SOME ASPECTS OF THE SD-WORLD

PATRICK DEHORNOY

Abstract. We survey a few of the many results now known about the self-distributivity law and selfdistributive structures, with a special emphasis on the associated word problems and the algorithms solving them in good cases.

Selfdistributivity (SD) is the algebraic law stating that a binary operation is distributive with respect to itself. It comes in two versions: *left* self-distributivity $x(yz) = (xy)(xz)$, also called LD, and *right* self-distributivity $(xy)z = (xz)(yz)$, also called RD. Their properties are, of course, entirely symmetric; however, because both versions occur in important examples (see below), it seems difficult to definitely choose one of them and stick to it. To avoid ambiguity, it is better to use an oriented operation symbol: we shall follow the excellent convention, now widely adopted, of using $\triangleright$ for LD and $\triangleleft$ for RD, both coherent with the intuition that the operation is an action on the term to which the triangle points. Then SD means that the action is compatible with the operation, and it takes the forms

$$x \triangleright (y \triangleright z) = (x \triangleright y) \triangleright (x \triangleright z), \tag{LD}$$

$$(x \triangleleft y) \triangleleft z = (x \triangleleft z) \triangleleft (y \triangleleft z). \tag{RD}$$

Although selfdistributivity syntactically resembles associativity—only one letter separates $x(yz) = (xy)(xz)$ from $x(yz) = (xy)z$—, their properties are quite different, selfdistributivity turning out to be much more complicated: even the solution of the word problem and the description of free structures is highly nontrivial.

Selfdistributivity has been considered explicitly as early as the end of the XIXth century [55], and selfdistributive structures have been extensively investigated, specially around Belousov in Kichinev from the late 1950s [1, 2] and around Ježek, Kepka, and Němec in Prague from the 1970s, with a number of structural results, in particular about two-sided selfdistributivity, see [41]. The interest in selfdistributivity was renewed and reinforced in the 1980s by the discovery of connections with low-dimensional topology by D. Joyce [42] and S. Matveev [51], and with the theory of large cardinals in set theory by R. Laver [46] and the current author [8]. More recently, the connection with topology was made more striking by the cohomological approach proposed by R. Fenn, D. Rourke, and B. Sanderson [36], considerably developed from the 1990s in work by J.S. Carter, S. Kamada, and other authors, see [5, 6, 7, 35]; the study of quandles has now become a full subject, with intertwined papers both on the topological and on the algebraic side.

The aim of this text is to survey some aspects of selfdistributive algebra, with a special emphasis on the involved word problems. A comprehensive reference is the monograph [19], which is not always easy to read. We hope that this text will be

1991 *Mathematics Subject Classification.* 20N02, 03D40, 08A50.

Key words and phrases. selfdistributivity, shelf, rack, quandle, spindle, word problem.

more reader-friendly and might inspire further research (several open questions are mentioned).

The paper is organized in four sections. In the first section, we recall the now standard terminology and mention a few examples of selfdistributive structures, some classical, some more exotic. In Sections 2 and 3, we survey the many results involving the word problem in the case of selfdistributivity alone. Finally, in Section 4, we similarly address the (easy) cases of racks and quandles and the (frustrating) case of spindles.

1. Shelves, spindles, racks, and quandles

Selfdistributive structures abound, and we begin with a few pictures from the SD-world. After recalling the usual terminology (Section 1.1), we mention the classical examples (Section 1.2), and some more exotic ones (Section 1.3). The description is summarized in a sort of chart of the SD-world (Section 1.4).

1.1. **Terminology.** Throughout the paper, we use the now well established terminology introduced in topology.

Definition 1.1. *A* shelf *(or* right-shelf*) is a structure $(S, \lhd)$, where $\lhd$ is a binary operation on a (non-empty) set S that obeys the right selfdistributivity law* RD*. Symmetrically, a* left-shelf *is a structure $(S, \rhd)$, with $\rhd$ obeying the left selfdistributivity law* LD*.*

Definition 1.2. *A* spindle *is a shelf $(S, \lhd)$, in which the operation $\lhd$ obeys the idempotency law $x \lhd x = x$.*

Definition 1.3. *A* rack *is a shelf $(S, \lhd)$, in which right translations are bijective, i.e., for every b in S, the map $R_b : x \mapsto x \lhd b$ is a bijection from S to itself.*

Of course, *left-racks* are left-shelves $(S, \rhd)$ with bijective left translations.

Racks can equivalently be defined as structures involving two operations:

Proposition 1.4. *If $(S, \lhd)$ is a rack, and, for b, c in S, we let $c \overline{\lhd} b$ be the (unique) element a satisfying $a \lhd b = c$, then $\overline{\lhd}$ also obeys RD, and $\lhd$ and $\overline{\lhd}$ are connected by the mixed laws*

$$(x \lhd y) \mathbin{\overline{\lhd}} y = (x \mathbin{\overline{\lhd}} y) \lhd y = x. \tag{1.1}$$

Conversely, if $(S, \lhd)$ is a shelf and there exists $\overline{\lhd}$ satisfying (1.1)*, then $(S, \lhd)$ is a rack and $\overline{\lhd}$ is the second operation associated with $\lhd$ as above.*

The proof is an easy verification. Note that, in the situation of Prop. 1.4, each of the operations $\lhd, \overline{\lhd}$ is distributive with respect to the other: (1.1) implies

$$((x \lhd y) \mathbin{\overline{\lhd}} z) \lhd z = x \lhd y, \text{ and}$$
$$((x \mathbin{\overline{\lhd}} z) \lhd (y \mathbin{\overline{\lhd}} z)) \lhd z = (((x \mathbin{\overline{\lhd}} z) \lhd z) \lhd ((y \mathbin{\overline{\lhd}} z)) \lhd z) = x \lhd y,$$

whence $(x \lhd y) \mathbin{\overline{\lhd}} z = (x \mathbin{\overline{\lhd}} z) \lhd (y \mathbin{\overline{\lhd}} z)$, whenever right translations of $\lhd$ are injective.

Definition 1.5. *A* quandle *is a rack $(S, \lhd)$, which is also a spindle, i.e., the operation $\lhd$ obeys the idempotency law $x \lhd x = x$.*

As the examples below show, a rack need not be a quandle; however, a rack is always rather close to a quandle, in that the actions of an element and its square coincide: every rack satisfies the law $x \lhd y = x \lhd (y \lhd y)$, as shows the computation

$$x \lhd (y \lhd y) = ((x \mathbin{\overline{\lhd}} y) \lhd y) \lhd (y \lhd y) = ((x \mathbin{\overline{\lhd}} y) \lhd y) \lhd y = x \lhd y.$$

1.2. **Classical examples.**

Example 1.6 (**trivial shelves**). *Let S be any set, and f be any map from S to itself. Then defining $a \lhd_f b := f(a)$ provides a selfdistributive operation on S. The shelf $(S, \lhd_f)$ is a spindle if, and only if, f is the identity map; it is a rack if, and only if, f is a bijection. Special cases are the* **cyclic racks** Cycl_n *corresponding to $S := \mathbb{Z}/n\mathbb{Z}$ with $a \lhd b := a+1$, and the* **augmentation rack** $\mathsf{Aug}(\mathbb{Z})$*, corresponding to $S := \mathbb{Z}$ with $a \lhd b := a+1$ again.*

Example 1.7 (**lattice spindles**). *If $(L, \wedge, \vee, 0, 1)$ is a lattice, then both $(L, \wedge)$ and $(L, \vee)$ are spindles. Moreover, they are both right- and left-spindles in the obvious sense, since the operations are commutative. Whenever L has at least two elements, these spindles are not racks: for every a in L, we have $0 \vee a = a \vee a = a$, so the right translation associated with a non-zero element is never injective.*

Example 1.8 (**Boolean shelves**). *Let $(B, \wedge, \vee, 0, 1, {}^-)$ be a Boolean algebra (i.e., a lattice that is distributive and complemented). For a, b in B, define $a \lhd b := a \vee \bar{b}$. Then $(B, \lhd)$ is a shelf, as we find*

$$\begin{aligned}(x \lhd y) \lhd z = x \vee \bar{y} \vee \bar{z} = (x \vee \bar{y} \vee \bar{z}) \wedge 1 = ((x \vee \bar{z}) \vee \bar{y}) \wedge ((x \vee \bar{z}) \vee z)\\ = (x \vee \bar{z}) \vee (\bar{y} \wedge z) = (x \vee \bar{z}) \vee (\overline{y \vee \bar{z}}) = (x \lhd z) \lhd (y \lhd z).\end{aligned}$$

This shelf is neither a spindle, nor a rack for $\#B \geqslant 2$ as, for $a \neq 1$, we have $a \lhd a = a \vee \bar{a} = 1$, and $a \lhd a = 1 = 1 \lhd a$. Note that, under the standard logical interpretation, $\lhd$ corresponds to a reverse implication $\Leftarrow$.

Example 1.9 (**Alexander spindles**). *Let R be a ring and t belong to R. Consider an R-module E. For a, b in E, define $a \lhd b := ta + (1-t)b$. Then $(E, \lhd)$ is a spindle, as $\lhd$ is idempotent and we find*

$$(x \lhd y) \lhd z = t^2 x + (t - t^2)y + (1-t)z = (x \lhd z) \lhd (y \lhd z).$$

This spindle is a quandle if, and only if, t is invertible in R, with the second operation then defined by $c \mathbin{\overline{\lhd}} b := t^{-1}c + (1 - t^{-1})b$.

Example 1.10 (**conjugacy quandles**). *Let G be a group. Define*

$$a \lhd b := b^{-1}ab \tag{1.2}$$

for a, b in G. Then $(G, \lhd)$ is a quandle, as $\lhd$ is idempotent and we find

$$(a \lhd b) \lhd c = c^{-1}b^{-1}abc = (a \lhd c) \lhd (b \lhd c).$$

The second operation is then given by

$$c \mathbin{\overline{\lhd}} b := bcb^{-1}. \tag{1.3}$$

Hereafter, these structures are denoted by $\mathsf{Conj}_{\lhd}(G)$ and $\mathsf{Conj}_{\lhd,\overline{\lhd}}(G)$, according to whether we consider the one operation or the two operations version.

Variants are obtained by defining $a \lhd b := b^{-n}ab^n$ for n a fixed integer, or $a \lhd b := \phi(b^{-1}a)b$, where ϕ is a fixed automorphism of G (one obtains a spindle whenever ϕ is an endomorphism).

Example 1.11 (**core**, or **sandwich**, **quandles**). *Let G be a group. For a, b in G, define $a \lhd b := ba^{-1}b$. Then $(G, \lhd)$ is a quandle, as we find*

$$(a \lhd b) \lhd c = cb^{-1}ab^{-1}c = (a \lhd c) \lhd (b \lhd c).$$

The second operation coincides with the first one ("involutory" quandle). Again variants are possible, involving powers or an involutory automorphism.

1.3. **More exotic examples.** We complete the overview with a few less classical examples. Note that all of them are variations around conjugation.

Example 1.12 (**half-conjugacy racks**). *Let G be a group, and let X be a subset of G. For a, b in G and x, y in X, define*

$$(x, a) \lhd (y, b) := (x, ab^{-1}yb). \tag{1.4}$$

Then $(X \times G, \lhd)$ is a rack, as we find

$$((x, a) \lhd (y, b)) \lhd (z, c) = (x, ab^{-1}ybc^{-1}zc) = ((x, a) \lhd (z, c)) \lhd ((y, b) \lhd (z, c)).$$

This structure is denoted by $\mathsf{HalfConj}(X, G)$. *The second operation is then given by*

$$(z, c) \,\overline{\lhd}\, (y, b) := (z, cb^{-1}y^{-1}b). \tag{1.5}$$

The name is natural, in that, when G is a free group based on X, the reduced words representing the elements of the associated conjugacy quandle are palindroms $a^{-1} \cdot x \cdot a$, and half-conjugacy then corresponds to extracting x and the "half-word" a. The main difference between conjugacy and half-conjugacy is that the latter need not be idempotent: for x in X, one finds $(x, 1) \lhd (x, 1) = (x, x)$ and, more generally, $(x, 1)^{[n]} = (x, x^{n-1})$ for $n \geqslant 1$—see Notation 2.7 for the definition of $a^{[n]}$. Note that, for for $G = (\mathbb{Z}, +)$ and $X = \{1\}$, one finds $\mathsf{HalfConj}(X, G) \simeq \mathsf{Aug}(\mathbb{Z})$.

