Geometric Algebra Determinant Preserving Transforms

Kurt Nalty

April 17, 2018

Abstract

By analogy to matrix implementations, we can define a determinant for generic multivectors in geometric algebra. This determinant measures the internal volume of the multivector, and is akin to a magnitude. Like matrices, the determinant of a product of multivectors is the product of their determinants. Like matrices, the determinant must be non-zero for an inverse to exist. Just as rotations leave a magnitude invariant, so do a large class of transforms leave determinants invariant. This note documents transforms akin to complements, duals, simple blade products and simple sandwich products in (3,0), (4,0), (3,1) and (4,1) geometric algebras.

Euclidean Space (3,0)

Euclidean space geometric algebra has generic multivectors consisting of a scalar q, three anticommuting space basis vectors e_x, e_y, e_z which square to one, three bivectors $e_{xy} = e_x e_y$, $e_{xz} = e_x e_z$, $e_{yz} = e_y e_z$ which square to negative one, and a trivector $e_{xyz} = e_x e_y e_z$ which squares to negative one, commutes with all geometric algebra elements, and mimics $i = \sqrt{-1}$.

The order sensitive multiplication table for these blades, in text format, with pre-factors on the left and post-factors on the top, is

*	q	x	y	z	xy	xz	yz	xyz
q	q	x	y	z	xy	xz	yz	xyz
x	x	q	xy	xz	y	z	xyz	yz
y	y	-xy	q	yz	-x	-xyz	z	-xz
z	z	-xz	-yz	q	xyz	-x	-y	xy
xy	xy	-y	x	xyz	-q	-yz	xz	-z
xz	xz	-z	-xyz	x	yz	-q	-xy	y
yz	yz	xyz	-z	y	-xz	xy	-q	-x
xyz	xyz	yz	-xz	xy	-z	y	-x	-q

These eight elements (blades) are the building blocks of generic multivectors.

$$MV = a + be_x + ce_y + de_z + ee_{xy} + fe_{xz} + ge_{yz} + he_{xyz}$$

In my programs, I define a data strucure GA3E with fields `q, x,y,z, xy,xz,yz, xyz`. The multivector above could be defined as

`MV = GA3E(a, b,c,d, e,f,g, h)`

The individual fields can be accessed as

```
a = MV.q ;
b = MV.x ;
c = MV.y ;
d = MV.z ;
e = MV.xy ;
f = MV.xz ;
g = MV.yz ;
h = MV.xyz ;
```

Determinant

The determinant is initially defined by translating the Clifford algebra to a matrix based implementation, then taking the determinant of the resulting matrix. Using complex 2x2 matrices (Pauli matrices) results in a complex value for the determinant, and does not extend to higher dimensions. Using 4x4 matrices (Dirac matrices) results in a real expression for the determinant and can be extended to four and five dimensions. I use 4x4 matrices for this work.

The matrix approach results in a complicated, slow expression. This expression is then used to validate determinant preserving properties of candidate conjugations, such as the Clifford conjugation. Products using these conjugates lead to much more efficient formulas for the determinant. Code illustrating this approach is shown below.

```
A = GA3E(a, b,c,d, e,f,g, h)
B = CliffordConjugation(MV); // GA3E(a, -b,-c,-d, -e,-f,-g, h)
C = Product(B,A);
det = (C.q*C.q + C.xyz*C.xyz);
```

Complements

The Clifford conjugation, shown above, is one of many conjugation patterns which preserve the determinant. These conjugations are similar to complex conjugation, where we change the sign of multivector components according to some pattern. Given just eight components for a 3D multivector, we have only 256 candidate transformations to examine, and quickly find 32 determinant preserving complements.

By inspection, seeing the binary count sequence, we identify the signs of a, b, c, d and e as the independent variables, with the signs of f, g and h being given by

$$\begin{aligned}\mathrm{Sign}(xz) &= \mathrm{Sign}(xy) * \mathrm{Sign}(z) * \mathrm{Sign}(y)\\ \mathrm{Sign}(yz) &= \mathrm{Sign}(xy) * \mathrm{Sign}(z) * \mathrm{Sign}(x)\\ \mathrm{Sign}(xyz) &= \mathrm{Sign}(xy) * \mathrm{Sign}(z) * \mathrm{Sign}(q)\end{aligned}$$

The routine `GA3E Comp(GA3E u, int i)` returns a multivector with the components' signs toggled via selected i, in accordance with the algorithm above.

