SIGNED-XOR STRUCTURAL CONSTANTS AND TABLE-FREE MULTIPLICATION IN CLIFFORD AND CAYLEY–DICKSON ALGEBRAS

PARK JUNHU

Abstract. We formulate a signed-XOR structural-constant calculus for basis multiplication in Clifford and Cayley–Dickson type algebras. The common support law is addition in an elementary abelian 2-group, while the algebra-specific content is carried by a sign twist. In the sorted Clifford sector, the quotient presentation with generators satisfying $e_i^2 = -1$ and $e_i e_j = -e_j e_i$ admits the closed product formula

$$e_J e_K = (-1)^{\mathrm{inv}(J,K)+|J\cap K|} e_{J\triangle K}.$$

We prove the corresponding normal-form theorem, identify the finite case with $\mathrm{Cl}_{0,n}(\mathbb{R})$, and express the sign in bit-vector form as $S(a,b) = a^T(U_n + I_n)b \pmod 2$. For general normalized signed-XOR twists we compute the associator, commutator, square, and gauge transformation formulae, thereby separating the XOR support law from the associativity obstruction. The classical Cayley–Dickson basis law is then recovered as a recursive twist with the same support update. The algorithmic consequence is table-free evaluation of each encountered structural constant: sparse multiplication still has the usual bilinear dependence on the number of stored terms, but no basis-multiplication table of size 4^n is required for the sorted sector.

Email: nller.park@snu.ac.kr.

1. Introduction

This article develops a signed-XOR structural-constant calculus for basis multiplication in Clifford and Cayley–Dickson type algebras. The guiding observation is that many of the relevant bases share the same support update: the product of two basis elements is supported on the

2020 *Mathematics Subject Classification.* Primary 65F99, 68W30; Secondary 15A66, 16S99, 17A35, 20C05.

Key words and phrases. Signed-XOR calculus, structural constants, Clifford algebras, Cayley–Dickson algebras, twisted group algebras, table-free multiplication, computational algebra.

XOR, or symmetric difference, of their supports. What distinguishes the examples is the sign attached to this support update.

In this formalism a basis product has the form

$$e_\alpha e_\beta = \theta(\alpha, \beta) e_{\alpha+\beta},$$

where the support labels form an elementary abelian 2-group and $\theta(\alpha, \beta) \in \{\pm 1\}$ is a sign twist. Associative and nonassociative examples differ not in the XOR support law, but in the twist and in the resulting associator sign. The algebraic setting is that of Clifford, composition-algebra, and nonassociative-algebra traditions [10, 8, 11, 12, 2].

From the numerical point of view, the claim is deliberately narrower than a new sparse-tensor paradigm. Sparse storage already replaces the ambient number of basis elements by the number of stored nonzero coefficients. The additional point here is that each basis product can be evaluated without a stored multiplication table: the output support is obtained by XOR and the sign is obtained by a parity rule. Thus the numerical contribution concerns the per-basis-product structural-constant evaluation and the associated memory footprint, not the elementary fact that sparse data structures avoid iterating over zero coefficients.

The signed-XOR formalism used here was first introduced, in broader programmatic form, in the thesis-level AXIOM framework [6]. The present article isolates the part of that construction that can be stated and proved as a self-contained algebraic normal-form and twist-calculus result.

1.1. **Main results.** The main results establish the following numerical and algebraic framework.

(1) We isolate a signed-XOR structural-constant formalism in which basis multiplication is decomposed into an elementary 2-group support law and a sign twist.
(2) In the associative Clifford sector, we prove a unique sorted normal form for $\mathcal{A}_{\mathrm{sort}}(I)$, including termination, spanning, and linear independence by a faithful model.
(3) We derive the structural constants of this sector as the closed signed-XOR product

$$e_J e_K = (-1)^{\mathrm{inv}(J,K)+|J\cap K|} e_{J\triangle K},$$

and show that the corresponding sorted sign is an associative cocycle.
(4) We identify the finite case with $\mathrm{Cl}_{0,n}(\mathbb{R})$ and the countable case with the filtered direct limit of finite Clifford algebras.

(5) For arbitrary normalized twists over elementary 2-groups, we compute the associator, commutator, square, and gauge transformation formulae, thereby separating the XOR support law from the associativity obstruction.
(6) We prove that classical Cayley–Dickson basis multiplication is obtained from a recursive sign twist with the same XOR support law.
(7) We record the algorithmic consequence: each encountered sorted structural constant can be evaluated by XOR and parity without storing the full basis-multiplication table, while sparse multiplication retains its usual bilinear dependence on the number of stored terms.

1.2. **Relation with existing constructions.** The plain sorted algebra is not proposed as a new Clifford algebra. Its finite form is precisely $\mathrm{Cl}_{0,n}(\mathbb{R})$ in the standard Clifford-algebra sense [10, 8]. In this paper it serves as the associative cocycle sector of a broader signed-XOR calculus. The role of the normal-form proof is to anchor the calculus in a quotient presentation and to make the structural constants explicit without appealing to a stored multiplication table.

The novelty of the paper is not the identification of the plain sector with a Clifford algebra. Rather, the contribution is the extraction of a uniform support/sign calculus in which the sorted Clifford sign is an associative cocycle, arbitrary signed-XOR twists have explicitly computable associators and gauge transformations, and the Cayley–Dickson basis law is recovered by a recursive nonassociative twist.

The Cayley–Dickson part shows that the same XOR support law also persists in the nonassociative setting after the sign is changed. The associator formula then records exactly where associativity fails. Thus the contribution is not a renaming of Clifford algebras, but a unified support/sign formalism for Clifford and Cayley–Dickson type basis multiplication.

