On certain identities in symmetric compositions ⋆

Pablo Alberca Bjerregaard [a,∗], Cándido Martín González [b]

[a] *University of Málaga, Department of Applied Mathematics. 29013. Málaga. Spain*

[b] *University of Málaga, Department of Algebra, Geometry and Topology, Science Faculty, Apdo. 59, 29080, Málaga. Spain*

Abstract

We give in this paper a computational approach to the verification of identities in symmetric compositions. We also show how to use rules and patterns in noncommutative and nonassociative algebras. It is also presented an implementation for linearization of identities. All the advantages and benefits of working with rules, patterns and functional programming at the same time will be pointed out. We have also included, as a previous section, a practical description of a package with all the necessary instructions for checking identities in para-octonions and pseudo-octononions algebras. This is a natural first step before working with general rings.

Key words: Nonassociative algebras, Symmetric Compositions, Identities, Computational algebra.
PACS: 68W30, 17A75, 17-04, 17-08

1 Introduction

Ottmar Loos recently proved in (5) certain formulas in symmetric compositions. We will confirm these identities (among others) by implementing a set

⋆ Both authors are supported by the Spanish MCYT projects MTM2004-06580-C02-02, MTM2004-08115-C04-04 and by the Junta de Andalucía PAI projects FQM-336 and FQM-1215. This work has been supported by the Spanish Ministry of Education and Science under project 'Ingenio Mathematica (i-Math)' No. CSD2006-00032 (Consolider-Ingenio 2010).

∗ Corresponding author.
Email addresses: pgalberca@uma.es (Pablo Alberca Bjerregaard), candido@apncs.cie.uma.es (Cándido Martín González).

of rules into a computer system. The strategy shown can be a starting point for future studies which need an almost automatic simplification technique.

In the mentioned paper, one of the results is the possibility to construct an structure of a cubic composition from a symmetric one. The whole proof of this result is based on the truthfulness of one identity (see (35)). We are going to establish some of the previous formulas and present a computational way of analyzing identities in algebraic structures and present a proof of the mentioned one. The present work is developed with the software *Mathematica*.

Definition 1 *A symmetric composition is a triple* $(A, \cdot, q)$ *where* $(A, \cdot)$ *is a nonassociative algebra over a field* K *and* q *is a nondegenerate quadratic form* q *which satisfies*

$$q(x \cdot y) = q(x)q(y), \tag{1}$$
$$b(x \cdot y, z) = b(x, y \cdot z), \tag{2}$$

for all $x, y, z \in A$*, and where* b *is the polar form of* q*.*

The second identity, the associativity of b, is equivalent to the fact that $b(x \cdot y, z)$ is invariant under cyclic permutation of x, y and z.

Remark 1 *We will use the following usual notation:* $xy := x \cdot y$, $x \cdot yz := x \cdot (y \cdot z)$, $xy \cdot z := (x \cdot y) \cdot z$, $s := xy$ *and* $t := yx$*.*

2 Checking identities in para-octonions and pseudo-octonions algebras

As a natural previous step, we have verified all the identities which will appear in this paper working with symmetric compositions over fields. This means we must work in the para-octonions and pseudo-octonions algebras (eight-dimensional case). We will show in this section how to use our package **SymmComp.m**, where we have implemented the necessary instructions in order to work with the para-octonions and pseudo-octonions algebras. The source of the package can be seen in a final section, and the can be freely download in http://agt2.cie.uma.es/descargas.htm.

Let $\mathbb{O}_s$ be the set of matrices $\begin{pmatrix} a & u \\ v & b \end{pmatrix}$, where a and b are elements in the field F and $u, v \in F^3$. The set $\mathbb{O}_s$ (Zorn matrices) is a vector space over F under addition and scalar multiplication componentwise. Define the product of two octonions by

$$\begin{pmatrix} a & u \\ v & b \end{pmatrix} \cdot \begin{pmatrix} a' & u' \\ v' & b' \end{pmatrix} = \begin{pmatrix} aa' + \langle u, v' \rangle & au' + b'u - v \times v' \\ a'v + bv' + u \times u' & \langle u, u' \rangle + bb' \end{pmatrix}. \tag{3}$$

The elements of $\mathbb{O}_s$ are usually called Zorn matrices, split octonions or simply octonions.

We define now the conjugate of an octonion as

$$x = \begin{pmatrix} a & u \\ v & b \end{pmatrix} \mapsto \bar{x} := \begin{pmatrix} b & -u \\ -v & a \end{pmatrix}, \tag{4}$$

and, given the order-three automorphism

$$x = \begin{pmatrix} a & (u_1, u_2, u_3) \\ (v_1, v_2, v_3) & b \end{pmatrix} \mapsto \tau(x) := \begin{pmatrix} a & (u_3, u_1, u_2) \\ (v_3, v_1, v_2) & b \end{pmatrix}, \tag{5}$$

we define the new products

$$x \star y = \bar{x} \cdot \bar{y}, \;\; x \circ y = \tau(\bar{x}) \cdot \tau(\tau(\bar{y})) \tag{6}$$

and call $(\mathbb{O}_s, \star)$ the para-octonions algebra and $(\mathbb{O}_s, \circ)$ the pseudo-octonions algebra, which are symmetric compositions.

We first invoke the paraoctonions algebra whose product is **paraoctonions**[,]. For shortness we define the dot product as the product in the paraoctonions algebra:

```
x_ · y_ := paraoctonion[x, y]
```

Next we define four generic element as linear combinations of the basis **B** of the split octonions algebra (recall that the underlying vector spaces of the paraoctonions algebra and that of split octonions algebras agree).

```
x = Sum[λi B[[i]], {i, 8}];
y = Sum[μi B[[i]], {i, 8}];
u = Sum[εi B[[i]], {i, 8}];
v = Sum[τi B[[i]], {i, 8}];
```

Then we start checking identities. We are going to illustrate the method with a previous list of identities (which will appear in Lemma 4) and with the main one of O. Loos in (5) (see 35). Let us see the calculus for the identities of Lemma 4. We shall use the notation $q(x)$ for the Cayley norm:

$$q[x\_] := \mathbf{norm}[x]$$

and the Lie bracket is defined here

$$\mathbf{lie}[x\_, y\_] := \mathbf{Expand}[x \cdot y - y \cdot x]$$

and also the cubic form

$$h[x\_] := b[x \cdot x, x]$$

We only have to execute:

$$\mathbf{Expand}[-b[y, t \cdot s] + b[x \cdot s, y \cdot y] - b[s, x \cdot (y \cdot y)] + b[s, s \cdot y]] == 0$$
True

$$\mathbf{Expand}[b[x, s \cdot t] - b[x \cdot x, y \cdot t] - b[y, t]q[x] + b[x, x \cdot x]q[y]] == 0$$
True

$$\mathbf{Expand}[b[s, s \cdot s] + b[x \cdot s, y \cdot t] - b[x \cdot s, (y \cdot y) \cdot x] - b[s \cdot s, t]] == 0$$
True

and the last one of the mentioned lemma:

$$\mathbf{Expand}[b[s, (x \cdot x) \cdot (y \cdot y)] - 2b[s, s \cdot s] - b[x \cdot s, y \cdot t] + b[t, t \cdot t]+$$
$$3b[x \cdot s, (y \cdot y) \cdot x] + b[x \cdot (y \cdot y), (x \cdot x) \cdot y] - b[x \cdot (y \cdot y), y \cdot (x \cdot x)]-$$
$$2b[x, y]q[x]q[y]] == 0$$
True