Blade Substitutions

In three dimensions, I have only three blades which square to one ant anticommute. This leads to six permutations for implementations of x, y and z, and higher order blade products.

```
   (q,   x, y, z,  xy,xz,yz, xyz)

 0 (+a, +b,+c,+d, +e,+f,+g, +h)
 1 (+a, +b,+c,+d, -e,-f,-g, -h)
 2 (+a, +b,+c,-d, +e,-f,-g, -h)
 3 (+a, +b,+c,-d, -e,+f,+g, +h)
 4 (+a, +b,-c,+d, +e,-f,+g, +h)
 5 (+a, +b,-c,+d, -e,+f,-g, -h)
 6 (+a, +b,-c,-d, +e,+f,-g, -h)
 7 (+a, +b,-c,-d, -e,-f,+g, +h)
 8 (+a, -b,+c,+d, +e,+f,-g, +h)
 9 (+a, -b,+c,+d, -e,-f,+g, -h)
10 (+a, -b,+c,-d, +e,-f,+g, -h)
11 (+a, -b,+c,-d, -e,+f,-g, +h)
12 (+a, -b,-c,+d, +e,-f,-g, +h)
13 (+a, -b,-c,+d, -e,+f,+g, -h)
14 (+a, -b,-c,-d, +e,+f,+g, -h)
15 (+a, -b,-c,-d, -e,-f,-g, +h)
16 (-a, +b,+c,+d, +e,+f,+g, -h)
17 (-a, +b,+c,+d, -e,-f,-g, +h)
18 (-a, +b,+c,-d, +e,-f,-g, +h)
19 (-a, +b,+c,-d, -e,+f,+g, -h)
20 (-a, +b,-c,+d, +e,-f,+g, -h)
21 (-a, +b,-c,+d, -e,+f,-g, +h)
22 (-a, +b,-c,-d, +e,+f,-g, +h)
23 (-a, +b,-c,-d, -e,-f,+g, -h)
24 (-a, -b,+c,+d, +e,+f,-g, -h)
25 (-a, -b,+c,+d, -e,-f,+g, +h)
26 (-a, -b,+c,-d, +e,-f,+g, +h)
27 (-a, -b,+c,-d, -e,+f,-g, -h)
28 (-a, -b,-c,+d, +e,-f,-g, -h)
29 (-a, -b,-c,+d, -e,+f,+g, +h)
30 (-a, -b,-c,-d, +e,+f,+g, +h)
31 (-a, -b,-c,-d, -e,-f,-g, -h)
```

Table 1: Three Dimensional Conjugation Table

```
GA3E Magic(GA3E u, int i)

if(i == 0) MV = GA3E( a, +b,+c,+d, +e,+f,+g, +h) ;
if(i == 1) MV = GA3E( a, +b,+d,+c, +f,+e,-g, -h) ;
if(i == 2) MV = GA3E( a, +c,+b,+d, -e,+g,+f, -h) ;
if(i == 3) MV = GA3E( a, +c,+d,+b, +g,-e,-f, +h) ;
if(i == 4) MV = GA3E( a, +d,+b,+c, -f,-g,+e, +h) ;
if(i == 5) MV = GA3E( a, +d,+c,+b, -g,-f,-e, -h) ;
```

Blade Products

The determinant of individual blades is one. Consequently, pre-multiplication, post-multiplication and sandwich multiplication by any combination of blades preserves the determinant. I find by enumeration that the combination of pre-multiplication by a blade and generalized conjugation also covers the post-multiplication and sandwich multiplication products. These products I call the generalized duality transforms, whereas pure duality involves only multiplication by the pseudo-scalar.

Combining these transforms leads to 1536 distinct simple determinant preserving transforms.

```
//  list composite Magic, Blade and Comp transforms

r = GA3E(a, b,c,d, e,f,g, h);  // 1536 transforms
for (i=0;i<6;i++) { // magic
    s = Magic(r,i);
    for (j=0;j<8;j++) { // blade prefactor
        t = Blade[j]*s;
        for (k=0; k<32; k++) { // Comps
            u = Comp(t,k);
            PrintMV(u);
            printf(" \n");
        }
    }
}
```

Euclidean Space (4,0)

Four dimensional Euclidean space has 16 blades. The scalar blade e_q squares to one, and commutes wth all blades. The vector blades e_x, e_y, e_z, and e_w

square to one and mutually anti-commute. The bivector blades $e_{xy}, e_{xz}, e_{yz}, e_{xw}, e_{yw}$, and e_{zw} square to negative one, as do the trivector blades $e_{xyz}, e_{xyw}, e_{xzw}$, and e_{yzw}. The quadvector blade e_{xyzw} squares to one.