Example 1.13 (**injection shelves**). *Let X be a non-empty set, and let $\mathfrak{S}_X$ be the group of all bijections from X to itself. Then one can consider the quandle* $\mathsf{Conj}(\mathfrak{S}_X)$ *as described in Example 1.10. When the group $\mathfrak{S}_X$ is replaced with the monoid $\mathfrak{I}_X$ of all injections from X to itself, conjugacy makes no more sense, but defining*

$$f \lhd g(x) := g(f(g^{-1}(x))) \text{ for } x \in \mathsf{Im}(g), \text{ and } f \lhd g(x) := x \text{ otherwise}$$

still provides a shelf, denoted by $\mathsf{Conj}(\mathfrak{I}_X)$. *If X is finite, $\mathfrak{I}_X$ coincides with $\mathfrak{S}_X$, and* $\mathsf{Conj}(\mathfrak{I}_X)$ *is the quandle of Example 1.10. But, if X is infinite, the inclusion of $\mathfrak{I}_X$ in $\mathfrak{S}_X$ is strict, and the shelf* $\mathsf{Conj}(\mathfrak{I}_X)$ *is neither a spindle nor a rack. Denoting $X \setminus \mathsf{Im}(f)$ by $\mathsf{coIm}(f)$, one easily checks in* $\mathsf{Conj}(\mathfrak{I}_X)$ *the following equalities*

$$\mathsf{Im}(f \lhd g) = g(\mathsf{Im}(f)) \cup \mathsf{coIm}(g) \quad \text{and} \quad \mathsf{coIm}(f \lhd g) = g(\mathsf{coIm}(f)). \tag{1.6}$$

As a typical example, consider $X := \mathbb{Z}_{>0}$, let sh *be the* shift *mapping $n \mapsto n + 1$, and let* Shift *be the subshelf of* $\mathsf{Conj}(\mathfrak{I}_X)$ *generated by* sh, *see Fig. 1. Then* Shift *is not a spindle, as we have* $\mathsf{sh} \lhd \mathsf{sh}(1) = 1 \neq \mathsf{sh}(1) = 2$. *It is not a rack, either, as we have* $\mathsf{coIm}(\mathsf{sh}) = \{1\}$ *and, by* (1.6), $\mathsf{coIm}(f \lhd \mathsf{sh}) = \mathsf{sh}(\mathsf{coIm}(f)) \subseteq \{2, 3, ...,\}$ *for every f, so the right translation by* sh *is not surjective.*

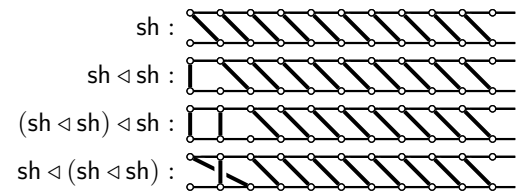


FIGURE 1. A few elements of the shelf Shift, which is neither a spindle, nor a rack; here, injections go from the top line to the bottom one.

Although its definition is quite simple, very little is known so far about Shift, see [9]. It is easy to deduce from the criterion of Lemma 2.19 below that Shift is not a free shelf. However, almost nothing is known about the following problem:

Question 1.14. *Find a presentation of the shelf* Shift.

Example 1.15 (**braid shelf**). *Artin's braid group B_∞ is the group defined by the (infinite) presentation*

$$\left\langle \sigma_1, \sigma_2, \dots \;\middle|\; \begin{array}{lll} \sigma_i\sigma_j = \sigma_j\sigma_i & \text{for} & |i-j| \geqslant 2 \\ \sigma_i\sigma_j\sigma_i = \sigma_j\sigma_i\sigma_j & \text{for} & |i-j| = 1 \end{array} \right\rangle. \tag{1.7}$$

By construction, B_∞ is the direct limit when n grows to ∞ of the groups B_n generated by $n-1$ generators $\sigma_1, \dots, \sigma_{n-1}$ submitted to the relations of (1.7)*, when B_n is embedded in B_{n+1} by adding σ_n. It is well known that the group B_n can be realized as the group of isotopy classes of n strand braid diagrams, and as the mapping class group of an n-punctured disk—see for instance* [3] *or* [24]*. It directly follows from the presentation* (1.7) *that the map* $\mathsf{sh} : \sigma_i \mapsto \sigma_{i+1}$ *extends into a non-surjective endomorphism of B_∞ ("*shift *endomorphism").*

Define on B_∞ a binary operation $\lhd$ by

$$a \lhd b := \mathsf{sh}(b)^{-1}\, \sigma_1\, \mathsf{sh}(a)\, b, \tag{1.8}$$

a shifted conjugation with the factor σ_1 added. Then one checks the equalities

$$(a \lhd b) \lhd c = \mathsf{sh}(c)^{-1}\, \mathsf{sh}^2(b)^{-1}\, \sigma_1\sigma_2\, \mathsf{sh}^2(a)\, \mathsf{sh}(b)\, c = (a \lhd c) \lhd (b \lhd c), \tag{1.9}$$

so $(B_\infty, \lhd)$ is a shelf. It is easy to check that this shelf is neither a spindle, nor a rack: for instance, we have $1 \lhd 1 = \sigma_1 \neq 1$, and $a \lhd 1 = 1$ is impossible, as we have $a \lhd 1 = \sigma_1\, \mathsf{sh}(a)$, and σ_1 does not belong to the image of sh*. For more information about the braid shelf, we refer to Section 2.4 below and to* [25].

We can try to further generalize Example 1.15. Let G be a group, let s a fixed element of G, and let ϕ is an endomorphism of G. Copying (1.8), define

$$a \lhd b := \phi(b)^{-1}\, s\, \phi(a)\, b. \tag{1.10}$$

Consider $(G, \lhd)$. This is a shelf if, and only if, the element s commutes with every element in the image of ϕ^2 and satisfies the relation $s\phi(s)s = \phi(s)s\phi(s)$, which means that the subgroup of G generated by s is a homomorphic image of the braid group B_∞. Thus, essentially, the above construction works only for the braid group and its quotients.

A typical example appears when (1.8) is applied in the quotient $\mathfrak{S}_\infty$ of B_∞ obtained by adding the involutivity relations $\sigma_1^2 = 1$: then $\mathfrak{S}_\infty$ is the group of all permutations of $\mathbb{Z}_{>0}$ that move finitely many points only. *Mutatis mutandis* (left-shelves are considered), it is proved in [19, Cor. I.4.18] that mapping f to the composed map $\mathsf{sh} \circ f$ defines an (injective) homomorphism from $(\mathfrak{S}_\infty, \lhd)$ into the shelf $\mathsf{Conj}(\mathfrak{I}_{\mathbb{Z}_{>0}})$.

Extensions and variations of the braid shelf appear in [15, 18, 23].

Example 1.16 (**iterations of an elementary embedding**). *Large cardinal axioms play a central rôle in modern set theory. A number of such axioms involve elementary embeddings, which are mappings from a set to itself that preserve all notions that are definable in first order logic from the membership relation $\in$. For every ordinal α, one denotes by V_α the set obtained from $\varnothing$ by applying α times the powerset operation. Let $\mathcal{E}_\lambda$ denote the family of all elementary embeddings from V_λ to itself. One says that an ordinal λ is a* Laver cardinal *if $\mathcal{E}_\lambda$ contains at least one element that is not the identity. The point is as follows. The specific properties of the sets V_α imply that, if i and j belong to $\mathcal{E}_\lambda$, then one can* apply *i to j and obtain*

a new element $i[j]$ of $\mathcal{E}_\lambda$. Moreover, for i, j, k, ℓ in $\mathcal{E}_\lambda$, if $\ell = j[k]$ holds, then so does $i[\ell] = i[j][i[k]]$, that is

$$i[j[k]] = i[j][i[k]]. \tag{1.11}$$

The reason for that is that i preserves every definable notion, in particular the operation of applying a function to an argument. Then (1.11) *means that the binary operation $(i, j) \mapsto i[j]$ on $\mathcal{E}_\lambda$ obeys the selfdistributivity law* LD*. When j is the identity mapping, the closure of $\{j\}$ under the "application" operation is trivial. But assume that λ is a Laver cardinal, and j is an element of $\mathcal{E}_\lambda$ that is not the identity. Then the closure of $\{j\}$ under the "application" operation is a nontrivial left-shelf $\mathsf{Iter}(j)$. This left-shelf has fascinating properties, yet its structure is still far from being understood completely* [47, 28, 29, 37, 22, 27].

Example 1.17 (Laver tables). *For every positive integer N, there exists a unique binary operation $\triangleright$ on $\{1, ..., N\}$ that obeys the law*

$$x \triangleright (y \triangleright 1) = (x \triangleright y) \triangleright (x \triangleright 1),$$

*and the structure so obtained is a left-shelf if, and only if, N is a power of 2. The structure with 2^n elements is called the n*th Laver table*, usually denoted by A_n. Laver tables appear as the elementary building bricks for constructing all (finite) monogenerated shelves* [30, 31, 57]*, and can adequately be seen as counterparts of cyclic groups in the SD-world. We refer to the survey by A. Drápal in this volume for a more complete introduction* [32]*. Let us simply mention here that Laver tables are natural quotients of the left-shelf $\mathsf{Iter}(j)$ of Example 1.16* [49]*, and that some of their combinatorial properties are so far established only assuming the existence of a Laver cardinal, which is an unprovable axiom, see for instance* [22].

1.4. **An overview of the SD-world.** We present in Fig. 2 a sort of chart of the SD-world as we can conceive it now. Here we leave the case of spindles aside, and, therefore, we have three classes included one in the other, namely quandles, racks, and shelves. The lanscape is rather different according to whether we consider quandles and racks, or general shelves, and, on the other hand, whether we consider monogenerated structures, or structures with more than one generator.

2. Word problem, the case of shelves I

For every algebraic law or family of algebraic laws, a basic question is the associated word problem, namely the question of deciding whether or not two terms in the absolutely free algebra for the relevant operations are equivalent with respect to the congruence generated by the law(s). In other words, whether or not they represent the same element in the corresponding free structure. Thus four word problems respectively involving shelves, spindles, racks, and quandles occur. In this section and the next one, we begin with the case of shelves, that is, of selfdistributivity alone: in this case, several solutions are known, none of which is trivial, and we explain them. Different solutions can be considered: *syntactic* solutions aim at manipulating terms and deciding their possible equivalence directly, whereas *semantic* solutions consist in evaluating terms in some particular concrete structure(s): when the latter is free, or includes a free structure, two terms are equivalent if, and only if, their evaluations coincide. For instance, in the case of associativity, a syntactic solution may consist in removing parentheses and checking whether the remaining

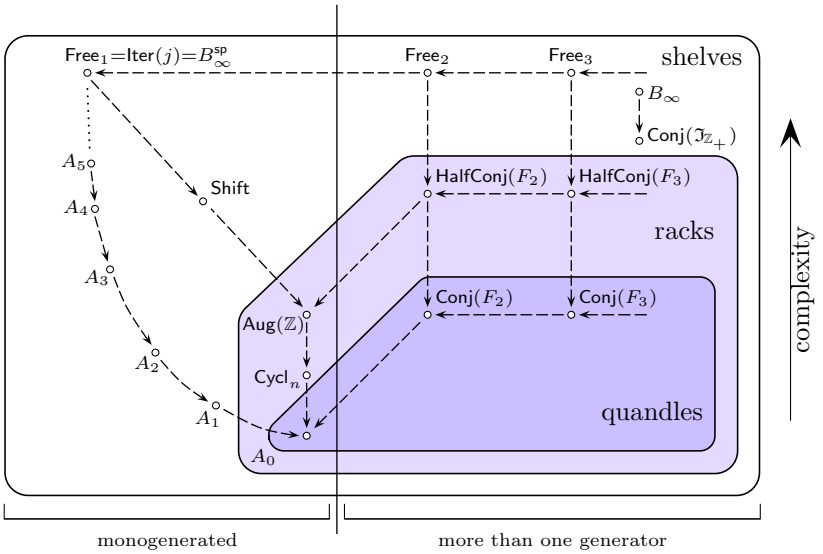


FIGURE 2. A chart of the SD-world, with dashed arrows for quotients. As every shelf with n generators is a quotient of the free shelf Free$_n$ on n generators, it is natural to put Free$_n$ on the top of the diagram; as will be seen in Section 3.3, Free$_n$ is not really more complicated than Free$_1$ for $n \geqslant 2$, so we put them on the same complexity line. We represent similarly free racks, that is, half-conjugacy racks associated with free groups, and free quandles, that is, conjugacy quandles associated with free groups (Prop. 4.2). Here, the complexity drops for one generator. On the monogenerated side, quandles and racks are almost trivial, whereas shelves are many, with a spine made by Laver tables, which form an inverse system with limit Free$_1$ (whenever a Laver cardinal exists); by contrast, on the multigenerated case, lots of racks and quandles have been investigated, whereas not so many really new examples of shelves are known.

words coincide, whereas a semantic solution may consist in evaluating terms in the free monoids X^*—essentially the same thing in this trivial case.

This first part about word problems is divided into three sections. Section 2.1 is preparatory and explains "comparison property", an important feature of selfdistributivity. In Section 2.2, we derive a conditional solution for the word problem in the case of one variable, namely one that is valid provided there exists a shelf with a certain "acyclicity" property. Then, in Section 2.3, we describe two examples of such shelves and thus solve the word problem. Finally, we describe in Section 2.4 a semantic solution based on the braid shelf of Example 1.15 that is more efficient than the syntactic solution so far considered.

2.1. **The comparison property.** To make our survey compatible with the existing literature [19, 47], we switch hereafter to the left version of selfdistributivity LD and, accordingly, use $\triangleright$ instead of $\triangleleft$.

We denote by $\mathsf{Term}_{\triangleright}(X)$ the family of all well formed terms constructed from a set X (usually $\{x_1, x_2, ...\}$, with elements called *variables*) using the operation $\triangleright$, *i.e.*, the absolutely free $\triangleright$-algebra based on X. We denote by $=_{\mathsf{LD}}$ the congruence on $\mathsf{Term}_{\triangleright}(X)$ generated by the instances of the law LD, *i.e.*, the least equivalence

relation that is compatible with the operations and contains the said instances. By construction, the quotient-structure $\mathsf{Term}_{\triangleright}(X)/=_{\mathrm{LD}}$ is a left-shelf generated by X, and it is universal for all such left-shelves, so it is a free left-shelf based on X.

We begin with a preparatory result about selfdistributivity in the case of terms in one variable. To state it, we start with the natural notion of a *divisor*.