Finally, we test the more difficult identity, (35), for which we use the following definitions ((33) and (34)):

$$Q[x\_] := \mathbf{Expand}[x[[1]]^3 - 3x[[1]]q[x[[2]]] + h[x[[2]]]]$$

$$m[x\_, y\_] := \mathbf{Expand}[\{x[[1]]y[[1]] + b[x[[2]], y[[2]]], x[[1]]y[[2]] + y[[1]]x[[2]]+$$
$$\alpha\ x[[2]] \cdot y[[2]] + \beta\ y[[2]] \cdot x[[2]]\}]$$

and the identity is

```
v = {λ, x};  w = {μ, y};  β = 1 − α;
Expand[Q[v]Q[w] − Q[m[v, w]] − (1 − αβ)(3b[x · (x · y), lie[x, y · y]]+
(1 + β)h[lie[x, y]]) − 3(1 − αβ)b[lie[x, y], −λ(x · y) · y + μ(y · x) · x + λμx · y]]
0
```

Once we have checked the identities in the paraoctonions algebra, we switch to the pseudo-octonions algebra. So as before we write:

```
x_ · y_ := pseudoctonion[x, y]
```

We only have to write again all the previous computations to confirm that all the identities are also true for this algebra. In the final section we present the complete package.

3 Previous formulas and linearizations

The aim of the following sections is to verified the previous identity, among others, but working with general rings. We have to work with symbolic identities and manipulate them by implementing rules and using functional programming.

Remark 2 *O. Loos generalized in (5, 3.5) this definition without assuming the nondegeneracy of the quadratic form q and K being an arbitrary ring. Most of the formulas that we are going to present are also true in that context but modulo* $\ker(q)$*. We will assume from now on that q is nonsingular.*

Let us show some rules that will be useful in the next sections. We first define three list of rules which expand the quadratic form, its polar form and the product:

```
expandq = {q[x_, y_] → b[x, y] + q[x] + q[y],  q[x · y] → q[x]q[y]};
expandb = {b[x_ + y_, z_] → b[x, z] + b[y, z],  b[x_, y_ + z_] → b[x, y] + b[x, z],
           b[−x_, y_] → −b[x, y], b[x_,  −y_ · z_] → −b[x, y · z],
           b[x_, −(z_ · t_)] → −b[x, z · t]};
expanddot = {(x_ + y_) · z_ → x · z + y · z,  x_ · (y_ + z_) → x · y + x · z,
             x_ · −y_ → −x · y};
```

Now we present a function which takes out the symbolic scalars of an expression. This command will be crucial in order to simplify and linearize. We thought it would be better to define the function **SOut** for each term that we will face with. If we work in this direction, it will be easy to add new cases as needed. The following definition can be used to extract scalars in terms with b's:

$$
\begin{aligned}
&\textbf{SOut}[\mathrm{b}[\mathrm{x\_},\mathrm{y\_}],\mathrm{scal\_}\textbf{List}] :=\\
&\textbf{Module}[\{\mathrm{scaln} = \textbf{Union}[\mathrm{scal},\textbf{Table}[\zeta_\mathrm{i},\{\mathrm{i},\textbf{Length}[\mathrm{scal}]\}]],\\
&\qquad\quad \mathrm{m} = \textbf{Length}[\mathrm{scal}],\mathrm{res} = \mathrm{b}[\mathrm{x},\mathrm{y}]\},\\
&\textbf{Do}[\mathrm{res} = \mathrm{res}//. - \mathrm{scal}[[\mathrm{j}]] \to \zeta_\mathrm{j},\{\mathrm{j},\mathrm{m}\}];\\
&\quad \textbf{Do}[\textbf{If}[\textbf{Not}[\textbf{FreeQ}[\mathrm{res},\mathrm{scaln}[[\mathrm{j}]]]],\\
&\qquad \mathrm{res} = \mathrm{scaln}[[\mathrm{j}]]^{\textbf{Length}[\textbf{Position}[\mathrm{res},\mathrm{scaln}[[\mathrm{j}]]]]} *\\
&\qquad (\mathrm{res}/.\mathrm{scaln}[[\mathrm{j}]] \to 1)],\{\mathrm{j},\textbf{Length}[\mathrm{scaln}]\}];\\
&\qquad \textbf{Do}[\mathrm{res} = \mathrm{res}//.\zeta_\mathrm{j} \to -\mathrm{scal}[[\mathrm{j}]],\{\mathrm{j},\mathrm{m}\}];\mathrm{res}];
\end{aligned}
$$

This function analyzes if an expression with b's has scalars in the set "scal" by using **FreeQ**. This predefined command gives back true or false in each case.

The next one is for terms which have the product "$\cdot$". In an expression $f[x, y, \ldots]$, the object f is called the *Head*. We have the possibility to recognize the head with a command with the same name, or using "_" as we have already done. We have then

$$
\begin{aligned}
&\textbf{SOut}[\mathrm{x\_}\textbf{CenterDot},\mathrm{scal\_}\textbf{List}] :=\\
&\textbf{Module}[\{\mathrm{scaln} = \textbf{Union}[\mathrm{scal},\textbf{Table}[\zeta_\mathrm{i},\{\mathrm{i},\textbf{Length}[\mathrm{scal}]\}]],\\
&\qquad\quad \mathrm{m} = \textbf{Length}[\mathrm{scal}],\mathrm{res} = \mathrm{x}\},\\
&\textbf{Do}[\mathrm{res} = \mathrm{res}//. - \mathrm{scal}[[\mathrm{j}]] \to \zeta_\mathrm{j},\{\mathrm{j},\mathrm{m}\}];\\
&\quad \textbf{Do}[\textbf{If}[\textbf{Not}[\textbf{FreeQ}[\mathrm{res},\mathrm{scaln}[[\mathrm{j}]]]],\\
&\qquad \mathrm{res} = \mathrm{scaln}[[\mathrm{j}]]^{\textbf{Length}[\textbf{Position}[\mathrm{res},\mathrm{scaln}[[\mathrm{j}]]]]} *\\
&\qquad (\mathrm{res}/.\mathrm{scaln}[[\mathrm{j}]] \to 1)],\{\mathrm{j},\textbf{Length}[\mathrm{scaln}]\}];\\
&\qquad \textbf{Do}[\mathrm{res} = \mathrm{res}//.\zeta_\mathrm{j} \to -\mathrm{scal}[[\mathrm{j}]],\{\mathrm{j},\mathrm{m}\}];\mathrm{res}];
\end{aligned}
$$

Now we have to consider some terms with the quadratic form q. We only need to write:

$$
\begin{aligned}
&\textbf{SOut}[\mathrm{q}[\alpha\_. * \mathrm{x\_}],\mathrm{scal\_}\textbf{List}] :=\\
&\qquad \textbf{Which}[\textbf{MemberQ}[\mathrm{scal},\alpha],\alpha^2\mathrm{q}[\mathrm{x}],\textbf{MemberQ}[\mathrm{scal},\mathrm{x}],\mathrm{x}^2\mathrm{q}[\alpha],\\
&\qquad \textbf{True},\mathrm{q}[\alpha\,\mathrm{x}]];
\end{aligned}
$$