The numerical layer should be read in the same restricted sense. For a sparse multivector with m stored terms, algorithms that only traverse the stored terms naturally scale with m rather than with the full ambient basis size. This is not a new consequence of the signed-XOR calculus. What the present formulation supplies is an explicit table-free rule for the structural constant attached to each encountered pair of supports.

2. Finite supports and inversion parity

Definition 2.1 (Finite support group)**.** Let I be a linearly ordered set. Let

$$\mathrm{Fin}(I) = \{J \subset I : |J| < \infty\}.$$

The group operation on $\mathrm{Fin}(I)$ is symmetric difference,

$$J \triangle K = (J \setminus K) \cup (K \setminus J).$$

Equivalently, $\mathrm{Fin}(I)$ is the elementary abelian 2-group of finite-support functions $I \to \mathbb{F}_2$.

Definition 2.2 (Sorted word)**.** For $J = \{j_1 < \cdots < j_m\} \in \mathrm{Fin}(I)$, define

$$e_J = e_{j_1} \cdots e_{j_m}, \qquad e_\varnothing = 1.$$

Definition 2.3 (Inversion count)**.** For $J, K \in \mathrm{Fin}(I)$ define

$$\mathrm{inv}(J, K) = \#\{(j, k) \in J \times K : j > k\}.$$

Lemma 2.4 (Elementary inversion identities)**.** *For $J, K, L \in \mathrm{Fin}(I)$:*

(a) if $J \cap K = \varnothing$, then $\mathrm{inv}(J \sqcup K, L) = \mathrm{inv}(J, L) + \mathrm{inv}(K, L)$*;*
(b) if $K \cap L = \varnothing$, then $\mathrm{inv}(J, K \sqcup L) = \mathrm{inv}(J, K) + \mathrm{inv}(J, L)$*;*
(c) $\mathrm{inv}(J, K) + \mathrm{inv}(K, J) = |J||K| - |J \cap K|$.

Proof. The first two statements follow by decomposing the relevant sets of ordered pairs. For the third, among the $|J||K|$ pairs in $J \times K$, the diagonal pairs with equal entries are exactly the $|J \cap K|$ pairs for which neither $j > k$ nor $k > j$ holds. Every remaining pair contributes to exactly one of the two inversion counts. □

3. The tensor-sorted Clifford presentation

Definition 3.1 (Tensor-sorted algebra)**.** Let I be a finite or countable ordered set. Let $\mathcal{T}(I)$ be the free unital real associative algebra generated by symbols $\{x_i : i \in I\}$. Let $\mathcal{R}$ be the two-sided ideal generated by

$$x_i^2 + 1, \qquad x_i x_j + x_j x_i \quad (i \neq j).$$

Define

$$\mathcal{A}_{\mathrm{sort}}(I) = \mathcal{T}(I)/\mathcal{R},$$

and write e_i for the image of x_i in the quotient.

Proposition 3.2 (Universal property)**.** *Let B be a unital real associative algebra and let $b_i \in B$ satisfy $b_i^2 = -1$ and $b_i b_j = -b_j b_i$ for $i \neq j$. Then there is a unique unital algebra homomorphism*

$$\Phi : \mathcal{A}_{\mathrm{sort}}(I) \to B$$

with $\Phi(e_i) = b_i$ for all $i \in I$.

Proof. The assignment $x_i \mapsto b_i$ extends uniquely to a homomorphism from the free algebra. The defining relations are sent to zero, hence the map factors through the quotient. Uniqueness follows because the images of the generators generate the quotient. □

4. Sorted normal forms

Definition 4.1 (Reduction rules)**.** In the rewriting-theoretic spirit of Newman and Bergman [9, 3], use the reductions

$$e_i e_i \longrightarrow -1, \qquad e_i e_j \longrightarrow -e_j e_i \quad (i > j).$$

Signs are carried as scalar coefficients.

Lemma 4.2 (Termination)**.** *Every sequence of reductions terminates.*

Proof. For a word $w = e_{i_1} \cdots e_{i_m}$ define

$$\mu(w) = (m, \#\{(r,s) : 1 \le r < s \le m,\ i_r > i_s\})$$

with lexicographic order on $\mathbb{N} \times \mathbb{N}$. Cancelling $e_i e_i$ decreases length, whereas swapping an adjacent inversion preserves length and decreases the inversion number by one. The lexicographic order is well founded. □

Lemma 4.3 (Irreducible words)**.** *A word is irreducible if and only if it is of the form e_J for some $J \in \mathrm{Fin}(I)$.*

Proof. An irreducible word has no adjacent repetition and no adjacent descent; hence its indices are strictly increasing. Conversely, a strictly increasing word has neither type of reducible subword. □

Construction 4.4 (Faithful normal-form model)**.** Let E_I be the real vector space with basis $\{u_J : J \in \mathrm{Fin}(I)\}$. For $i \in I$ define $L_i \in \mathrm{End}_{\mathbb{R}}(E_I)$ by

$$L_i u_J = (-1)^{\#\{j \in J : j < i\} + \mathbf{1}_{i \in J}} u_{J \triangle \{i\}}.$$

Lemma 4.5. *For all $i \neq j$, $L_i^2 = -\mathrm{id}$ and $L_i L_j = -L_j L_i$.*

