induction hypothesis implies  $\widehat{t}'_1 = \widehat{t}_1 \cdot L_{0\beta}$ . We find now

$$\begin{aligned}\widehat{t} \cdot L_{0\alpha} &= \widehat{t}_1 \cdot \mathbf{sh}_1(\widehat{t}_2) \cdot \Sigma_\emptyset \cdot \mathbf{sh}_1(\widehat{t}_1)^{-1} \cdot L_{01\beta} = \widehat{t}_1 \cdot \mathbf{sh}_1(\widehat{t}_2) \cdot \Sigma_\emptyset \cdot L_{01\beta} \cdot \mathbf{sh}_1(\widehat{t}_1)^{-1}, \\ \widehat{t}' &= \widehat{t}_1 \cdot \mathbf{sh}_1(\widehat{t}_2) \cdot L_{10\beta} \cdot \Sigma_\emptyset \cdot \mathbf{sh}_1(\widehat{t}_1)^{-1},\end{aligned}$$

and the equality follows from the (2.11) relation  $L_{10\beta} \cdot \Sigma_\emptyset = \Sigma_\emptyset \cdot L_{01\beta}$ . Finally, for  $\square = \circ$ , we have

$$\begin{aligned}\widehat{t} \cdot L_{0\alpha} &= \widehat{t}_1 \cdot \mathbf{sh}_1(\widehat{t}_2) \cdot A_\emptyset \cdot L_{01\beta}, \\ \widehat{t}' &= \widehat{t}_1 \cdot \mathbf{sh}_1(\widehat{t}_2) \cdot L_{01\beta} \cdot A_\emptyset,\end{aligned}$$

and the equality follows from the (2.15) relation  $L_{10\beta} \cdot A_\emptyset = A_\emptyset \cdot L_{01\beta}$ . The induction is complete.  $\square$

We thus completed the construction of a blueprint for the ALD-laws. It may be observed that this construction induces the construction of a similar  $\mathbf{sh}_0$ -blueprint for the LD-law considered alone: indeed, in the case of terms not containing the operator  $\circ$ , the blueprint does not involve any generator  $A_\alpha$ , and it can be checked that the only relations needed to check the blueprint condition are present in the group  $\mathbf{Geom}_{\text{LD}}$ .

**Remark 3.9.** Here the blueprints have been defined for terms in one variable only. Developing a similar approach for terms involving several variables is possible, at the expense of extending the geometry monoid  $\mathcal{Geom}_{\mathcal{L}}$  by introducing additional operators whose action is to shift the indices of the variables so as to still generate all terms starting from right vines. We refer to [9] for details in the specific case of the LD-law.

#### 4. USING $\mathbf{Geom}_{\mathcal{L}}$ TO STUDY $\mathcal{L}$

We claim that the geometry group  $\mathbf{Geom}_{\mathcal{L}}$  is an interesting object that provides useful information about the laws of  $\mathcal{L}$ . The way this vague statement can be made precise depends on the specific algebraic laws one considers. If the latter are simple, typically associativity and/or commutativity, constructing an  $\mathcal{L}$ -algebra or solving the word problem of  $\mathcal{L}$  is not a challenge, and in particular appealing to  $\mathbf{Geom}_{\mathcal{L}}$  is not necessary. However the group  $\mathbf{Geom}_{\mathcal{L}}$  may be of interest in itself, as the enormous literature devoted to R.Thompson's groups  $F$  and  $V$  shows [23, 20, 4]. In the case of complicated laws, typically self-distributivity or variants, constructing  $\mathcal{L}$ -algebras and solving the word problem of  $\mathcal{L}$  may be difficult (and often even open) questions, and then the geometry group may be useful. The basic scheme consists in exploiting the blueprint construction—when it exists—to define an algebraic system satisfying the laws of  $\mathcal{L}$  on some quotient of the group  $\mathbf{Geom}_{\mathcal{L}}$ .

**4.1. Construction of algebraic systems: principle.** The first, and more direct, application of the previous approach is the construction of an algebra that obeys some prescribed laws. The general principle is as follows:

**Proposition 4.1.** *Assume that  $\mathcal{L}$  is a family of balanced algebraic laws involving the signature  $\mathcal{F}$ , that  $\mathbf{Cc}$  is a  $\phi$ -blueprint for the laws  $\mathcal{L}$ , and that  $H$  is a subgroup of  $\mathbf{Geom}_{\mathcal{L}}$  that includes the image of  $\phi$ . For  $t$  a term in  $\mathbf{Term}_{\mathcal{F},x}$ , let  $\mathbf{cl}(t)$  be the*

left coset of  $\text{Cc}(t)$  modulo  $H$ . Then the map  $\text{cl}$  is constant on each  $\leftrightarrow_L$ -class, i.e.,  $\mathcal{L}$ -equivalent terms have the same image.