These three definitions are the nucleus of the future simplifications. Although, this command will find terms other than pure b', q's or dot's. We define then the behavior of **SOut** in the rest of the possible cases, which is leaving the term unchanged:

$$\mathbf{SOut}[\mathrm{x\_}\mathbf{Symbol}, \mathrm{scal\_}\mathbf{List}] := \mathrm{x};\ \mathbf{SOut}[\mathrm{x\_}\mathbf{Power}, \mathrm{scal\_}\mathbf{List}] := \mathrm{x};$$
$$\mathbf{SOut}[\mathrm{x\_}\mathbf{Subscript}, \mathrm{scal\_}\mathbf{List}] := \mathrm{x};\ \mathbf{SOut}[\mathrm{x\_?}\mathbf{NumberQ}, \mathrm{scal\_}\mathbf{List}] := \mathrm{x};$$

We use the great possibility that *Mathematica* has in defining values for functions depending on their heads.

The final two definitions insure that **SOut** applies to each term of a product or sum:

$$\mathbf{SOut}[\mathrm{x\_}\mathbf{Times}, \mathrm{scal\_}\mathbf{List}] := \mathbf{Map}[\mathbf{SOut}[\#, \mathrm{scal}]\&, \mathrm{x}];$$
$$\mathbf{SOut}[\mathrm{x\_}\mathbf{Plus}, \mathrm{scal\_}\mathbf{List}] := \mathbf{Map}[\mathbf{SOut}[\#, \mathrm{scal}]\&, \mathrm{x}]$$

3.1 Linearizing $q(x \cdot y) = q(x)q(y)$

We can linearized the multiplication property (1) of the quadratic form. We have to replace x and y by $z_1 + \alpha z_2$ and $z_3 + \beta z_4$, expand, simplify and collect as polynomial in α and β. Each term of this polynomial is, perhaps, a new identity. The identity is:

$$\mathrm{comp} = \mathrm{q}[\mathrm{x} \cdot \mathrm{y}] - \mathrm{q}[\mathrm{x}]\mathrm{q}[\mathrm{y}];$$

and we make

$$\mathrm{comp}//.\ \{\mathrm{x} \to \mathrm{z}_1 + \alpha\,\mathrm{z}_2, \mathrm{y} \to \mathrm{z}_3 + \beta\,\mathrm{z}_4\};$$

We then manipulate this resulting expression by using **expandq**, **expandb** and finally **expanddot**. With **Expand** we eliminate the parenthesis and we apply then

$$\mathbf{SOut}[\%, \{\alpha, \beta\}]$$

were % represents the previous output, and we get

$$\left\| \begin{array}{l} \beta b(z_2 \cdot z_3, z_2 \cdot z_4)\alpha^2 - \beta b(z_3, z_4)q(z_2)\alpha^2 + b(z_1 \cdot z_3, z_2 \cdot z_3)\alpha \\ +\beta b(z_1 \cdot z_3, z_2 \cdot z_4)\alpha + \beta b(z_1 \cdot z_4, z_2 \cdot z_3)\alpha + \beta^2 b(z_1 \cdot z_4, z_2 \cdot z_4)\alpha \\ -\beta b(z_1, z_2)b(z_3, z_4)\alpha - b(z_1, z_2)q(z_3)\alpha - \beta^2 b(z_1, z_2)q(z_4)\alpha \\ +\beta b(z_1 \cdot z_3, z_1 \cdot z_4) - \beta b(z_3, z_4)q(z_1) \end{array} \right\|$$

We only need to write

$$\left\| \textbf{CoefficientList}[\%, \{\alpha, \beta\}] \right\|$$

and the entries of the following list must be zero:

$$\left\| \begin{array}{l} \{\{0, b(z_1 \cdot z_3, z_1 \cdot z_4) - b(z_3, z_4)q(z_1), 0\}, \\ \{b(z_1 \cdot z_3, z_2 \cdot z_3) - b(z_1, z_2)q(z_3), \\ b(z_1 \cdot z_3, z_2 \cdot z_4) + b(z_1 \cdot z_4, z_2 \cdot z_3) - b(z_1, z_2)b(z_3, z_4), \\ b(z_1 \cdot z_4, z_2 \cdot z_4) - b(z_1, z_2)q(z_4)\}, \\ \{0, b(z_2 \cdot z_3, z_2 \cdot z_4) - b(z_3, z_4)q(z_2), 0\}\} \end{array} \right\|$$

and here we have new identities. We have established then:

Lemma 1 *Let $(A, \cdot, q)$ be a symmetric composition algebra. We have then*

$$b(x, y)b(u, v) = b(x \cdot u, y \cdot v) + b(x \cdot v, y \cdot u), \tag{7}$$
$$b(x \cdot y, x \cdot u) = q(x)b(y, u), \tag{8}$$
$$b(x \cdot y, u \cdot y) = q(y)b(x, u), \tag{9}$$

for all $x, y, u, v \in A$.

3.2 Linearizing $(x \cdot y) \cdot x = q(x)y$

As it can be seen in (4, Lemma 34.1), and as we are considering nonsingular quadratic forms, we have from (8) that $b((x \cdot y) \cdot x, u) = b(q(x)y, u)$, hence by nondegeneracy of b, $(x \cdot y) \cdot x = q(x)y$. Finally passing to the opposite algebra we obtain the identity

$$xy \cdot x = x \cdot yx = q(x)y, \text{ that is }, sx = xt = q(x)y, \ \forall x, y \in A. \tag{10}$$

Let us linearize $(x \cdot y) \cdot x = q(x)y$. We will follow the same steps and reasonings we used with the multiplicative identity of q at the previous section. We define first

$$\Big\| \ \mathrm{iden} = (\mathrm{x}\cdot\mathrm{y})\cdot\mathrm{x} - \mathrm{q}[\mathrm{x}]\mathrm{y}; \ \Big\|$$

and make the standard substitution

$$\Big\| \ \mathrm{iden}//.\{\mathrm{x}\to \mathrm{z}_1 + \alpha\,\mathrm{z}_2, \mathrm{y}\to \mathrm{z}_3 + \beta\,\mathrm{z}_4\}; \ \Big\|$$

After applying the rules **expanddot**, **expandq** and taking away the parenthesis, we only have to use the function **SOut**

$$\Big\| \ \mathbf{SOut}[\%, \{\alpha,\beta\}] \ \Big\|$$

to obtain a polynomial in α and β whose coefficients are

$$\left\| \begin{array}{l} \mathbf{CoefficientList}[\%, \{\alpha,\beta\}] \\ \begin{pmatrix} (z_1\cdot z_3)\cdot z_1 - q(z_1)z_3 & (z_1\cdot z_4)\cdot z_1 - q(z_1)z_4 \\ (z_1\cdot z_3)\cdot z_2 + (z_2\cdot z_3)\cdot z_1 - b(z_1,z_2)z_3 & (z_1\cdot z_4)\cdot z_2 + (z_2\cdot z_4)\cdot z_1 - b(z_1,z_2)z_4 \\ (z_2\cdot z_3)\cdot z_2 - q(z_2)z_3 & (z_2\cdot z_4)\cdot z_2 - q(z_2)z_4 \end{pmatrix} \end{array} \right\|$$