Proof. Fix $J \in \mathrm{Fin}(I)$ and set $a_i(J) = \#\{j \in J : j < i\} + \mathbf{1}_{i \in J}$. Applying L_i twice toggles the index i twice, and the exponent changes by one modulo 2, giving $L_i^2 = -\mathrm{id}$. For $i \neq j$, both $L_i L_j$ and $L_j L_i$ send u_J to a sign times $u_{J \triangle \{i,j\}}$. Toggling j changes the first exponent for i iff $j < i$, while toggling i changes the first exponent for j iff $i < j$. Exactly one of these alternatives holds, so the two signs differ by -1. □

Theorem 4.6 (Unique sorted normal form)**.** *The elements $\{e_J : J \in \mathrm{Fin}(I)\}$ form an $\mathbb{R}$-basis of $\mathcal{A}_{\mathrm{sort}}(I)$. Every monomial in the generators reduces uniquely to a signed sorted monomial $\pm e_J$. If $|I| = n < \infty$, then $\dim_{\mathbb{R}} \mathcal{A}_{\mathrm{sort}}(I) = 2^n$.*

Proof. The reduction system terminates and every irreducible word is sorted, so the sorted monomials span. By Lemma 4.5 and the universal property, the operators L_i define a representation $\rho : \mathcal{A}_{\mathrm{sort}}(I) \to \mathrm{End}_{\mathbb{R}}(E_I)$. For $J = \{j_1 < \cdots < j_m\}$, the operator $\rho(e_J) = L_{j_1} \cdots L_{j_m}$ sends $u_\varnothing$ to $\pm u_J$. A linear relation among distinct sorted monomials would therefore give a linear relation among distinct basis vectors u_J, which is impossible. Thus the sorted monomials form a basis, and uniqueness follows. □

5. The closed product formula

Theorem 5.1 (Signed-XOR product)**.** *For all $J, K \in \mathrm{Fin}(I)$,*

$$e_J e_K = (-1)^{\mathrm{inv}(J,K)+|J\cap K|} e_{J\triangle K}.$$

Proof. Write $J = \{j_1 < \cdots < j_m\}$ and $K = \{k_1 < \cdots < k_r\}$. In the concatenated word $e_J e_K$, each pair (j, k) with $j > k$ must be interchanged once to sort the word, giving the sign $(-1)^{\mathrm{inv}(J,K)}$. After sorting, each common index contributes one adjacent square $e_i^2 = -1$, giving $(-1)^{|J\cap K|}$. The remaining support is exactly $J\triangle K$. □

Corollary 5.2 (Associative sorted twist)**.** *The function*

$$\theta_{\mathrm{sort}}(J, K) = (-1)^{\mathrm{inv}(J,K)+|J\cap K|}$$

satisfies

$$\theta_{\mathrm{sort}}(J, K)\theta_{\mathrm{sort}}(J\triangle K, L) = \theta_{\mathrm{sort}}(K, L)\theta_{\mathrm{sort}}(J, K\triangle L).$$

Proof. This is associativity of $e_J e_K e_L$ after applying Theorem 5.1 to the two parenthesizations. □

6. The sorted Clifford twist as a twisted group algebra

Theorem 6.1 (Sorted Clifford algebra as a twisted group algebra)**.** *There is a canonical isomorphism*

$$\mathcal{A}_{\mathrm{sort}}(I) \cong \mathbb{R}_{\theta_{\mathrm{sort}}}[\mathrm{Fin}(I)],$$

where $\mathrm{Fin}(I)$ *is regarded as an elementary abelian* 2*-group under symmetric difference and*

$$\theta_{\mathrm{sort}}(J, K) = (-1)^{\mathrm{inv}(J,K)+|J\cap K|}.$$

Proof. Both vector spaces have bases indexed by $\mathrm{Fin}(I)$. The multiplication in $\mathcal{A}_{\mathrm{sort}}(I)$ is

$$e_J e_K = \theta_{\mathrm{sort}}(J, K) e_{J\triangle K}$$

by Theorem 5.1, which is exactly the twisted group algebra product on $\mathrm{Fin}(I)$ with twist θ_{sort}. The basis-preserving linear map is therefore multiplicative and hence an algebra isomorphism. □

Corollary 6.2 (Square sign of the sorted twist)**.** *For $J \in \mathrm{Fin}(I)$,*

$$\theta_{\mathrm{sort}}(J, J) = (-1)^{|J|(|J|+1)/2}.$$

Proof. If $|J| = m$, then $\mathrm{inv}(J, J) = m(m-1)/2$ and $|J \cap J| = m$. Hence

$$\theta_{\mathrm{sort}}(J, J) = (-1)^{m(m-1)/2+m} = (-1)^{m(m+1)/2}.$$

□

7. Finite and countable cases

Theorem 7.1 (Clifford identification)**.** *With the usual generator-and-relation presentation of real Clifford algebras* [10, 8]*, if I is finite with $|I| = n$, then*

$$\mathcal{A}_{\mathrm{sort}}(I) \cong \mathrm{Cl}_{0,n}(\mathbb{R}).$$

If I is countable, then

$$\mathcal{A}_{\mathrm{sort}}(I) \cong \varinjlim_{F \in \mathrm{Fin}(I)} \mathrm{Cl}_{0,|F|}(\mathbb{R}),$$

where the limit is over finite ordered subsets.

Proof. Let V have basis v_i, $i \in I$, and quadratic form $Q(v_i) = -1$ with $v_i \perp v_j$ for $i \neq j$. The Clifford algebra $\mathrm{Cl}(V, Q)$ has generators v_i subject to $v_i^2 = -1$ and $v_i v_j + v_j v_i = 0$ for $i \neq j$, which are exactly the defining relations of $\mathcal{A}_{\mathrm{sort}}(I)$. The universal properties of the two algebras give inverse homomorphisms. The countable case follows because every basis element has finite support and hence lies in a finite subalgebra. □

8. Grading, involutions, and signed permutation representations

The sorted basis carries the standard parity and reversal operations. These structures are not additional assumptions; they are forced by the sorted support calculus.