*Proof.* Assume that  $t$  and  $t'$  are  $\mathcal{L}$ -equivalent terms in  $\text{Term}_{*,\circ,x}$ . By Proposition 1.7, there must exist an operator in  $\text{Geom}_{\mathcal{L}}$  that maps  $t$  to  $t'$ , and, therefore, there must exist a word  $w$  in  $\mathcal{W}_{\mathcal{L}}$  such that  $t \cdot \text{eval}(w)$  is defined and equal to  $t'$ . Then, by (3.5), we have  $\text{Cc}(t') = \text{Cc}(t) \cdot \phi(\text{eval}(w))$  in  $\text{Geom}_{\mathcal{L}}$ , so the hypothesis gives

$$(\text{Cc}(t))^{-1} \cdot (\text{Cc}(t')) \in H,$$

i.e.,  $\text{Cc}(t)$  and  $\text{Cc}(t')$  lie in the same  $H$ -coset.  $\square$

**Corollary 4.2.** *Under the same hypothesis, let  $M$  be the image of the mapping  $\text{cl}$ . For each operation symbol  $\square$  in  $\mathcal{F}$ , define an operation in  $M$  by*

$$\text{Cc}(t)H \square \text{Cc}(t')H := \text{Cc}(t \square t')H.$$

*Then  $M$  equipped with these operations is an  $\mathcal{L}$ -algebra, i.e., an  $\mathcal{F}$ -structure satisfying all laws of  $\mathcal{L}$ .*

Note that, in general, nothing guarantees that the system so obtained is non-trivial: it might happen that the quotient-structure  $H \backslash \text{Geom}_{\mathcal{L}}$  collapses to a point. This does not happen in the cases we shall consider: on the contrary, the obtained algebraic systems will turn out to be free, i.e., as far from trivial as possible.

**4.2. Construction of algebraic systems: the example of ALD.** We return to our leading example, namely the ALD-laws. In Section 3 we constructed a  $\text{sh}_0$ -blueprint  $\text{Cc}$  for ALD, so Proposition 4.1 and Corollary 4.2 directly apply for each subgroup  $H$  of  $\text{Geom}_{\text{ALD}}$  including the image of  $\text{sh}_0$ , typically  $H := \text{sh}_0(\text{Geom}_{\text{ALD}})$ . We thus obtain an ALD-algebra whose domain is some subset of the coset set  $\text{sh}_0(\text{Geom}_{\text{ALD}}) \backslash \text{Geom}_{\text{ALD}}$ .

Actually, we can adapt the results to make them more simple and handy. Indeed, instead of restricting to the image of the mapping  $\text{cl}$ , we can extend the construction to the whole group  $\text{Geom}_{\text{ALD}}$ . To do that, the idea is obvious: we look at the inductive definition of the blueprint, and define  $*$  and  $\circ$  to be the operations used to construct  $\text{Cc}(t * t')$  and  $\text{Cc}(t \circ t')$  from  $\text{Cc}(t)$  and  $\text{Cc}(t')$ , i.e., we choose the operations on  $\text{Geom}_{\text{ALD}}$  that make  $\text{Cc}$  a morphism from the free algebra  $(\text{Term}_{*,\circ,x}, *, \circ)$  to  $\text{Geom}_{\text{ALD}}$ .

**Lemma 4.3.** *On the group  $\text{Geom}_{\text{ALD}}$  define two new binary operations  $*$ ,  $\circ$  by*

$$(4.1) \quad x * y := x \cdot \text{sh}_1(y) \cdot \Sigma_{\emptyset} \cdot \text{sh}_1(x)^{-1}, \quad x \circ y := x \cdot \text{sh}_1(y) \cdot A_{\emptyset}.$$

*Then, for all  $x, y, z$  in  $\text{Geom}_{\text{ALD}}$  and  $\square$  in  $\{*, \circ\}$ , we have*

$$(4.2) \quad (x * y) \square (x * z) = x * (y \square z) \cdot \Sigma_0,$$

$$(4.3) \quad (x \circ y) * z = x * (y * z) \cdot A_0,$$

$$(4.4) \quad (x \cdot \text{sh}_0(z)) \square y = x \square y \cdot \text{sh}_{00}(z),$$

$$(4.5) \quad x \square (y \cdot \text{sh}_0(z)) = x \square y \cdot \text{sh}_{01}(z).$$

*Proof.* The verifications are those already made in the proof of Lemma 3.8. The only difference is that, in Section 3, we only consider elements of  $\text{Geom}_{\text{ALD}}$  that are blueprints of terms, while, now, we consider arbitrary elements of  $\text{Geom}_{\text{ALD}}$ . Now inspecting the proof of Section 3 shows that the specific form of the elements is never used, and, therefore, the whole computation remains valid.  $\square$

Relations (4.2) and (4.3) control the obstruction for  $(\mathbf{Geom}_{\text{ALD}}, *, \circ)$  to be an ALD-algebra, and show that the latter belongs to the subgroup  $\mathbf{sh}_0(\mathbf{Geom}_{\text{ALD}})$ . Relations (4.4) and (4.5) show that the operations on  $\mathbf{Geom}_{\text{ALD}}$  induce well-defined operations on the coset set  $\mathbf{sh}_0(\mathbf{Geom}_{\text{ALD}}) \backslash \mathbf{Geom}_{\text{ALD}}$ . Thus we may state:

**Proposition 4.4.** *Let  $M$  be the coset set  $\mathbf{sh}_0(\mathbf{Geom}_{\text{ALD}}) \backslash \mathbf{Geom}_{\text{ALD}}$ . Then  $M$  equipped with the operations induced by those of Lemma 4.3 is an ALD-algebra.*

The subgroup  $\mathbf{sh}_0(\mathbf{Geom}_{\text{ALD}})$  is not normal, and, therefore, the associated coset set is not a group. When we replace  $\mathbf{sh}_0(\mathbf{Geom}_{\text{ALD}})$  with a larger subgroup  $H$  of  $\mathbf{Geom}_{\text{ALD}}$ , typically the normal subgroup generated by  $\mathbf{sh}_0(\mathbf{Geom}_{\text{ALD}})$ , we can still apply Proposition 4.1, but it is not a priori sure that the operations  $*$  and  $\circ$  induce well-defined operations on the whole of  $H \backslash \mathbf{Geom}_{\text{ALD}}$ . This however happens in good cases, as here with ALD.

**Lemma 4.5.** *Every normal subgroup of  $\mathbf{Geom}_{\text{ALD}}$  that includes the image of  $\mathbf{sh}_0$  contains all generators  $\Sigma_\alpha$  and  $A_\alpha$  such that  $\alpha$  contains at least one 0.*

*Proof.* For  $L = \Sigma$  or  $A$ , the commutation relation (2.15) gives  $L_{10} = A_\emptyset \cdot L_{01} \cdot A_\emptyset^{-1}$ , hence, inductively

$$L_{1^i 0 \alpha} = A_{1^{i-1}} \cdot \dots \cdot A_1 \cdot A_\emptyset \cdot L_{01^i \alpha} \cdot A_\emptyset^{-1} \cdot A_1^{-1} \cdot \dots \cdot A_{1^{i-1}}^{-1},$$

which shows that any normal subgroup containing all  $L_{0\gamma}$  must contain all  $L_\alpha$  such that  $\alpha$  contains at least one 0.  $\square$

Thus, collapsing all generators  $\Sigma_\alpha$  and  $A_\alpha$  such that  $\alpha$  begins with 0 in  $\mathbf{Geom}_{\text{ALD}}$  requires to collapse all generators  $L_\alpha$  such that  $\alpha$  contains at least one 0, in which case the quotient-group is generated by the images of the remaining generators, namely the generators  $\Sigma_{1^{i-1}}$  and  $A_{1^{i-1}}$  with  $i \geq 1$ . Considering what remains from the defining relations of  $\mathbf{Geom}_{\text{ALD}}$ , and using  $\sigma_i$  and  $a_i$  as simplified notation for  $\Sigma_{1^{i-1}}$  and  $A_{1^{i-1}}$ , we are led to the following group:

**Definition 4.6.** We let  $B_\bullet$  be the group generated two infinite sequences  $\sigma_1, \sigma_2, \dots$  and  $a_1, a_2, \dots$  of generators subject to the relations

$$(4.6) \quad \begin{cases} \sigma_i \sigma_j = \sigma_j \sigma_i, & \sigma_i a_j = a_j \sigma_i, & a_i a_{j-1} = a_j a_i, & a_i \sigma_{j-1} = \sigma_j a_i, \\ \sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1}, & \sigma_{i+1} \sigma_i a_{i+1} = a_i \sigma_i, & \sigma_i \sigma_{i+1} a_i = a_{i+1} \sigma_i \end{cases}$$

for  $i \geq 1$  and  $j \geq i + 2$ .

(We do not claim that  $B_\bullet$  is the quotient of  $\mathbf{Geom}_{\text{ALD}}$  by the normal subgroup generated by  $\mathbf{sh}_0(\mathbf{Geom}_{\text{ALD}})$ .) By construction, mapping  $\Sigma_{1^{i-1}}$  to  $\sigma_i$  and  $A_{1^{i-1}}$  to  $a_i$  defines a surjective homomorphism  $\pi$  of  $\mathbf{Geom}_{\text{ALD}}$  onto  $B_\bullet$  whose kernel includes  $\mathbf{sh}_0(\mathbf{Geom}_{\text{ALD}})$ , and therefore the normal subgroup  $N$  it generates. However, it might be that  $\pi$  collapses more than  $N$ , and proving that this does not happen would require a more complete algebraic study of the group  $\mathbf{Geom}_{\text{ALD}}$ , which is not our current aim. Here, the only thing we wish to observe that, by mimicking once again the construction of the blueprint, we can construct an ALD-structure on the group  $B_\bullet$ :

**Proposition 4.7.** [13, 14] *Let  $\mathbf{sh}$  denote the endomorphism of the group  $B_\bullet$  that maps  $\sigma_i$  to  $\sigma_{i+1}$  and  $a_i$  to  $a_{i+1}$  for every positive integer  $i$ . Define binary operations  $*, \circ$  on  $B_\bullet$  by*

$$(4.7) \quad x * y := x \cdot \mathbf{sh}(y) \cdot \sigma_1 \cdot \mathbf{sh}_1(x)^{-1}, \quad x \circ y := x \cdot \mathbf{sh}(y) \cdot a_1.$$

Then  $(B_\bullet, *, \circ)$  is an ALD-algebra. Moreover, this ALD-algebra is torsion-free, *i.e.*, every element of  $B_\bullet$  generates a free ALD-subsystem.