We get then a new identity which is

$$xy\cdot z + zy\cdot x = x\cdot yz + z\cdot yx = b(x,z)y. \tag{11}$$

It is now very easy to specialize this identity and change the name of the variables to obtain the following commutator formulas:

Lemma 2 *We have the following identities at any symmetric composition $(A,\cdot,q)$ assuming that q is nonsingular*

$$[x,y^2] = xy\cdot y - y\cdot yx = sy - yt, \tag{12}$$

$$[x^2,y] = x\cdot xy - yx\cdot x = xs - tx, \tag{13}$$

$$[x^2,y^2] = (x^2y)y - y(yx^2) = x(xy^2) - (y^2x)x. \tag{14}$$

3.3 More identities

We need next to establish two lemmas before we present the main formula, the most difficult one. The first one is

Lemma 3 *With the previous notation, and by introducing the abbreviations*

$$a := b(x,y^2)b(y,x^2), \ c := b(x,y)q(x)q(y), \ e := b(x,y)b(s,t), \tag{15}$$

the following identities hold in any symmetric composition algebra

$$b(s,t^2) = b(t,s^2) = a - c, \tag{16}$$
$$b(xs,yt) = e - c, \tag{17}$$
$$h(s) = a - b(sy,xs), \tag{18}$$
$$h(t) = a - b(tx,yt), \tag{19}$$
$$b(x,y)^3 = b(xy^2,yx^2) + c + e, \tag{20}$$

where h is the cubic form $h(x) := b(x, x \cdot x)$.

Proof.

We define first the the cubic form **h**:

$$\mathbf{h}[\mathrm{x\_}] := \mathrm{b}[\mathrm{x}, \mathrm{x} \cdot \mathrm{x}]$$

We have implemented a minimal list of rules to establish the identities of this lemma. Each rule is labeled by the number of the identity used:

```
rules1 = {
    b[x_, y_]b[z_, u_] → b[x · z, y · u] + b[x · u, y · z],  (3)
    b[x_ · y_, x_ · z_] → q[x]b[y, z],  (4)
    b[x_ · y_, z_ · y_] → q[y]b[x, z],  (5)
    (x_ · y_) · x_ → q[x]y,  x_ · (y_ · x_) → q[x]y,  (6)
    b[x_ ∗ q[y_], z_] → q[y]b[x, z],
    b[x_, z_ ∗ q[y_]] → q[y]b[x, z]};
```

where the last two ones are just to extract the scalar $q(y)$ form b. It will be useful to have a rule, using the associativity of b, (2), which associates to one side (to the left):

```
assleft = b[x_, y_ · z_] → b[x · y, z];
```

Now we have all the necessary ingredients. Let us prove the first identity. We will only verify $b(s,t^2) - b(s,y)b(t,x) + b(x,y)q(x)q(y) = 0$ because the other one can be obtained by just interchanching x and y. We apply **rules1** and get

```
b[(x · y), (y · x) · (y · x)] − b[x · y, y]b[y · x, x] + b[x, y]q[x]q[y]//. rules1
```

$$b(x{\cdot}y,(y{\cdot}x){\cdot}(y{\cdot}x)) - b((x{\cdot}y){\cdot}(y{\cdot}x), y{\cdot}x) - b(y, y{\cdot}(y{\cdot}x))q(x) + b(x,y)q(x)q(y)$$

We associate to the left

```
%//. assleft
```

$$b(x,y)q(x)q(y) - b((y \cdot y) \cdot y, x)q(x)$$

apply again **rules1**

```
%//. rules1
```
$b(x,y)q(x)q(y) - b(y,x)q(x)q(y)$

and finally we only have to use that b is symmetric to obtain 0:

```
%/. b[y, x] → b[x, y]
```
0

Let us see $b(xs, yt) - b(x,y)b(s,t) + b(x,y)q(x)q(y) = 0$. This result can be obtained immediately just doing:

```
b[x · (x · y), y · (y · x)] − b[x, y]b[x · y, y · x] + b[x, y]q[x]q[y]//. rules1
```
$b(x,y)q(x)q(y) - b(y,x)q(x)q(y)$

and finally using again that b is symmetric:

```
%/. b[y, x] → b[x, y]
```
0

If we prove $h(xy) - b(x,y^2)b(y,x^2) + b(sy,xs) = 0$, we have also verified the next identity because it is obtained by interchanching x and y. We apply **rules1** to a little previous change made to this formula, using associativity to avoid the squares, just to help the rules to obtain a simplified result:

```
h[x · y] − b[x · y, y]b[x · y, x] + b[(x · y) · y, x · (x · y)]//. rules1
```
$b(x \cdot y, (x \cdot y) \cdot (x \cdot y)) - b((x \cdot y) \cdot (x \cdot y), x \cdot y)$

and finally

```
%//. assleft
```
0

If we previously avoid the cubic power in the equality $b(x,y)^3 - b(xy^2, yx^2) - b(x,y)q(x)q(y) - b(x,y)b(s,t) = 0$, the identity can be obtained just using **rules1**. We have then, using the identity (7),

$$b(x,y)^3 = b(x,y)(b(x,y)b(x,y)) = b(x,y)(b(x^2,y^2) + b(s,t)). \tag{21}$$

The final formula can now be verified automatically

$$\begin{aligned}
&\mathrm{b[x,y]b[x\cdot x,y\cdot y]+b[x,y]b[x\cdot y,y\cdot x]-b[x\cdot(y\cdot y),y\cdot(x\cdot x)]}\\
&\mathrm{-b[x,y]q[x]q[y]-b[x,y]b[x\cdot y,y\cdot x]//.}\ \textbf{rules1}\\
&0
\end{aligned}$$

Q.E.D.

Lemma 4 *We have the following identities in any symmetric composition algebra*

$$-b(y,ts)+b(xs,y^2)-b(s,xy^2)+b(s,sy)=0, \tag{22}$$
$$b(x,st)-b(x^2,yt)-b(y,t)q(x)+b(x,x^2)q(y)=0, \tag{23}$$
$$b(s,s^2)+b(xs,yt)-b(xs,y^2x)-b(s^2,t)=0, \tag{24}$$
$$\begin{aligned}&b(s,x^2y^2)-2b(s,s^2)-b(xs,yt)+b(t,t^2)+3b(xs,y^2x)\\&+b(xy^2,x^2y)-b(xy^2,yx^2)-2b(x,y)q(x)q(y)=0.\end{aligned} \tag{25}$$

Proof.

To establish the first identity we only have to use (12) and (2) to obtain

$$\begin{aligned}
&-b(y,ts)+b(xs,y^2)-b(s,xy^2)+b(s,sy)\\
&=-b(y,ts)+b(s,y^2x)-b(s,xy^2)+b(s,sy)\\
&=-b(y,ts)+b(s,[y^2,x])+b(s,sy)\\
&=-b(y,ts)+b(s,yt-sy)+b(s,sy)=0.
\end{aligned} \tag{26}$$

We can write the second identity as

$$\begin{aligned}
&b(x,st)-b(x^2,yt)-b(y,t)q(x)+b(x,x^2)q(y)\\
&=b(x,st)-b(x^2,yt)-b(t,tx)+b(t,yx^2)=0
\end{aligned} \tag{27}$$

by using the first one interchanging x and y.