Definition 8.1 (Grade and parity)**.** For $J \in \mathrm{Fin}(I)$ define the grade of e_J by $|e_J| = |J|$. Let

$$\mathcal{A}_{\bar{0}} = \mathrm{span}_{\mathbb{R}}\{e_J : |J| \equiv 0 \pmod 2\}, \qquad \mathcal{A}_{\bar{1}} = \mathrm{span}_{\mathbb{R}}\{e_J : |J| \equiv 1 \pmod 2\}.$$

Proposition 8.2 ($\mathbb{Z}_2$-grading)**.** *One has*

$$\mathcal{A}_{\bar{a}} \mathcal{A}_{\bar{b}} \subset \mathcal{A}_{\overline{a+b}}.$$

Proof. The support of $e_J e_K$ is $J \triangle K$. Since

$$|J \triangle K| = |J| + |K| - 2|J \cap K|,$$

we have $|J \triangle K| \equiv |J| + |K| \pmod 2$. □

Definition 8.3 (Grade involution, reversion, and Clifford conjugation)**.** Define the grade involution α by

$$\alpha(e_J) = (-1)^{|J|} e_J.$$

For $J = \{j_1 < \cdots < j_m\}$ define reversion by

$$\tau(e_J) = e_{j_m} \cdots e_{j_1} = (-1)^{m(m-1)/2} e_J.$$

The Clifford conjugation is

$$\overline{x} = \tau(\alpha(x)).$$

Thus

$$\overline{e_J} = (-1)^{|J|(|J|+1)/2} e_J.$$

Proposition 8.4 (Involution formulae)**.** *The grade involution is an algebra automorphism of order* 2*, reversion is an anti-automorphism of order* 2*, and Clifford conjugation satisfies*

$$\overline{e_i} = -e_i, \qquad e_J \overline{e_J} = 1$$

for every generator e_i and every sorted basis element e_J.

Proof. The grade involution is multiplicative because $|J \triangle K| \equiv |J| + |K| \pmod 2$. Reversion reverses words, so $\tau(xy) = \tau(y)\tau(x)$ and $\tau^2 = \mathrm{id}$. For $J = \{j_1 < \cdots < j_m\}$, reversing the sorted word requires $m(m-1)/2$ adjacent swaps, giving the stated formula for $\tau(e_J)$ and hence for $\overline{e_J}$. Finally, Theorem 5.1 gives

$$e_J^2 = (-1)^{m(m-1)/2 + m}.$$

Multiplying by the conjugation sign $(-1)^{m(m+1)/2}$ gives $e_J \overline{e_J} = 1$. □

Proposition 8.5 (Local complex slices)**.** *For each $i \in I$, the plane*

$$C_i = \mathrm{span}_{\mathbb{R}}\{1, e_i\}$$

is a subalgebra of $\mathcal{A}_{\mathrm{sort}}(I)$ and is isomorphic to $\mathbb{C}$ by $a + be_i \mapsto a + bi$.

Proof. For $a, b, c, d \in \mathbb{R}$,

$$(a + be_i)(c + de_i) = (ac - bd) + (ad + bc)e_i,$$

because $e_i^2 = -1$. This is the multiplication law in $\mathbb{C}$. □

Proposition 8.6 (Two-generator quaternionic slices)**.** *For distinct $i, j \in I$, the subalgebra generated by e_i and e_j has basis*

$$\{1, e_i, e_j, e_i e_j\}$$

and is isomorphic to the quaternion algebra $\mathbb{H}$.

Proof. Set $\mathbf{i} = e_i$, $\mathbf{j} = e_j$, and $\mathbf{k} = e_i e_j$. Then $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -1$ and $\mathbf{ij} = \mathbf{k}$, $\mathbf{ji} = -\mathbf{k}$. The remaining quaternion relations follow by associativity and the anti-commutation relation. □

Proposition 8.7 (Signed permutation representation)**.** *Assume I is finite. In the sorted basis, left multiplication by every generator e_i is represented by a signed permutation matrix. More precisely,*

$$e_i e_J = (-1)^{\#\{j \in J : j < i\} + \mathbf{1}_{i \in J}} e_{J \triangle \{i\}}.$$

Consequently the matrix of left multiplication by e_i is orthogonal and satisfies $\lambda(e_i)^2 = -I$.

Proof. The displayed formula is the special case $\{i\}$ times J of Theorem 5.1. The map $J \mapsto J \triangle \{i\}$ is a bijection of $\mathrm{Fin}(I)$, so each basis vector is sent to exactly one signed basis vector and every basis vector is reached. Hence the matrix is a signed permutation matrix, and signed permutation matrices are orthogonal. The square relation follows from $e_i^2 = -1$. □

9. Structural constants for twisted XOR algebras

The following is the signed elementary-2-group version of the twisted group algebra viewpoint, close in spirit to quasialgebra descriptions of octonionic multiplication [1].