The order sensitive multiplication table for these blades, in text format, with pre-factors on the left and post-factors on the top, is shown in sideways Table 2.

Complements

Direct evaluation and enumeration discovers 64 sign variation complements in four dimensional Euclidean space. Examination of the pattern reveals e_q, e_x, e_y, e_z, e_w, and e_{xy} are the independent variables. The signs for the remaining terms follow simple logic.

```
if((32 & i) == 0) Sign[5] = +1; else Sign[5] = -1;  // q
if((16 & i) == 0) Sign[4] = +1; else Sign[4] = -1;  // x
if(( 8 & i) == 0) Sign[3] = +1; else Sign[3] = -1;  // y
if(( 4 & i) == 0) Sign[2] = +1; else Sign[2] = -1;  // z
if(( 2 & i) == 0) Sign[1] = +1; else Sign[1] = -1;  // w
if(( 1 & i) == 0) Sign[0] = +1; else Sign[0] = -1;  // xy

Sign[6]  = Sign[0]*Sign[2]*Sign[3]; // xz
Sign[7]  = Sign[0]*Sign[1]*Sign[3]; // xw
Sign[8]  = Sign[0]*Sign[2]*Sign[4]; // yz
Sign[9]  = Sign[0]*Sign[1]*Sign[4]; // yw
Sign[10] = Sign[0]*Sign[1]*Sign[2]*Sign[3]*Sign[4]; // zw

Sign[11] = Sign[0]*Sign[2]*Sign[5]; // xyz
Sign[12] = Sign[0]*Sign[1]*Sign[5]; // xyw
Sign[13] = Sign[0]*Sign[1]*Sign[2]*Sign[3]*Sign[5]; // xzw
Sign[14] = Sign[0]*Sign[1]*Sign[2]*Sign[4]*Sign[5]; // yzw

Sign[15] = Sign[1]*Sign[2]*Sign[3]*Sign[4]*Sign[5]; // xyzw
```

Blade Substitutions

In GA4E (Geometric Algebra, 4D, Euclidean), we have six blades which square to one, being e_q, e_x, e_y, e_z, e_w, and e_{xyzw}. The latter five blades can play the role of fundamental vectors, leading to $5! = 120$ variations, which are verified to preserve the determinant.

*	q	x	y	z	w	xy	xz	yz	xw	yw	zw	xyz	xyw	xzw	yzw	xyzw
q	q	x	y	z	w	xy	xz	yz	xw	yw	zw	xyz	xyw	xzw	yzw	xyzw
x	x	q	xy	xz	xw	y	z	xyz	w	xyw	xzw	yz	yw	zw	xyzw	yzw
y	y	-xy	q	yz	yw	-x	-xyz	z	-xyw	w	yzw	-xz	-xw	-xyzw	zw	-xzw
z	z	-xz	-yz	q	zw	xyz	-x	-y	-xzw	-yzw	w	xy	xyzw	-xw	-yw	xyw
w	w	-xw	-yw	-zw	q	xyw	xzw	yzw	-x	-y	-z	-xyzw	xy	xz	yz	-xyz
xy	xy	-y	x	xyz	xyw	-q	-yz	xz	-yw	xw	xyzw	-z	-w	-yzw	xzw	-zw
xz	xz	-z	-xyz	x	xzw	yz	-q	-xy	-zw	-xyzw	xw	y	yzw	-w	-xyw	yw
yz	yz	xyz	-z	y	yzw	-xz	xy	-q	xyzw	-zw	yw	-x	-xzw	xyw	-w	-xw
xw	xw	-w	-xyw	-xzw	x	yw	zw	xyzw	-q	-xy	-xz	-yzw	y	z	xyz	-yz
yw	yw	xyw	-w	-yzw	y	-xw	-xyzw	zw	xy	-q	-yz	xzw	-x	-xyz	z	xz
zw	zw	xzw	yzw	-w	z	xyzw	-xw	-yw	xz	yz	-q	-xyw	xyz	-x	-y	-xy
xyz	xyz	yz	-xz	xy	xyzw	-z	y	-x	yzw	-xzw	xyw	-q	-zw	yw	-xw	-w
xyw	xyw	yw	-xw	-xyzw	xy	-w	-yzw	xzw	y	-x	-xyz	zw	-q	-yz	xz	z
xzw	xzw	zw	xyzw	-xw	xz	yzw	-w	-xyw	z	xyz	-x	-yw	yz	-q	-xy	-y
yzw	yzw	-xyzw	zw	-yw	yz	-xzw	xyw	-w	-xyz	z	-y	xw	-xz	xy	-q	x
xyzw	xyzw	-yzw	xzw	-xyw	xyz	-zw	yw	-xw	-yz	xz	-xy	w	-z	y	-x	q