The third identity needs Lemma 1, Formula (10) and Lemma 3:

$$\begin{aligned}
&b(s,s^2)+b(xs,yt)-b(xs,y^2x)-b(s^2,t)\\
&=(a-b(sy,xs))+(e-c)-b(xs,y^2x)-(a-c) && \text{(Lemma 3)}\\
&=-b(sy,xs)+b(x,y)b(s,t)-b(xs,y^2x)\\
&=-b(sy,xs)+b(xs,yt)+b(xt,ys)-b(xs,y^2x) && \text{(Lemma 1)}\\
&=-b(sy,xs)+b(xs,yt)+b(q(x)y,q(y)x)-b(xs,y^2x) && \text{(by (10))}
\end{aligned}$$

$$
\begin{aligned}
&= -b(sy, xs) + b(xs, yt) + b(y, x)q(x)q(y) - b(xs, y^2x) \\
&= -b(sy, xs) + b(xs, yt) + b(y^2, s)q(x) - b(xs, y^2x) \qquad \text{(Lemma 1)} \\
&= -b(sy, xs) + b(xs, yt) + b(xy^2, xs) - b(xs, y^2x) \qquad \text{(Lemma 1)} \\
&= -b(xs, sy - yt) + b(xs, [x, y^2]) = 0. \qquad \text{(by (12))}
\end{aligned} \tag{28}
$$

Let us prove the last identity. First of all, we have that

$$
\begin{aligned}
& b(xy^2, x^2y) - b(xy^2, yx^2) - 2b(x, y)q(x)q(y) \\
&= b(xy^2, x^2y) - b(xy^2, yx^2) - 2b(xs, y^2) \qquad \text{(Lemma 1)} \\
&= b(xy^2, [x^2, y]) - 2b(xs, xy^2) = -b(xy^2, tx) - b(xs, xy^2),
\end{aligned} \tag{29}
$$

by (13). Then, by using (12) and Lemma 3, we can establish

$$
\begin{aligned}
& b(s, x^2y^2) - 2b(s, s^2) - b(xs, yt) + b(t, t^2) + 3b(xs, y^2x) \\
&\quad + b(xy^2, x^2y) - b(xy^2, yx^2) - 2b(x, y)q(x)q(y) \\
&= b(s, x^2y^2) - 2b(s, s^2) - b(xs, yt) + b(t, t^2) + 3b(xs, y^2x) \\
&\quad - b(xy^2, tx) - b(xs, xy^2) \\
&= b(s, x^2y^2) - 2b(s, s^2) - b(xs, yt) + b(t, t^2) + 2b(xs, y^2x) \\
&\quad + b(xs, [y^2, x]) - b(xy^2, tx) \\
&= b(s, x^2y^2) - 2b(s, s^2) - b(xs, yt) + b(t, t^2) + 2b(xs, y^2x) \\
&\quad + b(xs, yt - sy) - b(xy^2, tx) \qquad \text{(by (12))} \\
&= b(s, x^2y^2) - 2b(s, s^2) + b(t, t^2) + 2b(xs, y^2x) \\
&\quad - b(xs, sy) - b(xy^2, tx) \\
&= b(s, x^2y^2) - 2h(s) + a - b(tx, yt) + 2b(xs, y^2x) \\
&\quad - b(xy^2, tx) + h(s) - a \qquad \text{(Lemma 3)} \\
&= b(s, x^2y^2) - b(xy^2, tx) - h(s) + 2b(xs, y^2x) - b(tx, yt) = \triangle.
\end{aligned} \tag{30}
$$

This expression could be simplified by using (14), (12) and Lemma 3. First we have

$$
\begin{aligned}
& b(s, x^2y^2) - b(xy^2, tx) = b(s, [x^2, y^2]) + b(s, y^2x^2) - b(xy^2, tx) \\
&= b(s, x(xy^2)) - b(s, (y^2x)x) + b(s, y^2x^2) - b(xy^2, tx) \qquad \text{(by (14))} \\
&= b(sx, xy^2) - b(xs, y^2x) + b(s, y^2x^2) - b(xy^2, tx) \qquad \text{(by (2))} \\
&= b(xt, xy^2) - b(xy^2, tx) - b(xs, y^2x) + b(s, y^2x^2) \qquad \text{(by (10))} \\
&= b(tx, y^2x) - b(xy^2, tx) - b(xs, y^2x) + b(s, y^2x^2) \qquad \text{(Lemma 1)} \\
&= b(tx, [y^2, x]) - b(xs, y^2x) + b(s, y^2x^2) \\
&= b(tx, yt) - b(tx, sy) - b(xs, y^2x) + b(s, y^2x^2). \qquad \text{(by (12))}
\end{aligned} \tag{31}
$$

Using this result we can continue and write

$$\begin{aligned}\triangle &= b(tx, yt) - b(tx, sy) - b(xs, y^2x) + b(s, y^2x^2) \\ &\quad -h(s) + 2b(xs, y^2x) - b(tx, yt) \\ &= b(s, y^2x^2) - h(s) + b(xs, y^2x) - b(tx, sy) \\ &= -c + b(sy, xs) + b(xs, y^2x) - b(tx, sy) \qquad \text{(Lemma 3)} \\ &= -b(x, y)q(x)q(y) + b(xs, y^2x) + b(sy, xs) - b(tx, sy) \\ &= b(xs, [y^2, x]) + b(sy, xs) - b(tx, sy) \qquad \text{(by Lemma 3)} \\ &= b(xs, -sy + yt) + b(sy, xs) - b(tx, sy) = b(xs, yt) - b(sy, tx) \\ &= (e - c) - (e - c) = 0, \end{aligned} \tag{32}$$

by using that not only $b(xs, yt) = e - c$ but also $b(sy, tx) = e - c$, just passing to the opposite algebra.

Q.E.D.

4 Main formula

One of the most important theorems of O. Loos' paper (5) is Theorem 4.1. He defines a cubic form N in a direct sum $K \oplus M$, where M is a symmetric composition (generalized), as

$$N(\lambda \oplus x) := \lambda^3 - 3\lambda q(x) + h(x), \tag{33}$$

and multiplication

$$(\lambda \oplus x) \bullet (\mu \oplus y) := (\lambda\mu + b(x, y)) \oplus (\lambda y + \mu x + \alpha x \cdot y + \beta y \cdot x), \tag{34}$$

with $\alpha + \beta = 1$. The following identity is the last step which proves that $(K \oplus M, N, \bullet)$ is a unital cubic composition:

Proposition 1 *In any symmetric composition we have*

$$\begin{aligned} N(v)N(w) - N(v \bullet w) = \\ (1 - \alpha\beta)\Big(3b(x \cdot xy, [x, y^2]) + (1 + \beta)h([x, y])\Big) \\ +3(1 - \alpha\beta)b([x, y], -\lambda xy \cdot y + \mu yx \cdot x + \lambda\mu xy). \end{aligned} \tag{35}$$

Proof.