Definition 2.1 (**division relation**). *For $\triangleright$ a binary operation on S and a, b in S, we say that a* divides *b, written $a \sqsubset b$, if $a \triangleright x = b$ holds for some x. We write $\sqsubset^*$ for the transitive closure of $\sqsubset$.*

If $\triangleright$ is associative, there is no need to distinguish between $\sqsubset$ and $\sqsubset^*$, since we then have $(a \triangleright x_1) \triangleright x_2 = a \triangleright (x_1 \triangleright x_2)$, but, in general, $\sqsubset$ need not be transitive.

The following result about selfdistributivity is then fundamental:

Lemma 2.2 (**comparison property**). *If S is a monogenerated left-shelf and a, b belong to S, then at least one of $a \sqsubset^* b$, $a = b$, $b \sqsubset^* a$ holds.*

If ϕ is a morphism, $a \sqsubset^* b$ implies $\phi(a) \sqsubset^* \phi(b)$, so the point is to prove Lemma 2.2 when S is a free left-shelf generated by a single element, say x. By definition, the latter consists of the $=_{\mathrm{LD}}$-classes of terms in $\mathsf{Term}_{\triangleright}(x)$—we write $\mathsf{Term}_{\triangleright}(x)$ for $\mathsf{Term}_{\triangleright}(\{x\})$.

Notation 2.3 (**relation $\sqsubset_{\mathrm{LD}}$**). *For T, T' in $\mathsf{Term}_{\triangleright}(x)$, we write $T \sqsubset_{\mathrm{LD}} T'$ if there exists T_1 satisfying $T \triangleright T_1 =_{\mathrm{LD}} T'$, and $\sqsubset^*_{\mathrm{LD}}$ for the transitive closure of $\sqsubset_{\mathrm{LD}}$.*

Then, an equivalent form of Lemma 2.2 is

Lemma 2.4 (**comparison property**, reformulated). *For all T, T' in $\mathsf{Term}_{\triangleright}(x)$, at least one of $T \sqsubset^*_{\mathrm{LD}} T'$, $T =_{\mathrm{LD}} T'$, or $T' \sqsubset^*_{\mathrm{LD}} T$ holds.*

Elements of $\mathsf{Term}_{\triangleright}(X)$ can be seen as binary trees with internal nodes labeled $\triangleright$ and leaves labeled with elements of X (thus variables). What Lemma 2.4 says is that, if T and T' are any two terms in one variable, then, up to LD-equivalence, one is always an iterated left subterm of the other. When associativity is considered, the result is trivial, since a term in one variable is just a power. In the case of selfdistributivity, the result, which is difficult, was proved in [11], and independently reproved shortly after by R. Laver in [47] using a disjoint argument. Here we sketch the two steps of the former proof, which is simpler.

First, by definition, the relation $=_{\mathrm{LD}}$ is the congruence on $\mathsf{Term}_{\triangleright}(X)$ generated by all pairs of terms of the form

$$(\ T_1 \triangleright (T_2 \triangleright T_3)\ ,\ (T_1 \triangleright T_2) \triangleright (T_1 \triangleright T_3)\ ). \tag{2.1}$$

We consider an oriented, non-symmetric version of the latter relation.

Definition 2.5 (**LD-expansion**). *We let $\to_{\mathrm{LD}}$ be the smallest reflexive and transitive relation on $\mathsf{Term}_{\triangleright}(X)$ that is compatible with multiplication and contains all pairs* (2.1). *When $T \to_{\mathrm{LD}} T'$ holds, we say that T' is an* LD-expansion *of T.*

Thus, T' is an LD-expansion of T if T' can be obtained from T by applying the LD law, but always in the expanding direction. Clearly, $T \to_{\mathrm{LD}} T'$ implies $T =_{\mathrm{LD}} T'$, but the converse implication fails, as $\to_{\mathrm{LD}}$ is not symmetric: $x \triangleright (x \triangleright x) \to_{\mathrm{LD}} (x \triangleright x) \triangleright (x \triangleright x)$ holds, but $(x \triangleright x) \triangleright (x \triangleright x) \to_{\mathrm{LD}} x \triangleright (x \triangleright x)$ does not.

Lemma 2.6 (**confluence property**). *Two LD-equivalent terms admit a common LD-expansion.*

Idea of the proof. By definition, two terms T, T' are LD-equivalent if, and only if, there exists a finite zigzag of LD-expansions and inverses of LD-expansions connecting T to T'. The point is to show that there always exists a zigzag with only one expansion and one inverse of expansion. To this end, it suffices to prove that the relation $\to_{\text{LD}}$ is confluent, *i.e.*, that any two LD-expansions T', T'' of a term T admit a common LD-expansion. It is easy to check local confluence, namely confluence when T' and T'' are obtained from T by applying the LD law at most once ("atomic" LD-expansion). But a termination problem arises, because the relation $\to_{\text{LD}}$ is far from being noetherian (infinite sequences of LD-expansions exist) and Newman's standard diamond lemma [54] cannot be applied. To solve this, one considers, for every term T, the term ∂T inductively defined by

$$\partial T := \begin{cases} x & \text{for } T = x, \\ \partial T_0 \otimes \partial T_1 & \text{for } T = T_0 \triangleright T_1, \end{cases} \tag{2.2}$$

where $\otimes$ itself is inductively defined by

$$S \otimes T := \begin{cases} S \triangleright T & \text{for } T = x, \\ (S \otimes T_0) \triangleright (S \otimes T_1) & \text{for } T = T_0 \triangleright T_1. \end{cases} \tag{2.3}$$

One can show that ∂T is an LD-expansion of T and of all atomic LD-expansions of T, and that $T \to_{\text{LD}} T'$ implies $\partial T \to_{\text{LD}} \partial T'$. It is then easy to deduce that, for every p, the term $\partial^p T$ is an LD-expansion of all LD-expansions of T obtained using at most p atomic expansion steps. From there, any two LD-expansions of T admit as a common LD-expansion any term $\partial^p T$ with p large enough. □

The second ingredient is the following specific property.

Notation 2.7 (**powers**). *For $\triangleright$ a binary operation on S and a in S, the* left *and* right *powers of a are defined by $a_{[1]} := a^{[1]} := a$, and $a_{[n+1]} := a_{[n]} \triangleright a$ and $a^{[n+1]} := a \triangleright a^{[n]}$.*

Lemma 2.8 (**absorption property**). *If S is a left-shelf generated by an element g, then, for every a in S, we have $a \triangleright g^{[n]} = g^{[n+1]}$ for n large enough (depending on a).*

Proof. As in the case of the comparison property, it is sufficient to consider the case of the free left-shelf, namely to establish that, for every term T in $\mathsf{Term}_{\triangleright}(x)$,

$$T \triangleright x^{[n]} =_{\text{LD}} x^{[n+1]} \tag{2.4}$$

holds for n large enough. We use induction on (the size of) T. For $T = x$, (2.4) holds for every $n \geqslant 1$, with $=_{\text{LD}}$ being an equality. Otherwise, write $T = T_0 \triangleright T_1$, and assume that (the counterpart of) (2.4) holds for T_i for every $n \geqslant n_i$. For $n \geqslant \max(n_0, n_1) + 1$, we obtain

$$\begin{aligned} x^{[n+1]} &=_{\text{LD}} T_0 \triangleright x^{[n]} && \text{owing to } n \geqslant n_0 \text{ and the IH for } T_0, \\ &=_{\text{LD}} T_0 \triangleright (T_1 \triangleright x^{[n-1]}) && \text{owing to } n-1 \geqslant n_1 \text{ and the IH for } T_1, \\ &=_{\text{LD}} (T_0 \triangleright T_1) \triangleright (T_0 \triangleright x^{[n-1]}) && \text{by LD}, \\ &=_{\text{LD}} (T_0 \triangleright T_1) \triangleright x^{[n]} = T \triangleright x^{[n]} && \text{owing to } n-1 \geqslant n_0 \text{ and the IH for } T_0. \ \square \end{aligned}$$

Using Lemmas 2.6 and 2.8, it is now easy to deduce the comparison property.

Proof of Lemma 2.4. (See Fig. 3.) Let T, T' belong to $\mathsf{Term}_{\triangleright}(x)$. By Lemma 2.8, for n large enough, we have

$$T \triangleright x^{[n]} =_{\mathsf{LD}} x^{[n+1]} =_{\mathsf{LD}} T' \triangleright x^{[n]}. \tag{2.5}$$

By Lemma 2.6, we deduce that $T \triangleright x^{[n]}$ and $T' \triangleright x^{[n]}$ admit a common LD-expansion T'' (which can be assumed to be $\partial^p x^{[n+1]}$ for some p). Using an induction on the number of LD-expansion steps, it is easy to verify that, if T is not a variable and T' is an LD-expansion of T, then there exists r such that the rth iterated left subterm $\mathsf{left}^r(T')$ of T' is an LD-expansion of the left subterm $\mathsf{left}(T)$ of T. In the current case, the left subterm of $T \triangleright x^{[n]}$ is T, and we deduce that there exist r satisfying $T \to_{\mathsf{LD}} \mathsf{left}^r(T'')$, whence $T =_{\mathsf{LD}} \mathsf{left}^r(T'')$. Similarly, there exist r' satisfying $T' \to_{\mathsf{LD}} \mathsf{left}^{r'}(T'')$, whence $T' =_{\mathsf{LD}} \mathsf{left}^{r'}(T'')$.

Now three cases may occur. For $r = r'$, we find $T =_{\mathsf{LD}} \mathsf{left}^r(T) =_{\mathsf{LD}} T'$, whence $T =_{\mathsf{LD}} T'$. For $r > r'$, the term $\mathsf{left}^r(T)$ is an iterated left subterm of $\mathsf{left}^{r'}(T)$, that is, we have $\mathsf{left}^r(T) \sqsubset^* \mathsf{left}^{r'}(T)$, hence $T \sqsubset^*_{\mathsf{LD}} T'$. Similarly, for $r < r'$, we obtain $T \sqsupset^*_{\mathsf{LD}} T'$ by a symmetric argument. □

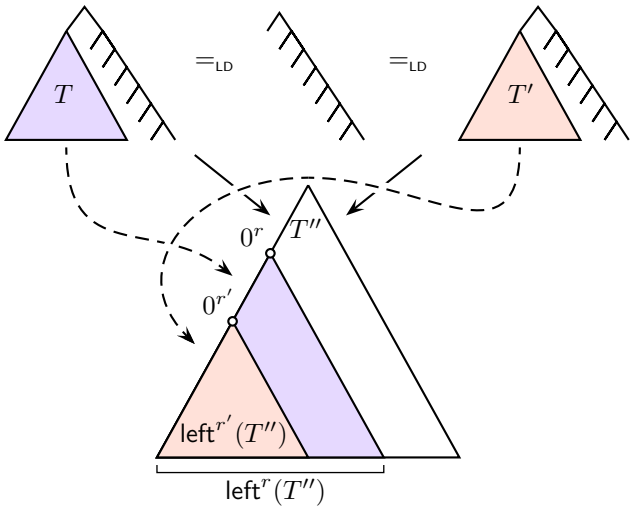


FIGURE 3. Proof of the comparison property: the terms T and T' admit LD-expansions that both are iterated left subterms of some (large) term T'': the latter coincide, or one is an iterated left subterm of the other.

2.2. **A conditional syntactic solution.** We are now ready to describe a syntactic solution of the word problem for LD. However, in a first step, the solution will remain conditional, as its correctness relies on an extra assumption that will be established in the next section only.

Definition 2.9 (**acyclic**). *A left-shelf S is called* acyclic *if the relation $\sqsubset$ on S has no cycle.*

Lemma 2.10 (**exclusion property**). *If there exists an acyclic left-shelf, the relations $=_{\mathsf{LD}}$ and $\sqsubset^*_{\mathsf{LD}}$ on $\mathsf{Term}_{\triangleright}(x)$ exclude one another.*

Proof. Assume that S is an acyclic left-shelf, and T, T' are terms in $\mathsf{Term}_{\triangleright}(x)$ satisfying $T \sqsubset^*_{\mathsf{LD}} T'$. Let g be an element of S, and let $T(g)$ and $T'(g)$ be the evaluations of T and T' in S when x is given the value g. By definition, $T \sqsubset^*_{\mathsf{LD}} T'$ implies $T(g) \sqsubset^* T'(g)$ in S, whence $T(g) \neq T'(g)$, since $\sqsubset$ has no cycle in S. As $T =_{\mathsf{LD}} T'$ would imply $T(g) = T'(g)$, we deduce that $T =_{\mathsf{LD}} T'$ is impossible. □

Proposition 2.11 (**conditional word problem**). [11, 47] *If there exists an acyclic left-shelf, the word problem of* LD *is decidable in the case of one variable.*

Proof. The relation $=_{\mathsf{LD}}$ on $\mathsf{Term}_{\triangleright}(x)$ is semi-decidable, meaning that there exists an algorithm that, starting with any two terms T, T', returns `true` if $T =_{\mathsf{LD}} T'$ holds, and runs forever if $T =_{\mathsf{LD}} T'$ fails: just start with T, "stupidly" enumerate all terms T'' that are LD-equivalent to T' by repeatedly applying LD in either direction at any position, and test $T =? T''$. Similarly, $\sqsubset^*_{\mathsf{LD}}$ is semi-decidable: starting with T, T', enumerate all terms T'' that are LD-equivalent to T' and test $T \sqsubset^* ? T''$, *i.e.*, test whether T is a proper iterated left subterm of T''. Thus, both $=_{\mathsf{LD}}$ and $\sqsubset^*_{\mathsf{LD}} \cup \sqsupset^*_{\mathsf{LD}}$ are semi-decidable relations on $\mathsf{Term}_{\triangleright}(x)$. By the comparison property (Lemma 2.4), their union is all of $\mathsf{Term}_{\triangleright}(x)$ and, if there exists an acyclic left-shelf, they are disjoint by the exclusion property (Lemma 2.10). Thus, in this case, $\sqsubset^*_{\mathsf{LD}} \cup \sqsupset^*_{\mathsf{LD}}$ coincides with $\neq_{\mathsf{LD}}$. Then $=_{\mathsf{LD}}$ and its complement are semi-decidable, hence they are decidable. □

The approach leads to a (certainly very inefficient) algorithm. Fix an exhaustive enumeration $(s_n, s'_n)_{n \geqslant 0}$ of the pairs of finite sequences of positions in a binary tree—that is, sequences of binary addresses, see Section 3.1 below—and let LD-EXPAND(T, s) be the result of applying LD in the expanding direction starting from T and according to the sequence of positions s.