The group  $B_\bullet$  was introduced by M. Brin in [2, 3] and the author in [12] independently, and its elements have been interpreted in [13] as parenthesized braids, an refinement of standard Artin braids in which one takes into account the distances between the strands. The above results show that, in the context of the ALD-laws, the connection between  $\mathbf{Geom}_{\text{ALD}}$  and  $B_\bullet$  is similar to the connection between the geometry group of self-distributivity and the braid group  $B_\infty$ , as investigated in [9]. The benefit of the geometry monoid approach is to make the operations (4.7) natural and explain why they had to appear in this form and in this group.

**4.3. Presentation of  $\mathcal{Geom}_\mathcal{L}$ .** A different use of a blueprint is to allow for a closer comparison between the monoid  $\mathcal{Geom}_\mathcal{L}$  and the group  $\mathbf{Geom}_\mathcal{L}$  in the non-linear case. We observed that  $\mathbf{Geom}_\mathcal{L}$  is constructed by means of a list of confluence relations holding in  $\mathcal{Geom}_\mathcal{L}$ , but, in general, there is no reason why this list should be complete, *i.e.*, provide a presentation of  $\mathcal{Geom}_\mathcal{L}$ . Actually, when the empty operator belongs to  $\mathcal{Geom}_\mathcal{L}$ , it cannot be the case that the confluence relations exhaust all relations of  $\mathcal{Geom}_\mathcal{L}$  as no such relation involves  $\emptyset$ . However, the true question is to control the relations of  $\mathcal{Geom}_\mathcal{L}$  that do not involve the empty operator, which amounts to describing the relation  $\sim$  of Definition 2.6 on  $\mathcal{Geom}_\mathcal{L}$ . Then, we have the following solution:

**Proposition 4.8.** *Assume that  $\phi$  is injective and  $\mathbf{Cc}$  is a  $\phi$ -blueprint for the laws  $\mathcal{L}$ . Then the confluence relations used in the definition of  $\mathbf{Geom}_\mathcal{L}$  generate all non-trivial relations in  $\mathcal{Geom}_\mathcal{L}$  in the following sense: for all word  $w, w'$  in  $\mathcal{W}_\mathcal{L}$ , the relation  $\text{eval}(w) \sim \text{eval}(w')$  holds only if  $w$  and  $w'$  represent the same element of the group  $\mathbf{Geom}_\mathcal{L}$ .*

*Proof.* Assume that  $w, w'$  are words in  $\mathcal{W}_\mathcal{L}$  such that the associated operators  $\text{eval}(w)$  and  $\text{eval}(w')$  in  $\mathcal{Geom}_\mathcal{L}$  are connected by  $\sim$ , *i.e.*, there exists at least one term  $t$  on which they agree. Let  $t'$  be the common image of  $t$  under these operators. The hypothesis that  $\mathbf{Cc}$  is a  $\phi$ -blueprint gives

$$\mathbf{Cc}(t) \cdot \phi(\text{eval}(w)) = \mathbf{Cc}(t') = \mathbf{Cc}(t) \cdot \phi(\text{eval}(w'))$$

in  $\mathbf{Geom}_\mathcal{L}$ , whence  $\phi(\text{eval}(w)) = \phi(\text{eval}(w'))$ . If  $\phi$  is injective, we deduce  $\text{eval}(w) = \text{eval}(w')$ , *i.e.*, the words  $w$  and  $w'$  represent the same element of the group  $\mathbf{Geom}_\mathcal{L}$ .  $\square$

**Example 4.9.** We conjecture that the endomorphism  $\mathbf{sh}_0$  is injective on the group  $\mathbf{Geom}_{\text{ALD}}$ , which would imply that the confluence relations listed in Section 2.3 generate, in the sense described above, all relations holding in the geometry monoid  $\mathcal{Geom}_{\text{ALD}}$ .

When we restrict from ALD to LD alone, then the argument is similar, and, in that case, the injectivity of  $\mathbf{sh}_0$  on  $\mathbf{Geom}_{\text{LD}}$  is known. What makes the case of ALD more difficult is that, in the latter case, some confluence relations are missing, and the group  $\mathbf{Geom}_{\text{ALD}}$  is not a group of fractions for the positive monoid  $\mathbf{Geom}_{\text{ALD}}^+$ , while  $\mathbf{Geom}_{\text{LD}}$  is a group of fractions for  $\mathbf{Geom}_{\text{LD}}^+$ . In both cases, proving the injectivity of  $\mathbf{sh}_0$  on the positive monoid is easy, but extending the result to the group is not obvious when the latter is not a group of fractions.

**4.4. Solving the word problem.** Still another possibility is to use the blueprint for solving the word problem of  $\mathcal{L}$ , *i.e.*, to construct an algorithm that decides whether two terms  $t, t'$  are equivalent modulo the laws of  $\mathcal{L}$ .