The strategy of this proof is to obtain a polynomial in λ and μ from the identity (35). Then we will observe that each coefficient is zero. We will spread all the routines, functions and rules previously defined to this purpose. First of all, the commutator:

$$\textbf{con}[x\_, y\_] := x \cdot y - y \cdot x;$$

We present now **rules2**, an enlarge version of **rules1**, which includes new identities already verified, and $q(x \cdot y) = q(x)q(y)$, among $b(x, x) = 2q(x)$. This set of rules will be the center of the future simplifications. We prefer to define a short list of rules in order to observe to where we can reach and then add new ones if needed. The list is:

```
rules2 = {
    b[x_, y_]b[z_, u_] → b[x · z, y · u] + b[x · u, y · z],  (3)
    b[x_ · y_, x_ · z_] → q[x]b[y, z],  (4)
    b[x_ · y_, z_ · y_] → q[y]b[x, z],  (5)
    (x_ · y_) · x_ → q[x]y,  x_ · (y_ · x_) → q[x]y,  (6)
    b[x_ ∗ q[y_], z_] → q[y]b[x, z],
    b[x_, z_ ∗ q[y_]] → q[y]b[x, z],
    b[x_, y_]^2 → b[x · x, y · y] + b[x · y, y · x],  (3)
    b[x_, y_]^3 → b[x, y]b[x · x, y · y] + b[x, y]b[x · y, y · x],  (b^3 = bb^2)
    b[x_, x_] → 2q[x],
    q[x_ · y_] → q[x]q[y] (1)};
```

It will be useful to present the next set of rules

```
assocb = {
    b[x_ · y_, (x_ · y_) · (y_ · x_)] → b[(x · y) · (x · y), y · x],
    b[x_ · y_, (y_ · x_) · (x_ · y_)] → b[(x · y) · (x · y), y · x],
    b[y_ · x_, (x_ · y_) · (x_ · y_)] → b[(x · y) · (x · y), y · x]};
```

easily obtained from the properties of b. These are the left and right side of the formula:

$$\begin{aligned}\textbf{leftside} &= (\lambda^3 - 3\lambda\mathrm{q[x]} + \mathrm{h[x]})(\mu^3 - 3\mu\mathrm{q[y]} + \mathrm{h[y]})\\ &-((\lambda\,\mu + \mathrm{b[x,y]})^3 - 3(\lambda\mu + \mathrm{b[x,y]})\mathrm{q}[\lambda\mathrm{y} + \mu\mathrm{x} + \alpha\mathrm{x}\cdot\mathrm{y} + \beta\,\mathrm{y}\cdot\mathrm{x}]\\ &+\mathrm{h}[\lambda\,\mathrm{y} + \mu\,\mathrm{x} + \alpha\,\mathrm{x}\cdot\mathrm{y} + \beta\,\mathrm{y}\cdot\mathrm{x}]);\\ \textbf{rightside} &= (1-\alpha\beta)(3\mathrm{b}[\mathrm{x}\cdot(\mathrm{x}\cdot\mathrm{y}), \textbf{con}[\mathrm{x},\mathrm{y}\cdot\mathrm{y}]] + (1+\beta)\mathrm{h}[\textbf{con}[\mathrm{x},\mathrm{y}]])\\ &+3(1-\alpha\beta)\mathrm{b}[\textbf{con}[\mathrm{x},\mathrm{y}], -\lambda(\mathrm{x}\cdot\mathrm{y})\cdot\mathrm{y} + \mu(\mathrm{y}\cdot\mathrm{x})\cdot\mathrm{x} + \lambda\mu\,\mathrm{x}\cdot\mathrm{y}];\end{aligned}$$

and the objective is to prove that

$$\textbf{dif} = \mathrm{Expand}[\textbf{leftside} - \textbf{rightside}];$$

is zero. After applying **expandq**, **expanddot**, **expandq**, **Expand**, **assocb**, **SOut**, **rules2** and $b[y,x] \to b[x,y]$, we obtain (preprint (1)) a polynomial in λ and μ, named **res**. We are now going to prove that this polynomial is zero by analyzing each coefficient. We will need some particular rules but just to move the variables inside b to collect terms. The first list is

$$\textbf{move1} = \{\mathrm{b}[\mathrm{y\_}, \mathrm{x\_}\cdot\mathrm{y\_}] \to \mathrm{b}[\mathrm{x}, \mathrm{y}\cdot\mathrm{y}], \mathrm{b}[\mathrm{y\_}, \mathrm{y\_}\cdot\mathrm{x\_}] \to \mathrm{b}[\mathrm{x}, \mathrm{y}\cdot\mathrm{y}]\};$$

and then the coefficient of $\lambda\mu^2$ is

$$\textbf{Factor}[\textbf{Coefficient}[\textbf{res}, \lambda\mu^2]//.\ \textbf{move1};$$
$$3(\alpha+\beta-1)b(y, x\cdot x)$$

By hypothesis $\alpha + \beta = 1$, and then this term is zero. We also have that

$$\textbf{Factor}[\textbf{Coefficient}[\textbf{res}, \lambda^2\mu]//.\ \textbf{move1};$$
$$3(\alpha+\beta-1)b(x, y\cdot y)$$

and then the coefficient of $\lambda^2\mu$ is also null. The next list of rules:

$$\begin{aligned}\textbf{move2} = \{&\\ &\mathrm{b}[\mathrm{x}, (\mathrm{x}\cdot\mathrm{y})\cdot\mathrm{y}] \to \mathrm{b}[\mathrm{x}\cdot\mathrm{y}, \mathrm{y}\cdot\mathrm{x}], \mathrm{b}[\mathrm{y}\cdot\mathrm{x}, \mathrm{x}\cdot\mathrm{y}] \to \mathrm{b}[\mathrm{x}\cdot\mathrm{y}, \mathrm{y}\cdot\mathrm{x}],\\ &\mathrm{b}[\mathrm{x}, \mathrm{y}\cdot(\mathrm{y}\cdot\mathrm{x})] \to \mathrm{b}[\mathrm{x}\cdot\mathrm{y}, \mathrm{y}\cdot\mathrm{x}], \mathrm{b}[\mathrm{y}, \mathrm{x}\cdot(\mathrm{x}\cdot\mathrm{y})] \to \mathrm{b}[\mathrm{x}\cdot\mathrm{y}, \mathrm{y}\cdot\mathrm{x}],\\ &\mathrm{b}[\mathrm{y}, (\mathrm{y}\cdot\mathrm{x})\cdot\mathrm{x}] \to \mathrm{b}[\mathrm{x}\cdot\mathrm{y}, \mathrm{y}\cdot\mathrm{x}]\};\end{aligned}$$

allows us to confirm that the coefficient of $\lambda\mu$ is also null, assuming of course that $\alpha + \beta = 1$. The command line is:

$$\textbf{Factor}[\textbf{Coefficient}[\textbf{res}, \lambda\mu]//.\ \textbf{move2};$$
$$-3(\alpha+\beta-1)(b(x\cdot y, y\cdot x) - \alpha q(x)q(y) - \beta q(x)q(y) + q(x)q(y))$$