Algorithm 2.12 (**syntactic solution, case of one variable**).
Input: Two terms T, T' in $\mathsf{Term}_{\triangleright}(x)$
Output: `true` if T and T' are LD-equivalent, `false` otherwise

```
FOUND := false
n := 0
while not FOUND do
   T_1 := LD-EXPAND(T, s_n)
   T'_1 := LD-EXPAND(T', s'_n)
   if T_1 = T'_1 then
      return true
      FOUND := true
   if T_1 ⊏* T'_1 or T ⊐* T'_1 then
      return false
      FOUND := true
   n := n + 1
```

Termination of the algorithm follows from the comparison property: for all T, T', there must exist a pair of sequences (s, s') such that expanding T according to s and T' according to s' yields a pair of terms (T_1, T'_1) consisting of equal or comparable terms; correctness follows from the exclusion property, which guarantees that $T_1 \sqsubset^* T'_1$ implies $T_1 \neq_{\mathsf{LD}} T'_1$, hence $T \neq_{\mathsf{LD}} T'$. So, at this point, we may state:

Proposition 2.13 (**conditional syntactic solution, case of one variable**). *If there exists an acyclic left-shelf, Algorithm 2.12 is correct, and the word problem of selfdistributivity is decidable in the case of one variable.*

Example 2.14 (**syntactic solution**). *Consider* $T := (x \triangleright x) \triangleright (x \triangleright (x \triangleright x))$ *and* $T' := x \triangleright ((x \triangleright x) \triangleright (x \triangleright x))$. *Expanding* T *at* $(1, \emptyset)$ *and* T' *at* $(\emptyset, 1, 0)$ *(see Definition 3.2 for the formalism), we obtain the common LD-expansion*

$$((x \triangleright x) \triangleright (x \triangleright x)) \triangleright ((x \triangleright x) \triangleright (x \triangleright x)),$$

and we conclude that T *and* T' *are* LD*-equivalent. Note that, here, the conclusion is certain without any hypothesis (no need of an acyclic left-shelf), contrary to the case of expansions that are proper iterated left subterms of one another.*

2.3. **A syntactic solution.** When the above method was first described (1989), no example of an acyclic left-shelf was known and its existence was a conjecture. Shortly after, R. Laver established in [47]:

Proposition 2.15 (**acyclic I**). *If* j *is a nontrivial elementary embedding of* V_λ *into itself, the left-shelf* $\mathsf{Iter}(j)$ *is acyclic.*

This resulted in the paradoxical situation of a finitistic problem (the word problem of LD) whose only known solution appeals to an unprovable axiom:

Corollary 2.16. *If there exists a Laver cardinal, Algorithm 2.12 is correct, and the word problem of selfdistributivity is decidable in the case of one variable.*

The previous puzzling situation was resolved by the construction, without any set theoretical assumption, of another acyclic left-shelf [10, 12]:

Proposition 2.17 (**acyclic II**). *The braid shelf of Example 1.15 is acyclic.*

Idea of the proof. As we are considering left selfdistributivity here, the relevant version is the operation $\triangleright$ defined on the braid group B_∞ by

$$a \triangleright b := a\, \mathsf{sh}(b)\, \sigma_1\, \mathsf{sh}(a)^{-1}. \tag{2.6}$$

Proving that $(B_\infty, \triangleright)$ is acyclic means that no equality of the form

$$a = (\cdots ((a \triangleright b_1) \triangleright b_2) \cdots) \triangleright b_n \tag{2.7}$$

with $n \geqslant 1$ is possible in B_∞. According to the definition of the operation $\triangleright$, the right term in (2.7) expands into an expression of the form

$$a \cdot \mathsf{sh}(c_0)\sigma_1\mathsf{sh}(c_1)\sigma_1 \cdots \sigma_1\mathsf{sh}(c_n), \tag{2.8}$$

and, therefore, for excluding (2.7), it suffices to prove that a braid of the form

$$\mathsf{sh}(c_0)\sigma_1\mathsf{sh}(c_1)\sigma_1 \cdots \sigma_1\mathsf{sh}(c_n) \tag{2.9}$$

is never trivial (equal to 1). It is natural to call the braids as in (2.9) σ_1*-positive*, since they admit a decomposition, in which there is at least one letter σ_1 and no letter σ_1^{-1}. So, the problem is to show that a σ_1-positive braid is never trivial.

Several arguments exist, see in particular [12], the simplest being the one, due to D. Larue [43], which appeals to the Artin representation of B_∞ in $\mathsf{Aut}(F_\infty)$, where F_∞ denotes a free group based on an infinite family $\{x_i \mid i \geqslant 1\}$, identified with the family of all freely-reduced words on $\{x_i^{\pm 1} \mid i \geqslant 1\}$. Artin's representation is defined by the rules

$$\rho(\sigma_i)(x_i) := x_i x_{i+1} x_i^{-1}, \quad \rho(\sigma_i)(x_{i+1}) := x_i, \quad \rho(\sigma_i)(x_k) := x_k \text{ for } k \neq i, i+1,$$

and simple arguments about free reduction show that, if c is a σ_1-positive braid, then $\rho(c)$ maps x_1 to a reduced word that finishes with the letter x_1^{-1} and, therefore, c cannot be trivial, since $\rho(1)$ maps x_1 to x_1, which does not finish with x_1^{-1}. □

Applying Prop. 2.11, we remove the exotic assumption in Corollary 2.16:

Corollary 2.18. *Algorithm 2.12 is correct, and the word problem of selfdistributivity is decidable in the case of one variable.*

2.4. **A semantic solution.** However, we can obtain more, namely a new, more efficient algorithm for the word problem of LD. The starting point is the following criterion, whose proof is essentially the same as the one of Prop. 2.11:

Lemma 2.19 (**freeness criterion**). *If S is an acyclic monogenerated left-shelf, then S is free.*

Proof. Assume that S is generated by g. Let T, T' be two terms in $\mathsf{Term}_{\triangleright}(x)$. As above, we write $T(g)$ for the evaluation of T at $x := g$. As S is a left-shelf, $T =_{\mathsf{LD}} T'$ certainly implies $T(g) = T'(g)$. Conversely, assume $T \neq_{\mathsf{LD}} T'$. By Lemma 2.4, at least one of $T \sqsubset^*_{\mathsf{LD}} T'$, $T \sqsupset^*_{\mathsf{LD}} T'$ holds, say for instance $T \sqsubset^*_{\mathsf{LD}} T'$. By projection, we deduce $T(g) \sqsubset^* T'(g)$ in S, whence $T(g) \neq T'(g)$, since $\sqsubset$ has no cycle in S. Therefore, $T =_{\mathsf{LD}} T'$ is equivalent to $T(g) = T'(g)$, and S is free. □

We deduce

Proposition 2.20 (**realization**). *For every braid a, the sub-left-shelf of $(B_\infty, \triangleright)$ generated by a is free.*

This applies in particular for $a = 1$; the braids obtained from 1 by iterating $\triangleright$ are called *special* in [19], so special braids provide a realization B_∞^{sp} of the rank 1 free left-shelf Free_1. Efficient solutions of the word problem for the presentation (1.7) of B_∞ are known [34, 16], *i.e.*, algorithms that decide whether or not a word in the letters $\sigma_i^{\pm 1}$ represents 1 in B_∞. We deduce a simple semantic algorithm for the word problem of LD. Below, we use BW_∞ for the family of all braid words, $\textsc{equiv}_{B_\infty}$ for a solution of the word problem of (1.7), $\frown$ for word concatenation, SHIFT for braid word shifting, and INV for braid word formal inversion.

Algorithm 2.21 (**semantic solution, case of one variable**).
Input: Two terms T, T' in $\mathsf{Term}_{\triangleright}(x)$
Output: `true` if T and T' are LD-equivalent, `false` otherwise

```
1: w := EVAL(T)
2: w' := EVAL(T')
3: return EQUIV_{B∞}(w, w')

7: function EVAL(T : term): word in BW∞
8: if T = x ∈ X then
9:    return ε
10: else if T = T0 ▷ T1 then
11:    return EVAL(T)⌢SHIFT(EVAL(T'))⌢σ1⌢INV(SHIFT(EVAL(T)))
```

The inductive definition of $\triangleright$ implies that the braid word associated with a term T of size n has length at most $2^{O(n)}$. On the other hand, in connection with the existence of an automatic structure on braid groups, there exist solutions of the word problem for (1.7) of quadratic complexity [34], so the overall complexity of

Algorithm 2.21 is simply exponential—whereas the only proved upper bound for the complexity of Algorithm 2.12 is a tower of exponentials of exponential height.

Example 2.22 (semantic solution). *Consider* $T := (x \triangleright x) \triangleright (x \triangleright (x \triangleright x))$ *and* $T' := x \triangleright ((x \triangleright x) \triangleright (x \triangleright x))$. *The reader is invited to check*

$$\text{EVAL}(T) = \sigma_1\sigma_3\sigma_2\sigma_1\sigma_2^{-1}, \qquad \text{EVAL}(T') = \sigma_2\sigma_3\sigma_2\sigma_3^{-1}\sigma_1.$$

As the latter braid words are equivalent (they both represent $\sigma_3\sigma_2\sigma_1$ *in* B_∞*), we conclude that* T *and* T' *are* LD*-equivalent.*

3. Word problem, the case of shelves II

We thus obtained in Section 2 a positive solution for the word problem of selfdistributivity in the case of terms in one variable. At this point, several questions remain open: Where does the exotic braid operation of (1.8) or (2.6) come from? What about the case of terms involving more than one variable? Can one find solutions based on normal terms, that is, describe a distinguished representative in every LD-class? Are there efficient syntactic solutions (the one of Section 2 is not)? These four questions are addressed in the four subsections below.

3.1. **Where does the braid shelf come from?** The answer lies in the approach developed in [12], which is parallel to the treatment of associativity and commutativity by R. Thompson in the 1970s [52, 59, 4]. In addition to the braid application, one also obtains a direct proof that the free left-shelf Free_1 is acyclic, without refering to any concrete realization like the one based on braids or the one based on iterations of elementary embeddings.

The idea is to see the relations $=_{\text{LD}}$ and $\to_{\text{LD}}$ as the result of applying the action of a monoid on terms. To this end, we take into account the positions and the orientations, where the selfdistributivity law is applied, as already alluded to in the definition of the procedure LD-expand of Algorithm 2.12. Every term T in $\mathsf{Term}_\triangleright(X)$ that is not a variable (*i.e.*, an element of X) admits a left and a right subterm. Iterating, we can specify each subterm of T by a finite sequence of 0s (for left) and 1s (for right): such finite sequences will be called *addresses*, denoted α, β,..., and we use $T_{/\alpha}$ for the α*-subterm* of T, that is, the subterm of T that corresponds to the fragment below the address α in the tree associated with T. With this notation, $T_{/0}$ (*resp.* $T_{/1}$) is the left (*resp.* right) subterm of T. Note that $T_{/\alpha}$ exists only for α short enough (the family of all αs for which $T_{/\alpha}$ exists will be called the *skeleton* of T).

Definition 3.1 (operator LD_α**, monoid** $\mathsf{Geom}_{\text{LD}}$**).** *For each address* α*, we denote by* LD_α *the partial operator on* $\mathsf{Term}_\triangleright(X)$ *such that* $T \bullet \text{LD}_\alpha$ *is defined if* $T_{/\alpha}$ *exists and can be written as* $T_1 \triangleright (T_2 \triangleright T_3)$*, in which case* $T \bullet \text{LD}_\alpha$ *is the term obtained by replacing the latter subterm with* $(T_1 \triangleright T_2) \triangleright (T_1 \triangleright T_3)$*. The* geometry monoid *of* LD *is the monoid* $\mathsf{Geom}_{\text{LD}}$ *generated by all operators* LD_α *and their inverses under composition*

Thus applying LD_α means applying the LD law at position α in the expanding direction. By definition, two terms T, T' are LD-equivalent if, and only if, some element of the monoid $\mathsf{Geom}_{\text{LD}}$ maps T to T'. When the selfdistributivity law LD is replaced with the associativity law A, the corresponding geometry monoid Geom_{A} turns out to (essentially) Richard Thompson's group F [59]. It is important to note

that the action of $\mathsf{Geom}_{\mathrm{LD}}$ is partial: for instance, $T \bullet \mathrm{LD}_\alpha$ is defined only when α is short enough (precisely: when $\alpha 0$, $\alpha 10$, and $\alpha 11$ belong to the skeleton of T).