**Proposition 4.10.** *Assume that  $\mathbf{Cc}$  is a Turing computable  $\phi$ -blueprint for the laws  $\mathcal{L}$ , that  $P, Q$  are disjoint recursively enumerable subsets of  $\mathbf{Geom}_{\mathcal{L}}$  such that  $P$  includes the image of  $\phi$ , and there is a binary relation  $\diamond$  on  $\mathbf{Term}_{\mathcal{F},x}$  such that, for all terms  $t, t'$ , at least one of  $t \leftrightarrow_{\mathcal{L}} t'$ ,  $t \diamond t'$  holds and  $t \diamond t'$  implies  $\mathbf{Cc}(t)^{-1}\mathbf{Cc}(t') \in Q$ . Then the word problem of the laws  $\mathcal{L}$  is solvable.*

*Proof.* Let  $\widehat{P}, \widehat{Q}$  be disjoint recursive sets that include  $P$  and  $Q$  respectively. First assume that  $t \leftrightarrow_{\mathcal{L}} t'$  holds. Under the hypotheses, this implies

$$\mathbf{Cc}(t)^{-1}\mathbf{Cc}(t') \in \text{Im}(\phi) \subseteq P \subseteq \widehat{P}.$$

Conversely, assume that  $t \leftrightarrow_{\mathcal{L}} t'$  fails. Then necessarily  $t \diamond t'$  holds, which implies

$$\mathbf{Cc}(t)^{-1}\mathbf{Cc}(t') \in Q \subseteq \widehat{Q},$$

so  $t \leftrightarrow_{\mathcal{L}} t'$  is equivalent to  $\mathbf{Cc}(t)^{-1}\mathbf{Cc}(t') \in \widehat{P}$ . As the function  $\mathbf{Cc}$  is assumed to be Turing computable and the set  $P$  is assumed to be Turing decidable, the latter condition is Turing decidable.  $\square$

**Example 4.11.** Let us consider the case of the LD-law. We know that there exists an  $\mathbf{sh}_0$ -blueprint  $\mathbf{Cc}$  for LD. Let  $t \diamond t'$  be the symmetric closure of the relation “ $t$  is LD-equivalent to some iterated left subterm of some term LD-equivalent to  $t'$ ”. Let  $P$  be the subset of  $\mathbf{Geom}_{\text{LD}}$  consisting of those elements that can be expressed using none of  $\Sigma_{\emptyset}, \Sigma_{\emptyset}^{-1}$ , and let  $Q$  be the subset of  $\mathbf{Geom}_{\text{LD}}$  consisting of those elements that can be expressed using exactly one of  $\Sigma_{\emptyset}, \Sigma_{\emptyset}^{-1}$ . It is easy to show that every element of  $\mathbf{Geom}_{\text{LD}}$  in the image of  $\mathbf{sh}_0$  belongs to  $P$ , while  $t \diamond t'$  implies  $\mathbf{Cc}(t)^{-1}\mathbf{Cc}(t') \in Q$ . Moreover, one can show that  $P$  and  $Q$  are disjoint and recursive, so Proposition 4.10 implies that the word problem of LD is solvable [9]. This was the first solution for a long standing open question—other solutions are known now.

A similar scheme was used in [10] to solve the word problem for the central duplication law  $x(yz) = (xy)(yz)$ —and, in this case, no alternative solution is known so far.

As for the word problem of the ALD-laws—for which a direct solution involving the group  $B_{\bullet}$  of Definition 4.6 is known—the scheme might work as well, but, as in Example 4.9, some pieces are missing as the group  $\mathbf{Geom}_{\text{ALD}}$  fails to be a group of fractions for the positive monoid  $\mathbf{Geom}_{\text{ALD}}^+$ , making the verification of certain technical conditions problematic, typically the fact the expected sets  $P, Q$  are disjoint.

## 5. PROVING GLOBAL CONFLUENCE

Let  $\mathcal{L}$  be a family of oriented algebraic laws. Then we have introduced both the geometry monoid  $\mathcal{G}eom_{\mathcal{L}}$ , which corresponds to the rewrite system  $R_{\mathcal{L}}$  and to using the laws of  $\mathcal{L}$  with either orientation, and the positive geometry monoid  $\mathcal{G}eom_{\mathcal{L}}^+$ , which corresponds to the rewrite system  $R_{\mathcal{L}}^+$  and to using the laws of  $\mathcal{L}$  in the distinguished direction only. An important case is when the system  $R_{\mathcal{L}}^+$  turns out to be confluent, and it is a technically significant issue to prove this confluence result when possible.

The standard way for proving a confluence result consists in checking local confluence and then using some noetherianity condition to conclude using the classical

Newman lemma—see for instance [1, 16]. However, it turns out that, in the current framework, the rules one considers are often ill-oriented, so that no noetherianity can be expected. The aim of this short section is to present an alternative method that can be used instead.