We present the set **move3**:

$$\mathbf{move3} = \{\mathrm{b}[\mathrm{y\_}, \mathrm{y\_} \cdot (\mathrm{y\_} \cdot \mathrm{x\_})] \to \mathrm{b}[\mathrm{y} \cdot \mathrm{x}, \mathrm{y} \cdot \mathrm{y}], \mathrm{b}[\mathrm{y\_}, (\mathrm{x\_} \cdot \mathrm{y\_}) \cdot \mathrm{y\_}] \to \mathrm{b}[\mathrm{x} \cdot \mathrm{y}, \mathrm{y} \cdot \mathrm{y}]\};$$

and as the coefficient of λ^2 is

$$\mathbf{Coefficient}[\mathbf{res}, \lambda^2]/.\ \mu \to 0$$

$$-\beta\, b(y, y\cdot(y\cdot x)) - \alpha\, b(y, (x\cdot y)\cdot y) - 2\alpha\, b(x,y)q(y) - 2\beta\, b(x,y)q(y) + 3b(x,y)q(y)$$

we obtain

$$\%//.\ \mathbf{move3}$$

$$-\alpha\, b(x\cdot y, y\cdot y) - \beta\, b(y\cdot x, y\cdot y) - 2\alpha\, b(x,y)q(y) - 2\beta\, b(x,y)q(y) + 3b(x,y)q(y)$$

We only have to factorize:

$$\mathbf{Factor}[\%//.\ \mathbf{rules1}]$$

$$-3(\alpha+\beta-1)b(x,y)q(y)$$

to confirm that this term is also zero. We can follow a similar reasoning to prove that the coefficient of μ^2 is null, but also, as the mentioned coefficient is:

$$\mathbf{Coefficient}[\mathbf{res}, \mu^2]/.\ \lambda \to 0$$

$$-\alpha\, b(x, x\cdot(x\cdot y)) - \beta\, b(x, (y\cdot x)\cdot x) - 2\alpha\, b(x,y)q(x) - 2\beta\, b(x,y)q(x) + 3b(x,y)q(x)$$

we can affirm that it is null just passing to the opposite algebra and using that the coeficient of λ^2 is null.

The last three coefficients need a little more calculus. As we found a bigger set of terms we needed a bigger set of rules, but just to organize the variables inside b. We define **move4**:

$$\mathbf{move4} = \{
\begin{aligned}
&\mathrm{b[x\cdot y, x] \to b[x, x\cdot y], b[x, y\cdot x] \to b[x, x\cdot y],}\\
&\mathrm{b[y\cdot x, x] \to b[x, x\cdot y], b[x\cdot y, y\cdot (y\cdot x)] \to b[y, (y\cdot x)\cdot (x\cdot y)],}\\
&\mathrm{b[y\cdot x, (x\cdot y)\cdot y] \to b[y, (y\cdot x)\cdot (x\cdot y)],}\\
&\mathrm{b[x\cdot y, y\cdot (y\cdot x)] \to b[y, (y\cdot x)\cdot (x\cdot y)],}\\
&\mathrm{b[y\cdot x, (x\cdot y)\cdot y] \to b[y, (y\cdot x)\cdot (x\cdot y)],}\\
&\mathrm{b[y, (x\cdot y)\cdot (x\cdot y)] \to q[y]b[x, x\cdot y],}\\
&\mathrm{b[y, (y\cdot x)\cdot (y\cdot x)] \to q[y]b[y\cdot x, x],}\\
&\mathrm{b[y, (x\cdot y)\cdot (y\cdot x)] \to q[y]b[x, y\cdot x]\};}
\end{aligned}$$

and after its effect on the coefficient of λ, and a straightfoward substitution of β by $1-\alpha$, we obtain

$$-3\alpha\, b(y,(y{\cdot}x){\cdot}(x{\cdot}y)) + 3\alpha\, b(x{\cdot}(x{\cdot}y), y{\cdot}y) - 3\alpha\, b(y, y{\cdot}y)q(x) + 3\alpha\, b(x, x{\cdot}y)q(y)$$

and this term is null because it is just the identity (22). To confirm this, we only have to write $b(y, y\cdot y)q(x) = b(s, xy^2)$ and $b(x, x\cdot y)q(y) = b(s, sy)$.

We define now the set **move5**:

$$\mathbf{move5} = \{
\begin{aligned}
&\mathrm{b[y, x\cdot y] \to b[y, y\cdot x], b[x\cdot y, y] \to b[y, y\cdot x],}\\
&\mathrm{b[y\cdot x, y] \to b[y, y\cdot x],}\\
&\mathrm{b[x\cdot y, (y\cdot x)\cdot x] \to b[x, (x\cdot y)\cdot (y\cdot x)],}\\
&\mathrm{b[x\cdot (x\cdot y), y\cdot x] \to b[x, (x\cdot y)\cdot (y\cdot x)],}\\
&\mathrm{b[y\cdot x, x\cdot (x\cdot y)] \to b[x, (x\cdot y)\cdot (y\cdot x)],}\\
&\mathrm{b[x, (y\cdot x)\cdot (y\cdot x)] \to q[x]b[y, y\cdot x],}\\
&\mathrm{b[x, (x\cdot y)\cdot (x\cdot y)] \to q[x]b[x\cdot y, y],}\\
&\mathrm{b[x, (y\cdot x)\cdot (x\cdot y)] \to q[x]b[y, x\cdot y]\};}
\end{aligned}$$

and after applying it to the coefficient of μ and making $\beta = 1-\alpha$ we get

$$\begin{aligned}
&3\alpha\, b(x,(x\cdot y)\cdot(y\cdot x)) - 3b(x,(x\cdot y)\cdot(y\cdot x)) - 3\alpha\, b(x\cdot x, y\cdot(y\cdot x))\\
&+3b(x\cdot x, y\cdot(y\cdot x)) - 3\alpha\, b(y, y\cdot x)q(x) + 3b(y, y\cdot x)q(x)\\
&-3b(x, x\cdot x)q(y) + 3\alpha\, b(x\cdot x, x)q(y)
\end{aligned}$$

It is not difficult to see that this identity is $(3\alpha - 1)$ times (23), and then it is zero (we can also use symmetry).

The constant term of the polynomial in λ and μ, after making $\beta = 1 - \alpha$, is also a polynomial but in α. We will apply **assocb** to each coefficient. The first one is

$$\left\| \begin{array}{l} \textbf{Coefficient}[\textbf{pol}, \alpha^2]//.\ \textbf{assocb} \\ -3b(x\cdot y, (x\cdot y)\cdot(x\cdot y)) - 3b(x\cdot(x\cdot y), y\cdot(y\cdot x)) \\ +3b(x\cdot(x\cdot y), (y\cdot y)\cdot x) + 6b((x\cdot y)\cdot(x\cdot y), y\cdot x) \\ -3b((y\cdot x)\cdot(y\cdot x), x\cdot y) \end{array} \right\|$$