For the sequel, it is crucial to work in a group context. However, $\mathsf{Geom}_{\mathrm{LD}}$ is not a group, but only an inverse monoid: exchanging LD_α and LD_α^{-1} and reversing the order of factors provides for every g in $\mathsf{Geom}_{\mathrm{LD}}$ an element g^{-1} satisfying $gg^{-1}g = g$ and $g^{-1}gg^{-1} = g^{-1}$, but gg^{-1} is only the identity of its domain. Contrary to the case of associativity, no quotient-group of $\mathsf{Geom}_{\mathrm{LD}}$ is useful. However, one can *guess* a list of relations $\mathsf{Rel}_{\mathrm{LD}}$ that connect the maps LD_α and consider the abstract group $\widetilde{\mathsf{Geom}}_{\mathrm{LD}}$ presented by $\mathsf{Rel}_{\mathrm{LD}}$: if $\mathsf{Rel}_{\mathrm{LD}}$ is exhaustive enough, we can hope to work with $\widetilde{\mathsf{Geom}}_{\mathrm{LD}}$ as we did with $\mathsf{Geom}_{\mathrm{LD}}$.

Here is the key point. The absorption property (Lemma 2.8) implies that, for every T in $\mathsf{Term}_\triangleright(x)$, the terms $x^{[n+1]}$ and $T \triangleright x^{[n]}$ are LD-equivalent for n large enough. Hence, there exists an element in $\mathsf{Geom}_{\mathrm{LD}}$ that maps $x^{[n+1]}$ to $T \triangleright x^{[n]}$. Reading step by step the inductive proof of Lemma 2.8 shows that such an element can be defined inductively as follows (note that n does not occur):

Definition 3.2 (**blueprint**). *For T in $\mathsf{Term}_\triangleright(x)$, we inductively define $\chi(T)$ in $\mathsf{Geom}_{\mathrm{LD}}$ by*

$$\chi(T) := \begin{cases} 1 & \text{for } T = x, \\ \chi(T_0) \cdot \mathsf{sh}_1(\chi(T_1)) \cdot \mathrm{LD}_\emptyset \cdot \mathsf{sh}_1(\chi(T_1))^{-1} & \text{for } T = T_0 \triangleright T_1, \end{cases} \tag{3.1}$$

We denote by $\widetilde{\chi}(T)$ the element of the group $\widetilde{\mathsf{Geom}}_{\mathrm{LD}}$ defined by a similar induction. Then $\widetilde{\chi}(T)$ should be viewed as a sort of copy of T inside $\widetilde{\mathsf{Geom}}_{\mathrm{LD}}$ (the "*blueprint*" of T). We then can obtain a non-conditional proof of the exclusion property (Lemma 2.10):

Lemma 3.3 (**exclusion property**). *The relations $=_{\mathrm{LD}}$ and $\sqsubset_{\mathrm{LD}}^*$ on $\mathsf{Term}_\triangleright(x)$ exclude one another.*

Sketch of the proof. Assume $T =_{\mathrm{LD}} T'$. There exists g in $\mathsf{Geom}_{\mathrm{LD}}$ that maps T to T', and therefore $\mathsf{sh}_0(g)$ maps $T \triangleright x^{[n]}$ to $T' \triangleright x^n$, where $\mathsf{sh}_0(g)$ is the shifted version of g that consists in applying g in the left subterm (that is, replacing LD_α with $\mathrm{LD}_{0\alpha}$ everywhere in g). Then both $\chi(T')$ and $\chi(T)\mathsf{sh}_0(g)$ map $x^{[n+1]}$ to $T' \triangleright x^{[n]}$, so the quotient $\chi(T)^{-1}\chi(T')$ belongs to $\mathsf{sh}_0(\mathsf{Geom}_{\mathrm{LD}})$. Using the explicit presentation of the group $\widetilde{\mathsf{Geom}}_{\mathrm{LD}}$, one checks that, similarly, the quotient $\widetilde{\chi}(T)^{-1}\widetilde{\chi}(T')$ belongs to $\mathsf{sh}_0(\widetilde{\mathsf{Geom}}_{\mathrm{LD}})$.

Assume now $T \sqsubset_{\mathrm{LD}}^* T'$. Then one reduces to the case $T \sqsubset^* T'$, in which case $\widetilde{\chi}(T)^{-1}\widetilde{\chi}(T')$ has an expression in which the generator $\mathrm{LD}_\emptyset$ appears but its inverse does not. An algebraic study of the group $\widetilde{\mathsf{Geom}}_{\mathrm{LD}}$ as group of right fractions then enables one to prove that $\widetilde{\chi}(T)^{-1}\widetilde{\chi}(T')$ does not belong to $\mathsf{sh}_0(\widetilde{\mathsf{Geom}}_{\mathrm{LD}})$—this is the key point. □

As a first consequence, one directly deduces:

Proposition 3.4 (**acyclic III**). *The free left-shelf Free_1 is acyclic.*

In the context of Prop. 2.11, this gives another proof of the validity of Algorithm 2.12, hence of the solvability of the word problem of LD—the first complete one chronologically [10]. We also obtain another solution using the group $\widetilde{\mathsf{Geom}}_{\mathrm{LD}}$. Indeed, Lemma 3.3 shows that $T =_{\mathrm{LD}} T'$ holds if, and only if, $\widetilde{\chi}(T)^{-1}\widetilde{\chi}(T')$ belongs to the subgroup $\mathsf{sh}_0(\widetilde{\mathsf{Geom}}_{\mathrm{LD}})$, which can be tested effectively (we skip the description).

Example 3.5 (blueprint solution). *Consider* $T := (x \triangleright x) \triangleright (x \triangleright (x \triangleright x))$ *and* $T' := x \triangleright ((x \triangleright x) \triangleright (x \triangleright x))$ *again. Then one finds (we write* α *for* LD_α*):*

$$\widetilde{\chi}(T) = \mathrm{LD}_\emptyset\,\mathrm{LD}_{11}\,\mathrm{LD}_1\,\mathrm{LD}_\emptyset\,\mathrm{LD}_1^{-1}, \qquad \widetilde{\chi}(T') = \mathrm{LD}_1\,\mathrm{LD}_{11}\,\mathrm{LD}_1\,\mathrm{LD}_{11}^{-1}\,\mathrm{LD}_1,$$

and we can check $\widetilde{\chi}(T)^{-1}\widetilde{\chi}(T') = \mathrm{LD}_{01}\mathrm{LD}_0\mathrm{LD}_{01}^{-1}\mathrm{LD}_{00}^{-1}\mathrm{LD}_0^{-1}$*: the latter element of the group* $\widetilde{\mathsf{Geom}}_{\mathrm{LD}}$ *belongs to the image of* sh_0*, hence* T' *and* T' *are* LD*-equivalent.*

A second consequence is the construction of the braid operation of (2.6). The explicit form of the relations $\mathsf{Rel}_{\mathrm{LD}}$—which we did not mention so far—implies that Artin's braid group B_∞ is a quotient of the geometry group $\widetilde{\mathsf{Geom}}_{\mathrm{LD}}$, namely the one obtained when all generators LD_α such that α contains at least one 0 are collapsed. Indeed, the remaining relations take the form

$$\mathrm{LD}_{1^i}\,\mathrm{LD}_{1^j}\,\mathrm{LD}_{1^i} = \mathrm{LD}_{1^j}\,\mathrm{LD}_{1^i}\,\mathrm{LD}_{1^i}\,\mathrm{LD}_{1^i0} \qquad \text{for } j = i+1 \geqslant 1,$$
$$\mathrm{LD}_{1^i}\,\mathrm{LD}_{1^j} = \mathrm{LD}_{1^j}\,\mathrm{LD}_{1^i} \qquad \text{for } j \geqslant i+2 \geqslant 2,$$

which project to the relations of (1.7) when LD_{1^i0} is collapsed and LD_{1^i} is mapped to σ_{i+1}. We saw that $T =_{\mathrm{LD}} T'$ holds if, and only if, the quotient $\widetilde{\chi}(T)^{-1}\widetilde{\chi}(T')$ belongs to the subgroup $\mathsf{sh}_0(\widetilde{\mathsf{Geom}}_{\mathrm{LD}})$. Hence, when all generators LD_α with 0 in α are collapsed, $\widetilde{\chi}(T)^{-1}\widetilde{\chi}(T')$ goes to 1, that is, $\widetilde{\chi}(T)$ and $\widetilde{\chi}(T')$ have the same image. In other words, if we consider the operation on the quotient that mimicks the inductive definition of (3.1), then that operation must obey the LD-law. The reader is invited to check that the operation introduced in this way is precisely that of (2.6). Therefore the latter does not appear out of the blue, but it directly stems from (3.1), which itself follows the inductive proof of the absorption property of Lemma 2.8.

We conclude with two remarks. First, the relations of $\mathsf{Rel}_{\mathrm{LD}}$ can be stated so as to involve no LD_α^{-1} and, therefore, one can introduce the monoid $\widetilde{\mathsf{Geom}}_{\mathrm{LD}}^+$ they present. The main open question in this area is

Question 3.6 (embedding conjecture). *Does the monoid* $\widetilde{\mathsf{Geom}}_{\mathrm{LD}}^+$ *embeds in the group* $\widetilde{\mathsf{Geom}}_{\mathrm{LD}}$*?*

A positive answer is conjectured. It amounts to proving that $\widetilde{\mathsf{Geom}}_{\mathrm{LD}}^+$, which is left cancellative, is also right cancellative, and it would imply a number of structural properties for selfdistributivity [19, Chapter IX], in particular that the LD-expansion relation $\to_{\mathrm{LD}}$ admits least upper bounds, thus reminiscent of associahedra and Tamari lattices for associativity [53].

The second remark is that a similar approach can be developed for other algebraic laws, for instance for the "central duplication" law $x(yz) = (xy)(yz)$, where it provides the only known solution of the word problem [21].

3.2. **Normal forms.** Connected with the word problem of selfdistributivity is the question of finding normal forms, namely finding one distinguished term in every LD-equivalence class. Several solutions have been described, in particular by R. Laver in [47] and in subsequent works [48, 50]. Here we sketch the solution developed in [13], which is simpler and directly follows from the properties mentioned above. Once again, we concentrate on the case of one variable.

Let T belong to $\mathsf{Term}_\triangleright(x)$. By the absorption property (Lemma 2.8), the term $T \triangleright x^{[n]}$ is LD-equivalent to $x^{[n+1]}$ for n large enough, hence, by the confluence property (Lemma 2.6), they admit a common LD-expansion, which, by the proof of Lemma 2.6, may be assumed to be of the form $\partial^p x^{[n]}$ for some p. Then, as explained

in the proof of the comparison property (see Fig. 3), there exists a number r such that the iterated left subterm $\mathsf{left}^r(\partial^p x^{[n]})$ is LD-equivalent to T, and, for given p and n, this number r is necessarily unique by the exclusion property. As is usual in this context, the index n does not matter (provided it is large enough) and, so, by choosing p to be minimal, we obtain a distinguished representative:

Proposition 3.7 (normal form). *Call a term T* normal *if T is an iterated left subterm of $\partial^p x^{[n]}$ for some p and n, and p is minimal with that property. Then every term in $\mathsf{Term}_\triangleright(x)$ is* LD*-equivalent to a unique normal term.*

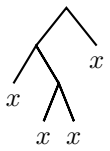
The above construction is effective, but it does not give an explicit description of normal terms. We provide it now. To this end, it is convenient to extend the notion of an iterated left subterm into that of a *cut*. By definition, an iterated left subterm of T corresponds to extracting from T the fragment that lies under some address 0^r, hence on the left of some leaf with address 0^r1^s. We extend the definition to all fragments corresponding to any leaf in (the tree associated with) T.

The easiest way to state a formal definition is to start from the *(right) Polish* expression of T:

Definition 3.8 (Polish expression). *For T in $\mathsf{Term}_\triangleright(X)$, we denote by $[T]$ the word in the letters x and $\bullet$ inductively defined by $[x] := x$ and $[T] := [T_0]^\frown[T_1]^\frown\bullet$ for $T = T_0 \triangleright T_1$ (using $^\frown$ for concatenation).*

The order of symbols in $[T]$ corresponds to the left–right–root enumeration of the associated binary tree, and one obtains a one-to-one correspondence between the nodes in the tree (associated with) T, hence what we called the skeleton of T, and the letters of the word $[T]$. We denote by $\mathsf{add}(p, T)$ the address that corresponds to the pth letter in $[T]$.

For instance, for $T = (x \triangleright (x \triangleright x)) \triangleright x$ (see on the right), one finds $[T] = xxx\bullet\bullet x\bullet$, and the correspondence between letters and addresses is as follows:



p	1	2	3	4	5	6	7
pth letter of $[T]$	x	x	x	$\bullet$	$\bullet$	x	$\bullet$
$\mathsf{add}(p, T)$	00	010	011	01	0	1	$\emptyset$

A word w in the letters $\{x, \bullet\}$ is the Polish expression of a term if, and only if, the number $|w|_X$ of variables in w is one more than the number $|w|_\bullet$ of $\bullet$ and, for every nonempty initial factor w' of w, one has $|w'|_X > |w'|_\bullet$. Iterated left subtrees then correspond to prefixes w' of $[T]$ that themselves are Polish expressions, *i.e.*, satisfy $|w'|_X = |w'|_\bullet + 1$.

For every α that is the address of a leaf in T, hence corresponds to a variable x_i in $[T]$, if w' is the prefix of $[T]$ that finishes with this letter x_i, there exists a unique nonnegative number p such that $w'^\frown\bullet^p$ is a Polish expression: the number p is the local defect in letters $\bullet$, and it is zero if, and only if, w' is the Polish expression of an iterated left subterm of T, hence if, and only if, α is of the form 0^r1^s.