**5.1. Least common expansion.** Let us start with the example of the LD-law  $x*(y*z) = (x*y)*(x*z)$ . If we orient it in the contracting direction  $(x*y)*(x*z) \rightarrow x*(y*z)$ , then the rule diminishes the size and is therefore noetherian, but it is easily checked that confluence fails. When one chooses the expanding orientation, namely  $x*(y*z) \rightarrow (x*y)*(x*z)$ , then, as seen in Section 2.3, we obtain a locally confluent system as, for each pair of addresses  $\alpha, \beta$ , there exists one (and exactly one) confluence relation  $\Sigma_\alpha^+ \bullet \dots = \Sigma_\beta^+ \bullet \dots$ , where  $\Sigma_\alpha^+$  denotes “applying LD at position  $\alpha$  in the expanding direction”. Now, the rule increases the size of the terms, and it is not noetherian as, for instance, starting with the term  $x*(x*x)$ , we can apply  $\Sigma_\emptyset^+$  any number of times. So the general question is:

How to prove that the locally confluent system  $R_{\mathcal{L}}^+$  is possibly globally confluent (without assuming any noetherianity condition)?

Let us say that a term  $t'$  is a *degree  $d$   $\mathcal{L}$ -expansion* of a term  $t$  if  $t'$  is the image of  $t$  under an element of  $\mathcal{G}eom_{\mathcal{L}}^+$  that can be expressed as the product of at most  $d$  elementary operators  $L_\alpha^+$ . The method developed in [9] uses the following criterion:

**Proposition 5.1.** *Assume that there exists a mapping  $\partial : \mathbf{Term}_{\mathcal{F}} \rightarrow \mathbf{Term}_{\mathcal{F}}$  such that, for each term  $t$ , the term  $\partial t$  is an  $\mathcal{L}$ -expansion of all degree 1  $\mathcal{L}$ -expansions of  $t$  and, moreover, the mapping  $\partial$  is increasing w.r.t.  $\rightarrow_{\mathcal{L}}^+$ , i.e.,  $t \rightarrow_{\mathcal{L}}^+ t'$  implies  $\partial t \rightarrow_{\mathcal{L}}^+ \partial t'$ . Then  $R_{\mathcal{L}}^+$  is confluent.*

*Proof.* It is enough to prove that, for each  $d$ , the term  $\partial^d t$  is a common  $\mathcal{L}$ -expansion of  $t$  of all degree  $d$   $\mathcal{L}$ -expansions of  $t$ . We use induction on  $d$ . The result is trivial for  $d = 0$ , and it is true for  $d = 1$  by hypothesis. Assume  $d \geq 2$ , and let  $t'$  be a degree  $d$   $\mathcal{L}$ -expansion of  $t$ . By hypothesis, there exists  $t_1$  such that  $t_1$  is a degree  $d-1$  expansion of  $t$ , and  $t'$  is a degree 1 expansion of  $t_1$ . On the one hand,  $t'$  is a degree 1 expansion of  $t_1$ , hence  $\partial t_1$  is an expansion of  $t'$ . On the other hand, by induction hypothesis,  $\partial^{d-1} t$  is an expansion of  $t_1$ , hence  $\partial^d t$  is an expansion of  $\partial t_1$ . Being an expansion is transitive, hence  $\partial^d t$  is an expansion of  $t'$ , as expected.  $\square$

The criterion was first designed for the case of self-distributivity [9], but it was subsequently also applied in the case of central duplication [10], and in the case of idempotency with or without self-distributivity [18]. In each case, the construction of the operator  $\partial$  heavily relies on the considered laws.

**5.2. Group of fractions.** An alternative solution for proving the possible confluence of the positive rewrite system  $R_{\mathcal{L}}^+$  consists in working in the geometry monoid  $\mathcal{G}eom_{\mathcal{L}}$ , or, rather, in its positive submonoid  $\mathcal{G}eom_{\mathcal{L}}^+$ . Then we have the following criterion:

**Proposition 5.2.** *Assume that the monoid  $\mathcal{G}eom_{\mathcal{L}}^+$  admits common right multiples in the following strong sense: for all  $f, g$  in  $\mathcal{G}eom_{\mathcal{L}}^+$ , there exist  $f', g'$  satisfying  $f \bullet g' = g \bullet f'$  and, in addition, the domain of  $f \bullet g'$  is the intersection of the domains of  $f$  and  $g$ . Then the rewrite system  $R_{\mathcal{L}}^+$  is confluent.*

*Proof.* Write  $t \rightarrow_{\mathcal{L}}^+ t'$  if  $t$  rewrites to  $t'$  with respect to  $R_{\mathcal{L}}^+$ . Now assume  $t \rightarrow_{\mathcal{L}}^+ t'$  and  $t \rightarrow_{\mathcal{L}}^+ t''$ . By Proposition 1.7, there exist  $f, g$  in  $\mathcal{G}eom_{\mathcal{L}}^+$  satisfying  $t' = t \cdot f$  and  $t'' = t \cdot g$ . By Proposition 5.2, there exist  $f', g'$  in  $\mathcal{G}eom_{\mathcal{L}}^+$  satisfying  $f \bullet g' = g \bullet f'$  and such that the domain of  $f \bullet g'$  is the intersection of the domains of  $f$  and  $g$ , hence contains  $t$ . Then we have

$$t' \cdot g' = t \cdot (f \bullet g') = t \cdot (g \bullet f') = t'' \cdot f',$$

*i.e.*, letting  $t''' = t \cdot (f \bullet g')$ , we have  $t' \rightarrow_{\text{LD}}^+ t'''$  and  $t'' \rightarrow_{\text{LD}}^+ t'''$ .  $\square$