Table 2: Four Dimensional Euclidean Geometric Algebra Multiplication Table

Blade Products

As with GA3E (Geometric Algebra, 3D, Euclidean), the combination of blade post-multiplication and sign conjugation also includes the pre-multiplication and sandwich product terms. Compositing all these transforms, we have 120K (122880) distinct discrete determinant preserving transforms.

```
//  list composite Magic, Blade and Comp transforms

MV1 = GA4E(a, b,c,d,e, f,g,h,j,k,l, m,n,p,r, s) ;
    for (ii=0;ii<120;ii++) { // 120K = 122880 variations
        MV2 = Magic(MV1,ii);
        for (jj=0;jj<16;jj++) { // blade prefactor
            MV3 = Blade[jj]*MV2;
            for (kk=0; kk<64; kk++) { // Comps
                MV4 = Comp(MV3,kk);
                PrintMV(MV4);
                printf(" \n");
            }
        }
    }
```

Minkowski Spacetime (3,1)

Four dimensional Minkowski space-time has 16 blades. The (3,1) metric designation indicates that three basis square to one, while one basis vector squares to negative one. The scalar blade e_q squares to one, and commutes wth all blades. The vector blades e_x, e_y, and e_z, square to one, while e_t squares to negative one. These four vector basis blades mutually anticommute. The bivector blades e_{xy}, e_{xz}, and e_{yz} square to negative one, as does the trivector blade e_{xyz} and the quadvector e_{xyzt}. However, due to the t component, bivector blades e_{xt}, e_{yt}, e_{zt} and trivectors blades e_{xyt}, e_{xzt}, e_{yzt} square to positive one.

The order sensitive multiplication table for these blades, in text format, with pre-factors on the left and post-factors on the top, is shown in sideways Table 3.

Complements

Direct evaluation and enumeration discovers 64 sign variation complements in four dimensional Euclidean space. Examination of the pattern reveals

Geometric product, Minkowski spacetime

*	q	x	y	z	t	xy	xz	yz	xt	yt	zt	xyz	xyt	xzt	yzt	xyzt
q	q	x	y	z	t	xy	xz	yz	xt	yt	zt	xyz	xyt	xzt	yzt	xyzt
x	x	q	xy	xz	xt	y	z	xyz	t	xyt	xzt	yz	yt	zt	xyzt	yzt
y	y	-xy	q	yz	yt	-x	-xyz	z	-xyt	t	yzt	-xz	-xt	-xyzt	zt	-xzt
z	z	-xz	-yz	q	zt	xyz	-x	-y	-xzt	-yzt	t	xy	xyzt	-xt	-yt	xyt
t	t	-xt	-yt	-zt	-q	xyt	xzt	yzt	x	y	z	-xyzt	-xy	-xz	-yz	xyz
xy	xy	-y	x	xyz	xyt	-q	-yz	xz	-yt	xt	xyzt	-z	-t	-yzt	xzt	-zt
xz	xz	-z	-xyz	x	xzt	yz	-q	-xy	-zt	-xyzt	xt	y	yzt	-t	-xyt	yt
yz	yz	xyz	-z	y	yzt	-xz	xy	-q	xyzt	-zt	yt	-x	-xzt	xyt	-t	-xt
xt	xt	-t	-xyt	-xzt	-x	yt	zt	xyzt	q	xy	xz	-yzt	-y	-z	-xyz	yz
yt	yt	xyt	-t	-yzt	-y	-xt	-xyzt	zt	-xy	q	yz	xzt	x	xyz	-z	-xz
zt	zt	xzt	yzt	-t	-z	xyzt	-xt	-yt	-xz	-yz	q	-xyt	-xyz	x	y	xy
xyz	xyz	yz	-xz	xy	xyzt	-z	y	-x	yzt	-xzt	xyt	-q	-zt	yt	-xt	-t
xyt	xyt	yt	-xt	-xyzt	-xy	-t	-yzt	xzt	-y	x	xyz	zt	q	yz	-xz	-z
xzt	xzt	zt	xyzt	-xt	-xz	yzt	-t	-xyt	-z	-xyz	x	-yt	-yz	q	xy	y
yzt	yzt	-xyzt	zt	-yt	-yz	-xzt	xyt	-t	xyz	-z	y	xt	xz	-xy	q	-x
xyzt	xyzt	-yzt	xzt	-xyt	-xyz	-zt	yt	-xt	yz	-xz	xy	t	z	-y	x	-q