Definition 3.9 (cut). *For every term T and every address α of a leaf of T, we let $\mathsf{cut}(T, \alpha)$ be the term, whose Polish expression is the word $w'^\frown\bullet^p$ as above.*

Example 3.10 (cut). *Let again T be $(x \triangleright (x \triangleright x)) \triangleright x$. The size of T (number of occurrences of variables) is 4, so there are four leaves, at addresses 00, 010, 011, and 1, and the corresponding four cuts are*

$$\mathsf{cut}(T, 00) = x,\ \mathsf{cut}(T, 010) = x \triangleright x,\ \mathsf{cut}(T, 011) = x \triangleright (x \triangleright x),\ \mathsf{cut}(T, 1) = T.$$

Cuts can alternatively be defined without mentioning the Polish expressions as follows: if a binary address α contains m times 1, it has the form $0^{r_0}10^{r_1}1\cdots 10^{r_m}$, and, then, when α is the address of a leaf in T, one has

$$\mathsf{cut}(T,\alpha) = T_{/\alpha_0} \triangleright (T_{/\alpha_1} \triangleright \cdots (T_{/\alpha_{m-1}} \triangleright T_{/\alpha_m}) \cdots)),$$

where we put $\alpha_k := 0^{r_0}10^{r_1}1\cdots 10^{r_k+1}$ for $0 \leqslant k < m$, and $\alpha_m := \alpha$.

The main result now is that the cuts of the term ∂T of (2.2) can be simply described from those of T, inductively leading to a description of normal terms.

Definition 3.11 (**descent**). *If α, β are binary addresses, declare $\alpha \ggg \beta$ if there exists an adress β satisfying $\alpha = \gamma 1^p$ and $\beta = \gamma 0\delta$ for some $p \geqslant 1$ and δ. Define a* descent *in T to be a finite sequence of addresses $(\alpha_1, ..., \alpha_m)$ such that, for every i, the address α_i is that of a leaf in T and $\alpha_i \ggg \alpha_{i+1}$ holds for $i < m$.*

Lemma 3.12. *There exists a unique bijection ϕ between the leaves of ∂T and the descents of T such that $\phi(\alpha) = (\alpha_1, ..., \alpha_m)$ implies*

$$\mathsf{cut}(\partial T, \alpha) =_{\mathsf{LD}} \mathsf{cut}(T, \alpha_1) \triangleright (\mathsf{cut}(T, \alpha_2) \triangleright \cdots (\mathsf{cut}(T, \alpha_{m-1}) \triangleright \mathsf{cut}(T, \alpha_m)) \cdots)).$$

So the cuts of ∂T, hence in particular its iterated left subterms, are simply defined in terms of those of T. In this way, we inductively obtain a description of all normal terms in terms of nested sequences of leaf addresses in $x^{[n]}$ and the associated cuts, which are the terms $x^{[i]}$ with $i < n$. In other words, we obtain a unique distinguished expression for every term in terms of $x^{[i]}$. Say that a normal term has *degree p* if it occurs as a cut of $\partial^p x^{[n]}$ and p is minimal with that property.

Example 3.13 (**normal terms**). *(See Fig. 4.) Here we describe all normal terms of degree at most 3 below $x^{[3]}$. For degree 0, there are two proper cuts of $x^{[3]}$, namely $x^{[1]}$ (that is, x) and $x^{[2]}$, hereafter abridged as 1 and 2.*

For degree 1, excluding (11), there are three descents in $x^{[3]}$, namely (0), (10), and (10, 0), leading to three proper cuts of $\partial x^{[3]}$, namely $x^{[1]}$, $x^{[2]}$, and $x^{[2]} \triangleright x^{[1]}$, or 1, 2, and 21• in abridged Polish notation. As 1 and 2 already appeared as cuts of $x^{[3]}$, there is only one normal term of degree 1, namely 21•.

For degree 2, excluding (11), there are five descents in $\partial x^{[3]}$, namely (00), (01), (10), (10, 00), and (10, 01), leading to five proper cuts of $\partial^2 x^{[3]}$, namely, in abridged notation, 1, 2, 21•, 21•1•, and 21•2•. As 1, 2, and 21• already appeared as cuts of $\partial x^{[3]}$, there are two normal terms of degree 2, namely 21•1• and 21•2•.

For degree 3, excluding (11), there are eleven descents in $\partial^2 x^{[3]}$, namely (000), (001), (01), (100), (100, 000), (100, 001), (100, 01), (101), (101, 000), (101, 001), and (101, 01), leading to eleven proper cuts of $\partial^3 x^{[3]}$, namely 1, 2, 21•, 21•1•, 21•1•1•, 21•1•2• , 21•1•21••, 21•2•, 21•2•1•, 21•2•2•, and 21•2•21••. As 1, 2, 21•, 21•1•, and 21•2• already appeared as cuts of $\partial^2 x^{[3]}$, there are six normal terms of degree 3, namely 21•1•1•, 21•1•2•, 21•1•21••, 21•2•1•, 21•2•2•, and 21•2•21••.

It follows from the proof of Prop. 3.7 (or rather of its counterpart stated in terms of cuts rather than in terms of iterated left subterms) that, for every T in $\mathsf{Term}_{\triangleright}(x)$, one can effectively find an LD-expansion of T that is a cut of some term $\partial^p x^{[n]}$ and, from there, determine the unique normal term that is LD-equivalent to T. Therefore, one obtains in this way a new solution for the word problem of $=_{\mathsf{LD}}$.

Example 3.14 (**normal form solution**). *Consider $T := (x \triangleright x) \triangleright (x \triangleright (x \triangleright x))$ and $T' := x \triangleright ((x \triangleright x) \triangleright (x \triangleright x))$ once more. The normal form of T is found by looking*

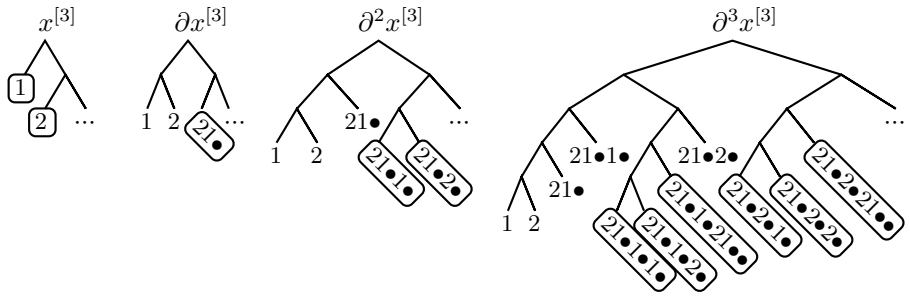


FIGURE 4. The terms $\partial^p x^{[3]}$ for $0 \leqslant p \leqslant 3$, and the canonical decompositions of their cuts: for each degree, the cuts that are normal, that is, that do not appear in a lower degree, are framed.

for a term $\partial^p x^{[n]}$ and an address α such that $\mathsf{cut}(\partial^p x^{[n]})$ is an LD*-expansion of T: here one easily sees that $\mathsf{cut}(\partial x^{[5]}, 011)$ is convenient both for T and for T', and one concludes that the normal form of both T and T' is the normal term $x^{[4]}$.*

3.3. **The case of more than one variable.** So far the word problem for LD was solved only in the case of terms involving one variable. Extending the solutions to the case of terms involving any number of variables is easy: essentially, the free left-shelf Free_n with $n \geqslant 2$ is a lexicographic extension of Free_1. Here is the key technical point:

Lemma 3.15. *Two terms T, T' such that, for some α and α', the cuts $\mathsf{cut}(T, \alpha)$ and $\mathsf{cut}(T', \alpha')$ coincide except in their rightmost variables, are never* LD*-equivalent.*

Proof. Assume for contradiction $T =_{\mathsf{LD}} T'$. Let x_i be the rightmost variable in $\mathsf{cut}(T, \alpha)$, and, respectively, let x_j be that of $\mathsf{cut}(T', \alpha')$. By the confluence property (Lemma 2.6), T and T' admit a common LD-expansion T''. Extending the proof of Lemma 2.4, one easily checks that T'' can be chosen so that some iterated left subterm $\mathsf{left}^r(T'')$, is an LD-expansion of $\mathsf{cut}(T, \alpha)$ and some iterated left subterm $\mathsf{left}^{r'}(T'')$ is an LD-expansion of $\mathsf{cut}(T', \alpha')$. By construction, the rightmost variable in $\mathsf{left}^r(T'')$ is x_i, whereas that of $\mathsf{left}^{r'}(T'')$ is x_j with $j \neq i$. Hence, we must have $r \neq r'$, say for instance $r < r'$. Let T''' be the term obtained from $\mathsf{left}^r(T'')$ by replacing the rightmost variable x_i with x_j. On the one hand, $\mathsf{left}^{r'}(T'')$ is a proper iterated left subterm of $\mathsf{left}^r(T'')$, and also of T''', hence $\mathsf{cut}(T', \alpha') \sqsubset^*_{\mathsf{LD}} T'''$ holds. On the other hand, by construction again, T''' is an LD-expansion of the term obtained from $\mathsf{cut}(T, \alpha)$ by replacing the final occurrence of x_i with x_j, which, by assumption, is $\mathsf{cut}(T', \alpha')$, hence $\mathsf{cut}(T', \alpha') =_{\mathsf{LD}} T'''$ holds. The conjunction of the above relations contradicts the exclusion property (Lemma 2.10). □

For T in $\mathsf{Term}_\triangleright(X)$, we denote by $\overline{T}$ the projection of T where every variable of X is mapped to x. Clearly, $T =_{\mathsf{LD}} T'$ implies $\overline{T} =_{\mathsf{LD}} \overline{T'}$. The converse implication need not be true, but we have

Lemma 3.16. *Assume that T and T' are terms in $\mathsf{Term}_\triangleright(X)$ satisfying $\overline{T} = \overline{T'}$. Then $T =_{\mathsf{LD}} T'$ holds if, and only if, T and T' coincide.*

Proof. If T and T' do not coincide, there must exist addresses α, α' as in Lemma 3.15, and the latter then implies that T and T' are not LD-equivalent. □

We deduce an solution for the word problem that directly extends Algorithm 2.12:

Algorithm 3.17 (**syntactic solution, general case**).
Input: Two terms T, T' in $\mathsf{Term}_{\triangleright}(X)$
Output: `true` if T and T' are LD-equivalent, `false` otherwise

```
FOUND := false
n := 0
while not FOUND do
   T_1 := LD-EXPAND(T, s_n)
   T'_1 := LD-EXPAND(T', s'_n)
   if T_1 = T'_1 then
      return true
      FOUND := true
   else if T̄_1 = T̄'_1 or T̄_1 ⊏* T̄'_1 or T̄_1 ⊐* T̄'_1 then
      return false
      FOUND := true
   n := n + 1
```

Extending the semantic algorithm of Section 2.4 is also possible. No realization of Free_n with $n \geqslant 2$ inside Artin's braid group B_∞ is known (but there is no proof that such a realization cannot exist). However, one can easily define variants of B_∞ with enough space to build such realizations. A first solution was described by D. Larue in [44]. Another one with a simple geometric interpretation ("charged braids") appears in [15].

3.4. **The Polish algorithm.** We conclude with one more approach to the word problem of LD, which leads to a very puzzling open question.

As in Section 3.2, let us consider the Polish expression of terms. By definition, expanding T to T' using the LD-law means replacing some subterm of T of the form $T_1 \triangleright (T_2 \triangleright T_3)$ with $(T_1 \triangleright T_2) \triangleright (T_1 \triangleright T_3)$. On Polish expressions, this means that $[T']$ is obtained from $[T]$ by replacing a factor of $[T]$ of the form

$$[T_1][T_2][T_3]\bullet\bullet \qquad \text{with} \qquad [T_1][T_2]\bullet[T_1][T_3]\bullet\bullet :$$

starting from the left, the words $[T]$ and $[T']$ coincide up to the last letter from $[T_2]$, which is followed by $\bullet$ in $[T']$, whereas, in $[T]$, it is followed by the first letter of $[T_3]$, which is necessarily a variable. This suggests, for comparing two terms T, T', to look at the first discrepancy between $[T]$ and $[T']$, try to expand the term where a variable occurs, and repeat until one possibly obtains twice the same word. This is what we shall call the "*Polish algorithm*". To give a precise description, let us review all possible relations between $[T]$ and $[T']$:

- If $[T]$ and $[T']$ coincide, then T and T' are equal, hence LD-equivalent.

- If $[T]$ is a proper prefix of $[T']$, or vice versa, then $T \sqsubset^* T'$ or $T' \sqsubset^* T$ holds, hence, by the exclusion property (Lemma 2.10), T and T' are not LD-equivalent.

- Otherwise, $[T]$ and $[T']$ have a first letter clash. If this clash is of the type "some x_i *vs.* some x_j with $i \neq j$", then, by Lemma 3.15, T and T' are not LD-equivalent.