Definition 9.1 (Twisted XOR algebra)**.** Let V be an elementary abelian 2-group and let $\theta : V \times V \to \{\pm 1\}$ be normalized, meaning $\theta(0, \alpha) = \theta(\alpha, 0) = 1$. Define $\mathbb{R}_\theta[V]$ to be the real vector space with basis $\{e_\alpha : \alpha \in V\}$ and multiplication

$$e_\alpha e_\beta = \theta(\alpha, \beta) e_{\alpha+\beta}.$$

Theorem 9.2 (Associator, commutator, and squares)**.** *For all $\alpha, \beta, \gamma \in V$,*

$$(e_\alpha e_\beta) e_\gamma = \omega_\theta(\alpha, \beta, \gamma) e_\alpha (e_\beta e_\gamma),$$

where

$$\omega_\theta(\alpha, \beta, \gamma) = \frac{\theta(\alpha, \beta)\theta(\alpha + \beta, \gamma)}{\theta(\beta, \gamma)\theta(\alpha, \beta + \gamma)}.$$

Moreover

$$e_\alpha^2 = \theta(\alpha, \alpha) e_0, \qquad e_\alpha e_\beta = \theta(\alpha, \beta)\theta(\beta, \alpha) e_\beta e_\alpha.$$

The algebra is associative if and only if $\omega_\theta \equiv 1$.

Proof. Both parenthesizations of $e_\alpha e_\beta e_\gamma$ are scalar multiples of $e_{\alpha+\beta+\gamma}$. Comparing the scalars gives the displayed associator. The square and commutator formulae follow directly from $\alpha + \alpha = 0$ and $\alpha + \beta = \beta + \alpha$. □

Proposition 9.3 (Gauge equivalence)**.** *If* $\lambda : V \to \{\pm 1\}$ *satisfies* $\lambda(0) = 1$ *and*

$$\theta'(\alpha, \beta) = \frac{\lambda(\alpha)\lambda(\beta)}{\lambda(\alpha + \beta)}\theta(\alpha, \beta),$$

then $\mathbb{R}_\theta[V] \cong \mathbb{R}_{\theta'}[V]$ *via* $e_\alpha \mapsto \lambda(\alpha)e'_\alpha$*. The associator sign is invariant under this gauge transformation.*

Proof. The proposed map preserves products by the defining equation for θ'. It is a vector-space isomorphism because every $\lambda(\alpha)$ is nonzero. The formula for $\omega_{\theta'}$ is obtained by substitution; the λ-factors cancel. □

10. Bit-level realization of Cayley–Dickson multiplication

The Cayley–Dickson construction and its low-dimensional composition-algebra examples are classical [4, 7, 11, 12, 2]. The point of this section is not to redefine those algebras, but to record a bit-level rule for their basis multiplication under the doubling convention fixed below: the support update is XOR, while the sign is supplied by a recursive twist. Different Cayley–Dickson sign conventions lead to gauge-equivalent or relabelled tables, so all sign statements in this section are relative to the displayed multiplication law.

Definition 10.1 (Cayley–Dickson doubling)**.** Let $\mathrm{CD}_0 = \mathbb{R}$ with basis $e_0 = 1$. Given CD_n with conjugation $a \mapsto a^*$, define

$$\mathrm{CD}_{n+1} = \mathrm{CD}_n \oplus \mathrm{CD}_n \ell$$

with multiplication

$$(a + b\ell)(c + d\ell) = (ac - d^*b) + (da + bc^*)\ell.$$

The basis is indexed by $\mathbb{F}_2^{n+1} = \mathbb{F}_2 \times \mathbb{F}_2^n$ through $e_{(0,p)} = e_p$ and $e_{(1,p)} = e_p\ell$.

Definition 10.2 (Recursive Cayley–Dickson sign)**.** Set $\theta_0(0, 0) = 1$. For $p, q \in \mathbb{F}_2^n$, define $\kappa_n(q) = 1$ if $q = 0$ and $\kappa_n(q) = -1$ otherwise, and

set

$$\begin{aligned}
\theta_{n+1}((0,p),(0,q)) &= \theta_n(p,q),\\
\theta_{n+1}((0,p),(1,q)) &= \theta_n(q,p),\\
\theta_{n+1}((1,p),(0,q)) &= \kappa_n(q)\theta_n(p,q),\\
\theta_{n+1}((1,p),(1,q)) &= -\kappa_n(q)\theta_n(q,p).
\end{aligned}$$

Theorem 10.3 (Cayley–Dickson twist realization)**.** *For every $n \geq 0$,*

$$e_p e_q = \theta_n(p,q)e_{p+q}, \qquad p,q \in \mathbb{F}_2^n.$$

Proof. We argue by induction on n. The case $n = 0$ is trivial. Assume the formula holds in CD_n. Write labels in $\mathbb{F}_2^{n+1}$ as $(0,p)$ and $(1,p)$, with $p \in \mathbb{F}_2^n$. The old–old product is

$$e_{(0,p)}e_{(0,q)} = e_p e_q = \theta_n(p,q)e_{p+q} = \theta_{n+1}((0,p),(0,q))e_{(0,p+q)}.$$

For the old–new product, the Cayley–Dickson formula gives

$$e_{(0,p)}e_{(1,q)} = e_p(e_q\ell) = (e_q e_p)\ell = \theta_n(q,p)e_{(1,p+q)}.$$

For the new–old product, using $e_q^* = \kappa_n(q)e_q$ on basis elements,

$$e_{(1,p)}e_{(0,q)} = (e_p\ell)e_q = (e_p e_q^*)\ell = \kappa_n(q)\theta_n(p,q)e_{(1,p+q)}.$$

Finally, for the new–new product,

$$e_{(1,p)}e_{(1,q)} = (e_p\ell)(e_q\ell) = -e_q^* e_p = -\kappa_n(q)\theta_n(q,p)e_{(0,p+q)}.$$

These four cases are precisely the recursive definitions of θ_{n+1}. □

Remark 10.4 (Convention check). For $n = 1$ the recursion gives the complex relation $e_1^2 = -1$. For $n = 2$ it gives a quaternionic table with the usual anti-commutation of the two imaginary generators after the basis labels are read in the convention fixed above. At $n = 3$ the same recursion produces the octonion associator signs used in the worked examples below. Thus the recursive twist is not an additional algebraic structure; it is the structural-constant form of the displayed Cayley–Dickson doubling convention.