In good cases, Proposition 5.2 can be proved using the group  $\mathbf{Geom}_{\mathcal{L}}$ , or rather the positive monoid  $\mathbf{Geom}_{\mathcal{L}}^+$ , and its presentation: here we denote by  $\mathbf{Geom}_{\mathcal{L}}^+$  the monoid generated by generators  $L_{\alpha}$  for  $L$  in  $\mathcal{L}$  and  $\alpha$  an address, subject to all confluence relations connecting the operators  $L_{\alpha}^+$  in  $\mathcal{G}eom_{\mathcal{L}}$ , *i.e.*, the monoid that admits, as a monoid, the same presentation as the group  $\mathbf{Geom}_{\mathcal{L}}$ . Optimally, we could expect that the possible existence of common right multiples in  $\mathbf{Geom}_{\mathcal{L}}^+$  implies the existence of common right multiples in  $\mathcal{G}eom_{\mathcal{L}}$ , in the strong form required in Proposition 5.2. Always because of the empty operator, this need not be the case in general. Nevertheless, we have

**Proposition 5.3.** *Assume that  $\mathcal{L}$  consists of semi-linear laws, *i.e.*, laws  $l \rightarrow r$  with  $l$  an injective term (no variable repeated), and  $\mathbf{Geom}_{\mathcal{L}}^+$  admits common right multiples. Then  $\mathcal{G}eom_{\mathcal{L}}^+$  admits common right multiples and Proposition 5.2 applies.*

*Sketch of proof.* Under the hypotheses, the empty operator does not appear in  $\mathcal{G}eom_{\mathcal{L}}^+$ , and it follows that  $\mathcal{G}eom_{\mathcal{L}}^+$  is a homomorphic image of  $\mathbf{Geom}_{\mathcal{L}}^+$ . Thus every relation in  $\mathbf{Geom}_{\mathcal{L}}^+$  projects into a relation in  $\mathcal{G}eom_{\mathcal{L}}^+$ , and, in particular, common multiples in  $\mathbf{Geom}_{\mathcal{L}}^+$  induce common multiples in  $\mathcal{G}eom_{\mathcal{L}}^+$ .  $\square$

So the question remains of proving the existence of common right multiples in the monoid  $\mathbf{Geom}_{\mathcal{L}}^+$ . By hypothesis, we start with a presented monoid where the relations are of the form  $L_{\alpha} \cdot \dots = L'_{\beta} \cdot \dots$ , *i.e.*, they assert the existence of some common right multiples, and what we need is to extend the result so as to show that any two elements of the monoid admit a common right multiple. This is easy if all relations in the presentation turns out to have length at most 2 for, in that case, an easy induction gives a common multiple of length at most  $p+q$  for any two elements that can be expressed by words of length  $p$  and  $q$ . In more complicated cases—typically in the case of the LD-law, where we have seen some confluence relations involve words of length 3 or 4, like  $\Sigma_{\emptyset}\Sigma_1\Sigma_{\emptyset} = \Sigma_1\Sigma_{\emptyset}\Sigma_1\Sigma_0$ —the method consists in finding a family of words  $X$  that includes all generators  $L_{\alpha}$  and is closed under complement in the sense that, for all words  $u, v$  in  $X$ , there exists  $u', v'$  in  $X$  such that both  $uv'$  and  $vu'$  represent a common right multiple of the elements of  $\mathbf{Geom}_{\mathcal{L}}^+$  represented by  $u$  and  $v$ . For instance, this approach works for LD, but it is rather delicate and leads to an upper bound which, instead of  $p+q$  as above, is a tower of exponentials of height  $2^{p+q}$  [9].

When one can prove that common (right) multiples exist in the monoid  $\mathbf{Geom}_{\mathcal{L}}^+$ , and, that, in addition, the latter admits cancellation, then standard results [5] guarantee that the group  $\mathbf{Geom}_{\mathcal{L}}$  is a group of fractions for  $\mathbf{Geom}_{\mathcal{L}}^+$ , and this turns out to be crucial in the study of  $\mathbf{Geom}_{\mathcal{L}}$ . Again, proving that  $\mathbf{Geom}_{\mathcal{L}}^+$  is cancellative requires specific tools connected with the so-called word reversing method [11]. In many good cases (self-distributivity, associativity, central duplication), one can

show that the divisibility relation gives  $\mathbf{Geom}_{\mathcal{L}}^+$  the structure of a lattice—but this is beyond the scope of this paper.

## 6. SUMMARY

We thus showed how introducing by means of confluence relations an abstract group that is supposed to mimic the properties of the geometry monoid—hence of the initial rewrite system—and then internalizing terms in that group so as to transform the initial external action into an internal multiplication may allow to solve nontrivial questions about a given family of algebraic laws.