- The remaining case is a first letter clash of the type "some x_i *vs.* $\bullet$". Then we shall see that there is a unique solution to push the clash further to the right, by expanding the term where x_i occurs. Consider a term of the form $T_1 \triangleright ((T_2 \triangleright T_3) \triangleright T_4)$: the Polish expression is $[T_1][T_2][T_3]\bullet[T_4]\bullet\bullet$. Because T_2 is nested in $T_2 \triangleright T_3$, in order

to insert a letter $\bullet$ after $[T_2]$, we need to first distribute T_1 to $T_2 \triangleright T_3$, and then distribute T_1 to T_2, which amounts to successively applying $\mathrm{LD}_\emptyset$ and LD_0, obtaining $[T_1][T_2]\bullet[T_1][T_3]\bullet\bullet[T_1][T_4]\bullet\bullet$, which has the expected form with a $\bullet$ after $[T_2]$. The situation is generic:

Definition 3.18 (**solution**). *Let α be an address that contains the factor* 10. *Write α as $\beta 10^p 1^r$ with $p \geqslant 1$ and $r \geqslant 0$. Then we define the* solution *at α to be the finite sequence* $\mathrm{SOL}(\alpha) := (\beta, \beta 0, \beta 00, ..., \beta 0^p)$.

Then $\mathrm{SOL}(\alpha)$ provides a recipe for replacing a variable with the symbol $\bullet$ at a prescribed position in the Polish expansion of a term:

Lemma 3.19 (**solution**). *The sequence* $\mathrm{SOL}(\alpha)$ *is the unique finite sequence s with the following property. Assume that T is a term in which there is a letter with address α in $[T]$ and the next letter in $[T]$ is a variable. Then the LD-expansion $T' := T \bullet \mathrm{LD}_s$ is defined,$[T']$ coincides with $[T]$ up to the letter with address α, and the next letter in $[T']$ is $\bullet$.*

The method can then be implemented as follows. For any two terms T, T', we define $\mathrm{DISC}(T, T')$ to be, when it exists, the pair (p, i) in $\mathbb{N} \times \{1, 2\}$ such that p is the length of the longest common initial factors between $[T]$ and $[T']$ and the $(p+1)$st letter in T is a variable, whereas the $(p+1)$st letter in T' is $\bullet$, in which case we put $i := 1$, or vice versa, in which case we put $i := 2$. We recall that $\mathsf{add}(p, T)$ is the address in T that corresponds to the pth letter in $[T]$ under the correspondence described below Definition 3.8.

Algorithm 3.20 (**Polish algorithm**).
Input: Two terms T, T' in $\mathsf{Term}_\triangleright(X)$
Output: `true` if T and T' are LD-equivalent, `false` otherwise

while $\mathrm{DISC}(T, T')$ is defined **do**
  $(p, i) := \mathrm{DISC}(T, T')$
  if $i = 1$ **then**
    $\alpha := \mathsf{add}(p, T)$
    $T := \text{LD-EXPAND}(T, \mathrm{SOL}(\alpha))$
  if $i = 2$ **then**
    $\alpha := \mathsf{add}(p, T')$
    $T' := \text{LD-EXPAND}(T', \mathrm{SOL}(\alpha))$
if $T = T'$ **then**
  return `true`
if $T \sqsubset^* T'$ **or** $T' \sqsubset^* T$ **then**
  return `false`

Example 3.21 (**Polish algorithm**). *Consider $T := (x_1 \triangleright x_2) \triangleright (x_1 \triangleright (x_3 \triangleright x_4))$ and $T' := x_1 \triangleright ((x_2 \triangleright x_3) \triangleright (x_2 \triangleright x_4))$, a multi-variable version of the terms of Examples 2.14 and 2.22. Then we start with*

$[T_0] = x_1 x_2 \parallel \bullet x_1 x_3 x_4 \bullet\bullet\bullet$,
$[T'_0] = x_1 x_2 \parallel x_3 \bullet x_2 x_4 \bullet\bullet\bullet$
(*we use the symbol $\parallel$ to emphasize the longest common prefix*).

The words $[T_0]$ and $[T'_0]$ have a clash after the second letter, followed by $\bullet$ in $[T_0]$ and by x_3 in $[T'_0]$. Thus $\mathrm{DISC}(T_0, T'_0)$ is defined, and equal to $(2, 2)$. We then find $\mathsf{add}(2, T'_0) = 100$, $\mathrm{SOL}(100) = (\emptyset, 0)$, and LD-expanding T'_0 at $\emptyset$ and then at 0 yields

$[T_1] = x_1 x_2 \bullet x_1 x_3 \parallel x_4 \bullet\bullet\bullet$,

$[T'_1] = x_1x_2\bullet x_1x_3 \parallel \bullet\bullet x_1x_2x_4\bullet\bullet\bullet$.

Now $[T_1]$ *and* $[T'_1]$ *have a clash after the fifth letter, followed by* x_3 *in* $[T_1]$ *and by* $\bullet$ *in* $[T'_1]$. *Thus* $\mathrm{DISC}(T_1, T'_1)$ *is defined, and equal to* $(5, 1)$. *We find* $\mathsf{add}(5, T_1) = 110$, $\mathrm{SOL}(110) = (1)$, *and expanding* T_1 *at* 1 *yields*

$[T_2] = x_1x_2\bullet x_1x_3\bullet \parallel x_1x_4\bullet\bullet\bullet$,
$[T'_2] = x_1x_2\bullet x_1x_3\bullet \parallel \bullet x_1x_2x_4\bullet\bullet\bullet$.

Then $[T_2]$ *and* $[T'_2]$ *have a clash after the sixth letter, followed by* x_1 *in* $[T_2]$ *and by* $\bullet$ *in* $[T'_2]$. *Thus* $\mathrm{DISC}(T_2, T'_2)$ *is defined, and equal to* $(6, 1)$. *We find* $\mathsf{add}(6, T_2) = 10$, $\mathrm{SOL}(10) = (\emptyset)$, *and expanding* T_2 *at* $\emptyset$ *yields*

$[T_3] = x_1x_2\bullet x_1x_3\bullet\bullet x_1x_2 \parallel \bullet x_1x_4\bullet\bullet\bullet$,
$[T'_3] = x_1x_2\bullet x_1x_3\bullet\bullet x_1x_2 \parallel x_4\bullet\bullet\bullet$.

The words $[T_3]$ *and* $[T'_3]$ *have a clash after the ninth letter, followed by* $\bullet$ *in* $[T_3]$ *and by* x_4 *in* $[T'_3]$. *Thus* $\mathrm{DISC}(T_3, T'_3)$ *is defined, and equal to* $(10, 2)$. *We find* $\mathsf{add}(10, T'_3) = 110$, $\mathrm{SOL}(110) = (1)$, *and expanding* T'_3 *at* 1 *yields*

$[T_4] = x_1x_2\bullet x_1x_3\bullet\bullet x_1x_2\bullet x_1x_4\bullet\bullet\bullet$,
$[T'_4] = x_1x_2\bullet x_1x_3\bullet\bullet x_1x_2\bullet x_1x_4\bullet\bullet\bullet$.

The words $[T_4]$ *and* $[T'_4]$ *are equal: the Polish Algorithm running on the pair* (T, T') *converges to* (T_4, T_4) *in 4 steps. We conclude that* T *and* T' *are* LD*-equivalent.*

As in the case of Algorithms 2.12 and 3.17, the correctness of the Polish algorithm follows from the exclusion property (Lemma 2.10): if the algorithm converges to a pair of equal terms (T_n, T'_n), the term T_n is a common LD-expansion of the initial terms, and the latter are LD-equivalent; if the algorithm converges to a pair of terms (T_n, T'_n) with $T_n \sqsubset^* T'_n$ or $T'_n \sqsubset^* T_n$, then, by the exclusion property, T_n and T'_n cannot be LD-equivalent and, therefore, the initial terms cannot either.

But this leaves the following question open:

Question 3.22 (convergence). *Does the Polish algorithm always converge?*

The problem seems very difficult—and, with the Embedding Conjecture (Question 3.6), it is the most puzzling open problem involving syntactic aspects of selfdistributivity. Experimental results and a number of partial results [14, 20] suggest a positive answer, but no complete result is known so far. Due to the nature of selfdistributivity, the sizes of LD-expansions quickly increase (see Fig. 4). However, many patterns are repeated in such LD-expansions, and clever encodings can lower the size, on the shape of S. Schleimer's approach in [56]. By doing so, O. Deiser [26] could perform exhaustive search for all pairs of terms up to size 9: interestingly, he discovered that the Polish algorithm usually converges very fast, except in a few isolated cases, where it still converges, but in an extremely long time. Let us mention that other algebraic laws are eligible for the Polish algorithm: as can be expected, convergence in the case of associativity is trivial, whereas selfdistributivity seems to be the most difficult one—but many questions remain open [26].

4. Word problem, the case of racks, quandles, and spindles

We conclude the survey with a similar investigation of the word problem for the three derived notions obtained by adding additional laws to selfdistributivity, namely racks, quandles, and spindles. We shall successively review the case of free racks and free quandles, which are very similar and easy (Subsection 4.1), and finish with the case of free spindles, which seems very difficult (Subsection 4.2).

4.1. **The case of racks and quandles.** The definition of racks and quandles comes in two versions, according to whether one or two binary operations are considered. If only one operation is concerned, the condition that the right translations are bijective does not correspond to obeying an algebraic law, and there is no natural word problem. By contrast, when two operations $\lhd, \overline{\lhd}$ are considered, being a rack (*resp.* a quandle) corresponds to obeying (RD) plus the two algebraic laws of (1.1) (*resp.* these, plus the idempotency law $x \lhd x = x$), and the word problem is a well posed question. We denote by $=_{\mathsf{rack}}$ and $=_{\mathsf{quandle}}$ the congruences on $\mathsf{Term}_{\lhd,\overline{\lhd}}(X)$ generated by the instances of the laws RD and (1.1) (*resp.* these plus idempotency). By construction, the quotient-structure $\mathsf{Term}_{\lhd,\overline{\lhd}}(X)/=_{\mathsf{rack}}$ is a free rack based on X, and, similarly, $\mathsf{Term}_{\lhd,\overline{\lhd}}(X)/=_{\mathsf{quandle}}$ is a free quandle based on X.

We shall see that the word problems are easy here, because there exists a simple family of distinguished (or "normal") terms. The general principle is as follows:

Lemma 4.1. *Let $\mathcal{L}$ be a family of algebraic laws involving a signature Σ. Let $\mathcal{N}$ be a subset of $\mathsf{Term}_\Sigma(X)$. Let $\mathcal{S}$ be a Σ-structure generated by X. Assume that*
(i) *every term in $\mathsf{Term}_\Sigma(X)$ is $=_{\mathcal{L}}$-equivalent to at least one term in $\mathcal{N}$,*
(ii) *distinct terms in $\mathcal{N}$ have distinct evaluations in $\mathcal{S}$.*
Then $\mathcal{S}$ is a free $\mathcal{L}$-algebra based on X.

Proof. Write $\mathsf{eval}(T, \mathcal{S})$ for the evaluation of a term T in $\mathcal{S}$. Let T, T' be two terms in $\mathsf{Term}_X(\Sigma)$. By (i), there exist T_1 and T_1' in $\mathcal{N}$ satisfying $T =_{\mathcal{L}} T_1$ and $T' =_{\mathcal{L}} T_1'$. As $=_{\mathcal{L}}$ is transitive, $T =_{\mathcal{L}} T'$ is equivalent to $T_1 =_{\mathcal{L}} T_1'$, hence, by (ii), to $\mathsf{eval}(T_1, \mathcal{S}) = \mathsf{eval}(T_1', \mathcal{S})$. As $\mathcal{S}$ is an $\mathcal{L}$-algebra, $T =_{\mathcal{L}} T_1$ implies $\mathsf{eval}(T, \mathcal{S}) = \mathsf{eval}(T_1, \mathcal{S})$ and, similarly, $\mathsf{eval}(T', \mathcal{S}) = \mathsf{eval}(T_1', \mathcal{S})$. Finally, we deduce that $T =_{\mathcal{L}} T'$ is equivalent to $\mathsf{eval}(T, \mathcal{S}) = \mathsf{eval}(T', \mathcal{S})$. Hence, $\mathcal{S}$ is free. □

Applying this to the cases of racks and quandles is easy.

Proposition 4.2 (realization). *Let F_X be a free group based on a set X.*
(i) *The structure $\mathsf{HalfConj}(X, F_X)$ is a free rack based on X.*
(ii) *The structure $\mathsf{Conj}_{\lhd,\overline{\lhd}}(F_X)$ is a free quandle based on X.*

Proof (principle). Write $\lhd^{+1}$ for $\lhd$ and $\lhd^{-1}$ for $\overline{\lhd}$, and define $\mathcal{N}$ to be the family of all terms of the particular form

$$(\cdots ((x \lhd^{e_1} x_1) \lhd^{e_2} x_2) \cdots) \lhd^{e_n} x_n, \tag{4.1}$$

with $n \geqslant 0$, $x, x_1, ..., x_n \in X$, $e_1, ..., e_n = \pm 1$, and $e_i \neq e_{i+1}$ implying $x_i \neq x_{i+1}$. An easy induction shows that every term in $\mathsf{Term}_{\lhd,\overline{\lhd}}(X)$ is $=_{\mathsf{rack}}$-equivalent to a term as in (4.1). Next, the evaluation of such a term in $\mathsf{HalfConj}(X, F_X)$ is the pair

$$(x, x_1^{e_1} x_2^{e_2} \cdots x_n^{e_n}),$$

where the second entry is a freely reduced signed X-word. Hence distinct terms of the form (4.1) have distinct $\mathsf{HalfConj}(X, F_X)$-evaluations. By Lemma 4.1, the structure $\mathsf{HalfConj}(X, F_X)$ is a free rack based on X.