11. Synthesis of the table-free multiplication formalism

We now collect the algebraic content in a form that makes the role of the signed-XOR formalism explicit.

Theorem 11.1 (Unified support/sign synthesis)**.** *The following statements hold.*

(i) *The sorted Clifford sector is an associative twisted group algebra over* $\mathrm{Fin}(I)$ *with product*

$$e_J e_K = \theta_{\mathrm{sort}}(J,K) e_{J\triangle K}, \qquad \theta_{\mathrm{sort}}(J,K) = (-1)^{\mathrm{inv}(J,K)+|J\cap K|}.$$

Its associativity is exactly the cocycle identity for θ_{sort}.

(ii) *An arbitrary normalized sign twist on an elementary abelian 2-group produces a signed-XOR algebra*

$$e_\alpha e_\beta = \theta(\alpha,\beta) e_{\alpha+\beta},$$

whose associativity obstruction is the explicitly computable sign ω_θ *of Theorem 9.2.*

(iii) *The Cayley–Dickson basis law is obtained from the same XOR support law by replacing the sorted Clifford cocycle with the recursive twist* θ_n. *Thus Clifford and Cayley–Dickson basis multiplication are treated within a single support/sign calculus, with associativity determined by the twist.*

Proof. Item (i) is Theorem 6.1 together with the cocycle identity in Corollary 5.2. Item (ii) is Theorem 9.2. Item (iii) is the Cayley–Dickson twist realization theorem. The common structure in all three cases is the decomposition of basis multiplication into XOR support and a sign rule. □

12. Constructive structural-constant evaluation

The word "table-free" has a precise algebraic meaning here. It does not mean that products are unspecified. It means that the full multiplication table is reconstructed from a support law and a sign formula.

Theorem 12.1 (Table reconstruction from the sorted rule)**.** *Let* I *be finite with* $|I| = n$. *The complete multiplication table of* $\mathcal{A}_{\mathrm{sort}}(I)$ *is determined by*

$$(J,K) \longmapsto \big(J\triangle K,\ \mathrm{inv}(J,K) + |J\cap K| \bmod 2\big).$$

Thus the 4^n *basis products are generated by one uniform parity rule.*

Proof. There are 2^n sorted basis elements e_J. Theorem 5.1 computes every basis product as

$$e_J e_K = (-1)^{\mathrm{inv}(J,K)+|J\cap K|} e_{J\triangle K}.$$

Thus every table entry is determined by the output support $J\triangle K$ and the displayed parity exponent. □

Theorem 12.2 (Coefficient multiplication formula)**.** *Let*

$$x = \sum_{J \in F} a_J e_J, \qquad y = \sum_{K \in G} b_K e_K,$$

where $F, G \subset \mathrm{Fin}(I)$ *are finite. Then*

$$xy = \sum_{L \in \mathrm{Fin}(I)} c_L e_L,$$

where only finitely many c_L *are nonzero and*

$$c_L = \sum_{J \triangle K = L} a_J b_K (-1)^{\mathrm{inv}(J,K) + |J \cap K|}.$$

Proof. By bilinearity and Theorem 5.1,

$$xy = \sum_{J \in F} \sum_{K \in G} a_J b_K (-1)^{\mathrm{inv}(J,K) + |J \cap K|} e_{J \triangle K}.$$

Collecting the terms with the same support $L = J \triangle K$ gives the formula. □

Theorem 12.3 (Uniqueness of the sorted sign)**.** *Suppose an associative real algebra* B *has a basis* $\{u_J : J \in \mathrm{Fin}(I)\}$ *with* $u_\varnothing = 1$ *and generators* $u_i = u_{\{i\}}$ *satisfying*

$$u_i^2 = -1, \qquad u_i u_j = -u_j u_i \quad (i \neq j),$$

and suppose $u_J = u_{j_1} \cdots u_{j_m}$ *for every* $J = \{j_1 < \cdots < j_m\}$*. Then necessarily*

$$u_J u_K = (-1)^{\mathrm{inv}(J,K) + |J \cap K|} u_{J \triangle K}.$$

Proof. Starting from the concatenated word $u_J u_K$, each out-of-order pair (j, k) with $j > k$ must be swapped once, and each swap contributes one negative sign. After sorting, each common index produces one adjacent square $u_i^2 = -1$. The remaining ordered word is $u_{J \triangle K}$. Since the basis elements are defined by sorted products, no additional sign can appear. □

Corollary 12.4 (Finite verification principle)**.** *To verify any identity among finite sums of sorted basis elements in the plain sector, it suffices to verify equality of the resulting sorted supports and parity exponents after bilinear expansion.*

Proof. The sorted monomials form a basis by Theorem 4.6. Therefore two finite sums are equal if and only if the coefficients of all sorted basis elements agree. □

13. Worked low-dimensional computations

The following examples are outputs of the sign rules, not additional definitions.

Example 13.1 (Quaternion multiplication from supports)**.** Let $I = \{1, 2\}$, put $i = e_1$, $j = e_2$, and $k = e_1e_2$. For

$$x = a + bi + cj + dk, \qquad y = p + qi + rj + sk,$$

Theorem 12.2 gives

$$\begin{aligned} xy = {} & (ap - bq - cr - ds) \\ & + (aq + bp + cs - dr)i \\ & + (ar - bs + cp + dq)j \\ & + (as + br - cq + dp)k. \end{aligned}$$

Thus the usual quaternion product is recovered from supports and signs rather than from a stored table.