In the case of non linear algebraic laws, *i.e.*, when some variable is repeated at least twice, due to the problem of the empty operator, the geometry monoid is an intrinsically inconvenient object, and our approach for replacing it with a group is the only known one. Of course, one might consider other relations than the confluence relations. The latter proved to be suitable for the examples considered here, but different schemes might prove relevant for other identities: for instance, when commutativity is involved, the operators are involutive, and the associated relations  $L_{\alpha}^2 = 1$  are not confluence relations. In some cases, one can keep the principle of introducing the structure  $\mathbf{Geom}_{\mathcal{L}}$  presented by the confluence relations in  $\mathcal{G}eom_{\mathcal{L}}$ , but taking  $\mathbf{Geom}_{\mathcal{L}}$  to be a monoid rather than a group: typically, this has to be done in the case of the *idempotency law*  $x = xx$ , as, in this case, confluence relations of the type  $L_{\emptyset}^+ \bullet L_{\emptyset}^+ = L_{\emptyset}^+ \bullet L_0^+ \bullet L_1^+$  are satisfied, preventing  $\mathcal{G}eom_{\mathcal{L}}$  from admitting left cancellation—see [17]. So we see that the methods described here require some flexibility in their application.

However, it should be possible to adapt all three main steps, namely introducing a monoid of partial operators, replacing it with a group using a presentation, and internalizing the rewrite system by representing the objects on which the initial action is defined by copies inside the group, to more general frameworks. An example is given in [12] where algebraic laws are replaced with a more complicated action on terms (“twisted commutativity”); we think that more rewrite systems, possibly of a completely different type, could be investigated using similar tools.

## REFERENCES

- [1] R. Book, *Thue systems as rewriting systems*, J. Symb. Comput. **3** (1987) 39–68.
- [2] M. Brin, *The algebra of strand splitting. I. A braided version of Thompson’s group V*, arXiv math.GR/040642.
- [3] M. Brin, *The algebra of strand splitting. II. A presentation for the braid group on one strand*, Intern. J. of Algebra and Computation **16** (2006) 203–219.
- [4] J.W. Cannon, W.J. Floyd, & W.R. Parry, *Introductory notes on Richard Thompson’s groups*, Ens. Math. **42** (1996) 215–257.
- [5] A.H. Clifford & G.B. Preston, *The Algebraic Theory of Semigroups, vol. 1*, AMS Surveys **7**, (1961).
- [6] D.E. Cohen, *String rewriting, a survey for group theorists*. in G.A. Niblo, M.A. Roller eds., Geometric Group Theory, vol. 1, London Math. Soc. Lect. Note Ser. **181** (1993) 37–47.
- [7] P. Dehornoy, *Structural monoids associated to equational varieties*, Proc. Amer. Math. Soc. **117-2** (1993) 293–304.
- [8] P. Dehornoy, *The structure group for the associativity identity*, J. Pure Appl. Algebra **111** (1996) 59–82.
- [9] P. Dehornoy, *Braids and Self-Distributivity*, Progress in Math. vol. 192, Birkhäuser (2000).
- [10] P. Dehornoy, *Study of an identity*, Alg. Universalis **48** (2002) 223–248.
- [11] P. Dehornoy, *Complete positive group presentations*, J. of Algebra **268** (2003) 156–197.

- [12] P. Dehornoy, *Geometric presentations of Thompson's groups*, J. Pure Appl. Algebra **203** (2005) 1–44.
- [13] P. Dehornoy, *The group of parenthesized braids*, Advances in Math. **205** (2006) 354–409.
- [14] P. Dehornoy, *Free augmented LD-systems*, arXiv: math.GR/0507196; J. Algebra & Appl., in press.
- [15] N. Dershowitz & J.P. Jouannaud, *Rewrite Systems*, in Handbook of Theor. Comp. Sc. vol. B, J. van Leeuwen ed., Elsevier (1994).
- [16] G. Huet, *Confluent reductions: Abstract properties and applications to term rewriting systems*, Journal of the ACM **27** (1980) 797–821.
- [17] P. Jedlicka, *Geometry monoid of left distributivity and left idempotency*, J. Algebra and Discrete Math., in press.
- [18] D. Larue, *Left-distributive and left-distributive idempotent algebras*, PhD Thesis, University of Colorado, Boulder (1994).
- [19] S. Mac Lane, *Natural associativity and commutativity*, Rice Univ. Studies **49** (1963) 28–46
- [20] R. McKenzie & R.J. Thompson, *An elementary construction of unsolvable word problems in group theory*, in Word Problems, Boone & al. eds., North Holland, Studies in Logic vol. 71 (1973).
- [21] A. Paterson, *Groupoids, inverse semigroups, and their operator algebras*, Progress in Math. vol. 170, Birkhäuser (1998).
- [22] J. Robinson, *A machine-oriented logic based on the resolution principle*, J. of the ACM **10** (1963) 163–174.
- [23] R.J. Thompson, *Embeddings into finitely generated simple groups which preserve the word problem*, in Word problems II: The Oxford book, S.I. Adjan & al. eds., Studies in Logic vol. 95, North Holland (1980) 401–440.

LABORATOIRE DE MATHÉMATIQUES NICOLAS ORESME UMR 6139, UNIVERSITÉ DE CAEN, 14032 CAEN, FRANCE

*E-mail address:* [dehornoy@math.unicaen.fr](mailto:dehornoy@math.unicaen.fr)

*URL:* [//www.math.unicaen.fr/~dehornoy](http://www.math.unicaen.fr/~dehornoy)