The argument is similar for quandles, demanding in addition $x_1 \neq x$ in (4.1). The evaluation in $\mathsf{Conj}_{\lhd,\overline{\lhd}}(F_X)$ is then the freely reduced signed X-word

$$x_n^{-e_n} \cdots x_2^{-e_2} x_1^{-e_1} x x_1^{e_1} x_2^{e_2} \cdots x_n^{e_n},$$

which again determines the term (4.1) it comes from. □

We derive a semantic algorithm for the word problems of racks and quandles. Below we use RED for free group reduction, that is, iteratively deleting length 2 factors of the form xx^{-1} or $x^{-1}x$. We use ε for the empty word.

Algorithm 4.3 (semantic algorithm for $=_{\mathsf{rack}}$).
Input: Two X-terms T, T'
Output: `true` if T and T' are $=_{\mathsf{rack}}$-equivalent, `false` otherwise

$(x, w) := \text{EVAL}(T)$
$(x', w') := \text{EVAL}(T')$
if $x \neq x'$ **or** $\text{RED}(w) \neq \text{RED}(w')$ **then**
  return `false`
else
  return `true`

function EVAL(T : term): element of $X \times (X \cup X^{-1})^*$
if $T = x \in X$ **then**
  return (x, ε)
else if $T = T_1 \lhd T_2$ **then**
  return $\text{EVAL}(T_1) \lhd \text{EVAL}(T_2)$ with $\lhd$ as in (1.4)
else if $T = T_1 \overline{\lhd} T_2$ **then**
  return $\text{EVAL}(T_1) \overline{\lhd} \text{EVAL}(T_2)$ with $\overline{\lhd}$ as in (1.5)

Composing free reduction with the function EVAL provides the evaluation in the structure $\mathsf{HalfConj}(X, F_X)$. So, Algorithm 4.3 is a direct translation of Prop. 4.2(i).

An entirely similar algorithm solves the word problem for the quandle laws, at the expense of replacing the evaluation in $\mathsf{HalfConj}(X, F_X)$ with one in $\mathsf{Conj}_{\lhd,\overline{\lhd}}(F_X)$ using the conjugacy operations of (1.2), and appealing to Prop. 4.2(ii).

4.2. **The case of spindles.** Let us finally address the word problem for free spindles, that is, the question of deciding whether two terms T, T' are LDI-equivalent, where LDI refers to the conjunction of (left) selfdistributivity LD and idempotency I. We use $=_{\mathsf{LDI}}$ for the associated congruence on terms.

The word problem of LDI is trivial for terms involving only one variable, as an easy induction gives $T =_{\mathsf{LDI}} x$ for every T in $\mathsf{Term}_{\rhd}(x)$. As a consequence, the free (left) spindle on one generator has only one element.

Frustratingly, this case is the only one, for which a solution is known. No semantic solution is known, because no realization of free (left) spindles is known. As observed in Example 1.10, the conjugacy structure $\mathsf{Conj}_{\lhd}G$ of any group G is a quandle, hence a spindle. One might think that, starting with a free group, the associated conjugacy spindle might be free. This is not true:

Proposition 4.4 (not free). *If G is a group distinct of $\{1\}$, the conjugacy spindle $\mathsf{Conj}_{\lhd}G$ is not free.*

Principle of proof. There exist algebraic laws satisfied by conjugation and not consequences of LDI [33], a typical example being

$$((x \rhd y) \rhd y) \rhd (y \rhd z) = (x \rhd y) \rhd ((y \rhd x) \rhd z). \tag{4.2}$$

Then (4.2) holds in every conjugacy left-spindle since, when $x \rhd y$ is evaluated to xyx^{-1}, the two terms of (4.2) evaluate to $xyx^{-1}yxy^{-1}zyx^{-1}y^{-1}xy^{-1}x^{-1}$, whereas (4.2) fails for $x = z = 2$ and $y = 3$ in the four-element left-spindle whose table is shown on the right.

1	1	2	3	4
1	1	2	3	4
2	1	2	1	2
3	3	4	3	4
4	1	2	3	4

An infinite family of similar laws is constructed in [45], and it is shown that no finite subfamily generates it. Also see [58] and [17]. □

In the direction of syntactic solutions, a counterpart of the approach of Section 3.1 was successfully developed by P. Jedlička in [38, 39, 40]: in the case of LDI, the confluence property still holds, and one can investigate a geometry monoid $\mathsf{Geom}_{\mathrm{LDI}}$ similar to $\mathsf{Geom}_{\mathrm{LD}}$ but, at least because no relevant blueprint was found so far, the approach falls short of solving the word problem. Thus, once again, we meet with a puzzling open question:

Question 4.5. *Is the word problem for* LDI *solvable?*

A positive answer may seem likely, but it remains unknown so far.

References

[1] V.D. Belousov, *Transitive distributive quasigroups (Tranzitivnyje distributivnyje kvazygruppy)*, Ukr. Mat. Žh. **10** (1958) 13–22.

[2] V.D. Belousov, *On the structure of distributive quasigroups (O strykture distributivnyh kvazigrupi)*, Math. Sb. **50** (1960) 267–298.

[3] J. Birman, *Braids, Links, and Mapping Class Groups*, Annals of Math. Studies **82** Princeton Univ. Press (1975).

[4] J.W. Cannon, W.J. Floyd, & W.R. Parry, *Introductory notes on Richard Thompson's groups*, Enseign. Math. **42** (1996) 215–256.

[5] J.S. Carter, *A survey of quandle ideas*, in: Introductory lectures on Knot Theory, J.S. Carter, L. Kauffmann and al. eds, Series on Knots and Everything vol. 46, World Scientific (2012), pages 22–53.

[6] J.S. Carter, D. Jelsovsky, S. Kamada, L. Langford, & M. Saito, *Quandle cohomology and state-sum invariants of knotted curves and surfaces*, Trans. Am. Math. Soc. **355-10** (2003) 3947–3989.

[7] J.S. Carter, S. Kamada, & M. Saito, *Geometric interpretations of quandle homology*, J. Knot Theory Ramifications. **10** (2001) 345–386.

[8] P. Dehornoy, Π^1_1*-complete families of elementary sequences*, Ann. Pure Appl. Logic **38** (1988) 257–287.

[9] P. Dehornoy, *Algebraic properties of the shift mapping*, Proc. Am. Math. Soc. **106-3** (1989) 617–623.

[10] P. Dehornoy, *Preuve de la conjecture d'irréflexivité pour les structures distributives libres*, C. R. Acad. Sci. Paris **314** (1992) 333–336.

[11] P. Dehornoy, *Sur la structure des gerbes libres*, C. R. Acad. Sci. Paris **309** (1989) 143–148.

[12] P. Dehornoy, *Braid groups and left distributive operations*, Trans. Am. Math. Soc. **345-1** (1994) 115–150.

[13] P. Dehornoy, *A normal form for the free left distributive law*, Int. J. Algebra Comput **4-4** (1994) 499–528.

[14] P. Dehornoy, *On the syntactic algorithm for the word problem of left distributivity*, Algebra Univers. **37** (1997) 191–222.

[15] P. Dehornoy, *Construction of left distributive operations and charged braids*, Int. J. Algebra Comput. **10-1** (2000) 173–190.

[16] P. Dehornoy, *A fast method for comparing braids*, Adv. Math. **125** (1997) 200–235.

[17] P. Dehornoy, *A conjecture about conjugacy in free group*, Discuss. Math., Algebra Stoch. Methods **19** (1999) 75–112.

[18] P. Dehornoy, *Strange questions about braids*, J. Knot Theory Ramifications **8-5** (1999) 589–620.

[19] P. Dehornoy, *Braids and Self-Distributivity*, Progress in Math. vol. 192, Birkhäuser (2000).

[20] P. Dehornoy, *The fine structure of LD-equivalence*, Adv. Math. **155** (2000) 264–316.

[21] P. Dehornoy, *Study of an identity*, Algebra Univ. **48** (2002) 223–248.

[22] P. Dehornoy, *Elementary embeddings and algebra*, Chapter 11 in: Handbook of Set Theory, vol. 2 (Foreman, Kanamori, Eds.), Springer, pp. 737–774 (2010).

[23] P. Dehornoy, *The group of parenthesized braids*, Adv. Math. **205** (2006) 354–409.

[24] P. Dehornoy, I. Dynnikov, D. Rolfsen, & B. Wiest, *Ordering Braids*, Mathematical Surveys and Monographs vol. 148, Amer. Math. Soc. (2008).
[25] P. Dehornoy, *The braid shelf*, J. Knot Theory Ramifications, to appear; arXiv:1711.09794.
[26] O. Deiser, *Notes on the Polish Algorithm*, arXiv 1711.09580
[27] V. Dimonte, *LD-algebras beyond I0*, arXiv:1701.01343.
[28] R. Dougherty, *Critical points in an algebra of elementary embeddings*, Ann. Pure Appl. Logic **65** (1993) 211–241.
[29] R. Dougherty & T. Jech, *Finite left-distributive algebras and embedding algebras*, Adv. Math. **130** (1997) 201–241.
[30] A. Drápal, *Finite left distributive algebras with one generator*, J. Pure Appl. Algebra **121** (1997) 233–251.
[31] A. Drápal, *Finite left distributive groupoids with one generator*, Int. J. Algebra Comput. **7-6** (1997) 723–748.
[32] A. Drápal, *About Laver tables*, In this volume.
[33] A. Drápal, T. Kepka & M. Musílek, *Group conjugation*, Commentat. Math. Univ. Carol. **35-2** (1994) 219–222.
[34] D. Epstein, J. Cannon, D. Holt, S. Levy, M. Paterson & W. Thurston, *Word Processing in Groups*, Jones & Bartlett Publ. (1992).
[35] R. Fenn & C.P. Rourke, *Racks and links in codimension 2*, J. Knot Theory Ramifications **1** (1992) 343–406.
[36] R. Fenn, C.P. Rourke, & B. Sanderson, *James bundles and applications* , http://www.mps.warwick.ac.uk/∼cpr/ftp/james.pdf.
[37] T. Jech, *Large ordinals*, Adv. Math. **125** (1997) 155–170.
[38] P. Jedlička, *On left distributive left idempotent groupoids*, Comment. Math. Univ. Carol. **46-1** (2005) 15-20.
[39] P. Jedlička, *Geometry monoid of the left distributivity and the left idempotency*, Algebra Discrete Math.; 4; 2006; 12–39.
[40] P. Jedlička, *On a partial syntactical criterion for the left distributivity and the idempotency*, Math. Slovaca; 60; 2010; 213–222.
[41] J. Ježek, T. Kepka & P. Němec, *Distributive groupoids*, Rozpr. Cesk. Akad. Ved, Rada Mat. Přirod. Ved **91** (1981) 1–94.
[42] D. Joyce, *A classifying invariant of knots, the knot quandle*, J. Pure Appl. Algebra **23** (1982) 37–65;
[43] D.M. Larue, *On braid words and irreflexivity*, Algebra Univers. **31** (1994) 104–112.
[44] D.M. Larue, *Left-distributive and left-distributive idempotent algebras*, Ph D Thesis, University of Colorado, Boulder (1994).
[45] D.M. Larue, *Left-distributive idempotent algebras*, Commun. Algebra **27(5)** (1999) 2003-2029.
[46] R. Laver, *Elementary embeddings of a rank into itself*, Abstr. Amer. Math. Soc. **7** (1986) 6.
[47] R. Laver, *The left distributive law and the freeness of an algebra of elementary embeddings*, Adv. Math. **91-2** (1992) 209–231.
[48] R. Laver, *A division algorithm for the free left distributive algebra*, Oikkonen et *al.* eds, Logic Colloquium '90, Lect. Notes Log. **2** (1993) 155–162.
[49] R. Laver, *On the algebra of elementary embeddings of a rank into itself*, Adv. Math. **110** (1995) 334–346.
[50] R. Laver & S.K. Miller, *The free one-generated left distributive algebra: basics and a simplified proof of the division algorithm*, Cent. Eur. J. Math. **11** (2013) 2150–2175.
[51] S.V. Matveev, *Distributive groupoids in knot theory*, Math. USSR Sb. **47** (1984) 73–83.
[52] R. McKenzie & R.J. Thompson, *An elementary construction of unsolvable word problems in group theory*, in: Word Problems, W.W. Boone and al. eds, Studies Logic and Foundations Math., vol. 71, North Holland (1973), pp 457-478.
[53] F. Müller-Hoissen et al., *Associahedra, Tamari lattices and related structures*, Progress in Mathematics, vol. 299, Birkhäuser/Springer (2012).
[54] M.H.A. Newman, *On theories with a combinatorial definition of "equivalence"*, Ann. Math. **43** (1942) 223–243.
[55] C.S. Peirce, *On the algebra of logic*, Am. J. Math. **3** (1880) 15–57.
[56] S. Schleimer, *Polynomial-time word problems*, Comment. Math. Helv. **83** (2008) 741–765.
[57] M. Smedberg, *A dense family of well-behaved finite monogenerated left-distributive groupoids*, Arch. Math. Logic **52** (2013) 377–402.

[58] D. Stanovský, *On the equational theory of group conjugation*, in: Contributions to general algebra. 15, Heyn, Klagenfurt (2004) pp 177–185.
[59] R.J. Thompson, *Tranformation structure of algebraic logic*, PhD Thesis Univ. of Berkeley; ProQuest LLC, Ann Arbor, MI (1979), 345 pages, MR2628711.

Laboratoire de Mathématiques Nicolas Oresme UMR 6139, Université de Caen, 14032 Caen, France
E-mail address: patrick.dehornoy@unicaen.fr
URL: dehornoy.users.lmno.cnrs.fr