Example 13.2 (Plain sorted cancellation)**.** Let $J = \{3, 5, 8\}$ and $K = \{2, 3, 8, 10\}$. Then $J \triangle K = \{2, 5, 10\}$, $|J \cap K| = 2$, and the concatenated word has five cross inversions. Therefore

$$e_3e_5e_8e_2e_3e_8e_{10} = -e_2e_5e_{10}.$$

Example 13.3 (Octonion associator from the Cayley–Dickson twist)**.** At the octonion level, using binary labels $001, 010, 100$, the recursive twist computes

$$(e_{001}e_{010})e_{100} = e_{111}, \qquad e_{001}(e_{010}e_{100}) = -e_{111}.$$

The support is 111 in both parenthesizations. The difference is exactly the associator sign $\omega_{\theta_3}(001, 010, 100)$.

Example 13.4 (Sedenion zero divisor from the recursive twist)**.** At the sedenion level, the recursive Cayley–Dickson twist gives

$$e_1e_4 = e_5, \qquad e_1e_{15} = -e_{14}, \qquad e_{10}e_4 = -e_{14}, \qquad e_{10}e_{15} = e_5.$$

Hence

$$\begin{aligned} (e_1 + e_{10})(e_4 - e_{15}) &= e_1e_4 - e_1e_{15} + e_{10}e_4 - e_{10}e_{15} \\ &= e_5 - (-e_{14}) + (-e_{14}) - e_5 \\ &= 0. \end{aligned}$$

The zero divisor is therefore generated by XOR supports and recursive signs; it is not hidden in a stored multiplication table. This is consistent with the classical boundary imposed by composition-algebra and Hurwitz-type phenomena [5, 12].

14. Bit-level structural constants for the sorted sector

For finite $I = \{1, \dots, n\}$, a support J may be represented by a bit vector $a = (a_1, \dots, a_n) \in \mathbb{F}_2^n$, where $a_i = 1$ if and only if $i \in J$.

Proposition 14.1 (Bit-level structural constants)**.** *Let $a, b \in \mathbb{F}_2^n$ represent supports $J, K \subset \{1, \dots, n\}$. Then*

$$e_a e_b = (-1)^{S(a,b)} e_{a+b},$$

where addition is bitwise XOR and

$$S(a,b) = \sum_{1 \le k < i \le n} a_i b_k + \sum_{i=1}^{n} a_i b_i \pmod 2.$$

Equivalently, if U_n is the strictly lower triangular matrix over $\mathbb{F}_2$ with $(U_n)_{ik} = 1$ for $i > k$, then

$$S(a,b) = a^T (U_n + I_n) b \pmod 2.$$

Proof. The first sum counts cross inversions $\mathrm{inv}(J, K)$ modulo 2, and the second sum counts $|J \cap K|$ modulo 2. This is exactly the exponent in Theorem 5.1. The matrix expression is the same bilinear form written in coordinates. □

15. Numerical algorithms and operation counts

This section records the numerical-algebraic interpretation of the preceding structural formulae. The distinction is important: sparse storage reduces the number of coefficient pairs that must be visited, whereas the signed-XOR rule reduces the cost and storage needed to evaluate the basis product associated with each visited pair.

Assume first that $I = \{1, \dots, n\}$ is finite and that a support is stored as a bit vector $a \in \mathbb{F}_2^n$. On a word-RAM model with word size w, this support may be stored in

$$q = \left\lceil \frac{n}{w} \right\rceil$$

words. The output support of a basis product is computed by wordwise XOR. The sign is the parity of $S(a, b)$ in Proposition 14.1.

Algorithm 15.1 (Sorted basis multiplication)**.** Given two supports $a, b \in \mathbb{F}_2^n$:

(1) compute the output support $c = a + b$ by bitwise XOR;
(2) compute the parity

$$S(a,b) = \sum_{1 \le k < i \le n} a_i b_k + \sum_{i=1}^{n} a_i b_i \pmod 2;$$

(3) return $((-1)^{S(a,b)}, c)$.

Proposition 15.2 (Cost of sorted basis multiplication)**.** *In the bit-vector model above, Algorithm 15.1 evaluates a sorted basis product using $O(q)$ words of storage for each support and $O(q)$ word operations if prefix parities of the second support are supplied or computed in one scan. In particular, for fixed machine-word support size the basis-product cost is constant at the word level.*

Proof. The support update $c = a + b$ is wordwise XOR and therefore uses $O(q)$ word operations. The diagonal term $\sum_i a_i b_i$ is the parity of the bitwise AND of a and b. The inversion term can be evaluated by scanning the words of b while maintaining the parity of the lower-index prefix and accumulating its intersection with the active bits of a. This gives one linear scan over the q words, hence $O(q)$ word operations. If q is bounded by a fixed hardware representation, this is constant at that level of abstraction. □

Now let

$$x = \sum_{J \in F} a_J e_J, \qquad y = \sum_{K \in G} b_K e_K$$

be finite sparse multivectors, with $m_x = |F|$ and $m_y = |G|$ stored terms. The following algorithm is the direct coefficient multiplication induced by Theorem 12.2.

Algorithm 15.3 (Sparse sorted product)**.** Given sparse coefficient maps $F \ni J \mapsto a_J$ and $G \ni K \mapsto b_K$:

(1) initialize an empty coefficient map C;
(2) for each stored support $J \in F$ and each stored support $K \in G$, compute (σ, L) from Algorithm 15.1;
(3) add $\sigma a_J b_K$ to the coefficient C_L;
(4) remove zero coefficients if exact arithmetic or a chosen numerical threshold permits cancellation detection.