As we know that $b(t^2, s) = b(s^2, t)$, this identity is -3 times (24), which is zero. The coefficient of α of this las polynomial is:

$$\left\| \begin{array}{l} \textbf{Coefficient}[\textbf{pol}, \alpha]//.\ \textbf{assocb} \\ 3b(x\cdot y, (x\cdot y)\cdot(x\cdot y)) + 3b(x\cdot(x\cdot y), y\cdot(y\cdot x)) \\ -3b(x\cdot(x\cdot y), (y\cdot y)\cdot x) - 9b((x\cdot y)\cdot(x\cdot y), y\cdot x) + 6b((y\cdot x)\cdot(y\cdot x), x\cdot y) \end{array} \right\|$$

which is also zero because, after taking into account again that $b(t^2, s) = b(s^2, t)$, the identity is 3 times (24). The last coefficient of this polynomial in α is

$$\left\| \begin{array}{l} \textbf{pol}/.\ \alpha \to 0//.\ \textbf{assocb} \\ b(x\cdot y, (x\cdot x)\cdot(y\cdot y)) - 2b(x\cdot y, (x\cdot y)\cdot(x\cdot y)) \\ -b(x\cdot(x\cdot y), y\cdot(y\cdot x)) + 3b(x\cdot(x\cdot y), (y\cdot y)\cdot x) \\ -b(x\cdot(y\cdot y), y\cdot(x\cdot x)) + b(x\cdot(y\cdot y), (x\cdot x)\cdot y) \\ +b(y\cdot x, (y\cdot x)\cdot(y\cdot x)) + 6b((x\cdot y)\cdot(x\cdot y), y\cdot x) \\ -6b((y\cdot x)\cdot(y\cdot x), x\cdot y) - 2b(x, y)q(x)q(y) \end{array} \right\|$$

which, after using again that $b(t^2, s) = b(s^2, t)$, is (25), and then it is also null, and the identity (35) holds.

Q.E.D.

5 SymmComp.m package

(*:Name: Algebra‘SymmComp‘ *)

(*:Title: Symmetric Composition algebras *)

(*:Authors: Pablo Alberca Bjerregaard and Cándido Martín González *)

(*:Keywords: spit octonions, Zorn matrices, Para-octonions, Pseudo-octonions *)

(*:Summary:

Package for computations with some symmetric compositions algebras*)

BeginPackage[”Algebra‘SymmComp‘”]

p::usage = ”Gives the product p[x,y] of the octonions x and y.”

rec::usage= ”Writes an octonion as a formal linear combinations of elements in the standard basis.”

B::usage=”Standar basis of split Octonions.”

cero::usage=”cero Zorn matrix.”

uno::usage=”unit Zorn matrix.”

Cayley::usage=”Conjugate of octonion.”

trace::usage=”trace of octonion.”

norm::usage=”norm of octonion.”

avec::usage=”octonion to vector.”

aoct::usage=”vector to octonion.”

LinearMap::usage=”Linear map with given matrix”

e::usage=”Idempotents (needs subscript = 1,2).”

u::usage=”Elements in the (1,0) Peirce space (relative to e_1) of Zorn’s algebra (needs subscript = 1,3).”

v::usage=”Elements in the (0,1) Peirce space (relative to e_1) of Zorn’s algebra (needs subscript = 1,3).”

recT::usage=”TeX Form of rec.”

paraoctonion::usage=”Gives the product paraoctonion[x,y] of elements x,y in the para-octonions algebra asociated to Zorn matrices algebra”

pseudoctonion::usage=”Gives the product pseudoctonion[x,y] of elements x,y in the pseudo-octonions algebra asociated to Zorn matrices algebra”

b::usage= ”Polar form of the octonions norm”

auto::usage=”Order-three automorphism of Zorn matrices”

```
Begin["Algebra`Zorn`Private`"]
p[x_,y_]:={{x[[1,1]]y[[1,1]]+x[[1,2]].y[[2,1]],x[[1,1]]y[[1,2]]+y[[2,2]]x[[1,2]]-Cross[x[[2,1]],
y[[2,1]]]},{y[[1,1]]x[[2,1]]+x[[2,2]]y[[2,1]]+Cross[x[[1,2]],y[[1,2]]],
x[[2,1]].y[[1,2]]+x[[2,2]]y[[2,2]]}};
rec[x_]:=Sum[x[[i,i]] Subscript["e",i],{i,2}]+Sum[x[[1,2]][[i]] Subscript["u",i],{i,3}]+
Sum[x[[2,1]][[i]] Subscript["v",i],{i,3}]
recT[x_]:=TeXForm[rec[x]]
B={{{1,{0,0,0}},{{0,0,0},0}},
{{0,{0,0,0}},{{0,0,0},1}},
{{0,{1,0,0}},{{0,0,0},0}},
{{0,{0,1,0}},{{0,0,0},0}},
{{0,{0,0,1}},{{0,0,0},0}},
{{0,{0,0,0}},{{1,0,0},0}},
{{0,{0,0,0}},{{0,1,0},0}},
{{0,{0,0,0}},{{0,0,1},0}}}
Do[Subscript[e,i]=B[[i]],{i,2}]
Do[Subscript[u,i]=B[[2+i]],{i,3}]
Do[Subscript[v,i]=B[[5+i]],{i,3}]
cero={{0,{0,0,0}},{{0,0,0},0}}
uno={{1,{0,0,0}},{{0,0,0},1}}
Cayley[x_]:={{x[[2,2]],-x[[1,2]]},{-x[[2,1]],x[[1,1]]}}
trace[x_]:=Expand[x+Cayley[x]][[1,1]]
norm[x_]:=Expand[p[x,Cayley[x]]][[1,1]]
avec[x_]:={x[[1,1]],x[[2,2]],x[[1,2]][[1]],x[[1,2]][[2]],x[[1,2]][[3]],x[[2,1]][[1]],
x[[2,1]][[2]],x[[2,1]][[3]]}
aoct[x_]:={{x[[1]],{x[[3]],x[[4]],x[[5]]}},{{x[[6]],x[[7]],x[[8]]},x[[2]]}}
LinearMap[M_,x_]:=aoct[M.avec[x]]//Expand
paraoctonion[x_,y_]:=p[Cayley[x],Cayley[y]]
b[x_,y_]:=Expand[norm[x+y]-norm[x]-norm[y]]
auto[x_]:={
{x[[1,1]],{x[[1,2]][[3]],x[[1,2]][[1]],x[[1,2]][[2]]}},
{{x[[2,1]][[3]],x[[2,1]][[1]],x[[2,1]][[2]]},x[[2,2]]}}
```

```
pseudoctonion[x_,y_]:=p[auto[Cayley[x]],auto[auto[Cayley[y]]]]
End[] (* Algebra`SymmComp`Private` *)
EndPackage[] (* Algebra`SymmComp` *)
```

6 Acknowledgments

We are very grateful to professor Ottmar Loos for sharing with us his recent investigation and for motivating us with the possiblity of getting a more direct prove of certain identities by using a computer system.

References

[1] P. Alberca Bjerregaard and C. Martín González. *Computational techniques for proving identities in symmetric compositions.* http://arxiv.org/. 2007.
[2] P. Alberca Bjerregaard and C. Martín González. *Paraoctonions and pseudo-octonions package. Checking identities.* http://agt2.cie.uma.es/descargas.htm/. 2007.
[3] A. Elduque. *Symmetric Composition Algebras.* Journal of Algebra, 196. 283-300. 1997.
[4] M. -A. Knus, A. Merkurjev, M. Rost and J. -P. Tignol. *The book of involutions.* American Mathematical Society Colloquium Publications. Vol. 44. American Mathematical Society. Providence, RI. 1998.
[5] O. Loos. *Cubic and symmetric compositions over rings.* http://www.uibk.ac.at/mathematik/loos/jordan/index.html. Preprint. 2006.