Proposition 15.4 (Sparse product cost)**.** *Algorithm 15.3 performs $m_x m_y$ pairwise basis-product evaluations. With the bit-vector cost of Proposition 15.2, its arithmetic and structural-constant cost is*

$$O(m_x m_y\, q)$$

word operations, up to the cost of coefficient arithmetic and dictionary updates. The factor $m_x m_y$ is the usual sparse-bilinear cost; the signed-XOR contribution is the table-free $O(q)$ evaluation of each encountered structural constant.

Proof. The algorithm visits exactly one pair (J, K) for every element of $F \times G$, hence $m_x m_y$ pairs. For each pair, the output support and sign are computed by Algorithm 15.1. Proposition 15.2 therefore gives $O(q)$ word operations per pair, apart from scalar coefficient multiplication and accumulation into the output map. □

Remark 15.5 (What is and is not new in the complexity statement). The reduction from an ambient basis size such as 2^n to a stored support size m is the standard advantage of sparse representation. The present construction should not be credited with that fact. Its numerical role is instead that, once a pair of supports has to be multiplied, the structural constant is generated by XOR and parity rather than by a dimension-specific multiplication table or by a stored list of all 4^n basis products.

Proposition 15.6 (Table memory versus rule memory)**.** *For $|I| = n$, a complete stored basis multiplication table for the sorted sector contains 4^n entries, one for each ordered pair of the 2^n basis elements. The signed-XOR rule stores no such table: it requires only the ordering of the n generators and the uniform parity rule of Proposition 14.1. Consequently the table storage is $O(n)$, or $O(1)$ if the generator order and word-level convention are fixed externally.*

Proof. There are 2^n sorted basis elements, hence 4^n ordered basis products. A stored multiplication table must record the output support and sign for each such ordered pair. By Theorem 12.1 and Proposition 14.1, the same information is generated from XOR support addition and one parity formula. The formula itself is uniform in n; only the ordered list of generators and the support length must be known. □

16. Concluding remarks

The results above give a self-contained signed-XOR structural-constant calculus for basis multiplication in Clifford and Cayley–Dickson type algebras, building on standard Clifford, rewriting, Cayley–Dickson, and quasialgebraic backgrounds [10, 8, 9, 3, 4, 1]. The plain sorted Clifford sector is the associative cocycle case: its normal form, structural constants, involutions, local classical slices, and bit-level multiplication formula are all forced by the quotient presentation. The Cayley–Dickson sector keeps the XOR support law but changes the sign twist; its nonassociativity is measured exactly by the associator sign ω_θ.

The complexity statements should be interpreted conservatively. Sparse storage accounts for the dependence on the number of stored

terms. The specific contribution of the signed-XOR calculus is the table-free evaluation of the basis-product support and sign, together with the exponential memory saving relative to a fully stored multiplication table.

Further applications should be formulated as additional structure on top of this algebraic substrate rather than as part of the present normal-form theorem.

Acknowledgments and declarations

The authors declare that this research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors, and that no known competing financial interest or personal relationship could have appeared to influence the work reported in this paper. No data were used for the research described in this article.

References

1. Helena Albuquerque and Shahn Majid, *Quasialgebra structure of the octonions*, Journal of Algebra **220** (1999), no. 1, 188–224.
2. John C. Baez, *On quaternions and octonions: Their geometry, arithmetic, and symmetry by john h. conway and derek a. smith*, Bulletin of the American Mathematical Society **42** (2005), 229–243.
3. George M. Bergman, *The diamond lemma for ring theory*, Advances in Mathematics **29** (1978), no. 2, 178–218.
4. Leonard E. Dickson, *On quaternions and their generalization and the history of the eight square theorem*, Annals of Mathematics **20** (1919), no. 3, 155–171.
5. Adolf Hurwitz, *Über die komposition der quadratischen formen*, Mathematische Werke. Band II, Birkhäuser, Basel, 1963.
6. Park Junhu, *AXIOM: Algebraic eXtension by Index–Ordered Multiplication*, Master's thesis, Seoul National University, 2026, Master's thesis.
7. T. Y. Lam, *Hamilton's quaternions*, Handbook of Algebra, vol. 3, Elsevier, 2003, pp. 429–454.
8. Pertti Lounesto, *Clifford algebras and spinors*, Clifford Algebras and Their Applications in Mathematical Physics (J. S. R. Chisholm and A. K. Common, eds.), Springer Netherlands, Dordrecht, 1986, pp. 25–37.
9. M. H. A. Newman, *On theories with a combinatorial definition of equivalence*, Annals of Mathematics **43** (1942), no. 2, 223–243.
10. Ian R. Porteous, *Clifford algebras and the classical groups*, Cambridge Studies in Advanced Mathematics, vol. 50, Cambridge University Press, 1995.
11. R. D. Schafer, *An introduction to nonassociative algebras*, Pure and Applied Mathematics, vol. 22, Academic Press, New York, 1966.
12. Tonny A. Springer and Ferdinand D. Veldkamp, *Composition algebras*, Octonions, Jordan Algebras and Exceptional Groups, Springer Monographs in Mathematics, Springer, Berlin, 2000, pp. 1–23.

(P. Junhu) Computational Science & Technology, College of Natural Sciences, Seoul National University, Seoul, Republic of Korea

(P. Junhu) R&D Center, iTrix Co., Ltd., Seoul, Republic of Korea
Email address: nller.park@snu.ac.kr