Table 3: Four Dimensional Euclidean Geometric Algebra Multiplication Table

e_q, e_x, e_y, e_z, e_t, and e_{xy} are the independent variables. The signs for the remaining terms follow simple logic, matching that of GA4E. I make the observation that the signs display a mirror symmetry of sorts. Interchanging Sign[0] with Sign[15], Sign[1] with Sign[14], and so on results in another valid complement in this list. This implies that an alternative set of independent blades could be made using the duals of the set above. (This alternative set is e_{xyzt}, e_{yzt}, e_{xzt}, e_{xyt}, e_{xyz}, and e_{zt})

```
if((Mask & i) == 0) Sign[5] = +1; else Sign[5] = -1;  Mask >>= 1;  // q
if((Mask & i) == 0) Sign[4] = +1; else Sign[4] = -1;  Mask >>= 1;  // x
if((Mask & i) == 0) Sign[3] = +1; else Sign[3] = -1;  Mask >>= 1;  // y
if((Mask & i) == 0) Sign[2] = +1; else Sign[2] = -1;  Mask >>= 1;  // z
if((Mask & i) == 0) Sign[1] = +1; else Sign[1] = -1;  Mask >>= 1;  // t
if((Mask & i) == 0) Sign[0] = +1; else Sign[0] = -1;  Mask >>= 1;  // xy

Sign[6]  = Sign[0]*Sign[2]*Sign[3]; // xz
Sign[7]  = Sign[0]*Sign[1]*Sign[3]; // xt
Sign[8]  = Sign[0]*Sign[2]*Sign[4]; // yz
Sign[9]  = Sign[0]*Sign[1]*Sign[4]; // yt
Sign[10] = Sign[0]*Sign[1]*Sign[2]*Sign[3]*Sign[4]; // zt
Sign[11] = Sign[0]*Sign[2]*Sign[5]; // xyz
Sign[12] = Sign[0]*Sign[1]*Sign[5]; // xyt
Sign[13] = Sign[0]*Sign[1]*Sign[2]*Sign[3]*Sign[5]; // xzt
Sign[14] = Sign[0]*Sign[1]*Sign[2]*Sign[4]*Sign[5]; // yzt
Sign[15] = Sign[1]*Sign[2]*Sign[3]*Sign[4]*Sign[5]; // xyzt
```

Blade Substitutions

In Minkowski spacetime, we have ten blades which square to one, being e_q, e_x, e_y, e_z, e_{xt}, e_{yt}, e_{zt}, e_{xyt}, e_{xzt}, and e_{yzt}. We have six blades which square to negative one, namely e_t, e_{xy}, e_{xz}, e_{yz}, e_{xyz}, and e_{xyzt}. We need three positive metric basis, and one negative metric basis to define a mapping. Given that scalar basis e_q must stay assigned to the scalar slot, I had expected 9*8*7 choices for the positive slots, and 6 choices for the negative slot, thus expecting 3024 variations. However, only 72 determinant preserving mappings are found, corresponding to a magic-square pattern. The vast majority of candidates are rejected due redundancies amoung the basis and products. For example, a candidate set based on (e_x, e_y, e_z, e_{xy}) will fail, as $e_x e_{xy} = e_y$, which conflicts with the e_y basis. In effect, the magic square criteria is a completeness and uniqueness set of specifications for the alternative basis sets.

Blade Products

As with GA4E, the combination of blade post-multiplication and sign conjugation also includes the pre-multiplication and sandwich product terms. Compositing all these transforms, we have 72K (73728) distinct discrete determinant preserving transforms.

```
//  list composite Magic, Blade and Comp transforms

MV1 = Mink(a, b,c,d,e, f,g,h,j,k,l, m,n,p,r, s) ;
    for (ii=0;ii<72;ii++) { // 72K = 73728 variations
        MV2 = Magic(MV1,ii);
        for (jj=0;jj<16;jj++) { // blade prefactor
            MV3 = Blade[jj]*MV2;
            for (kk=0; kk<64; kk++) { // Comps
                MV4 = Comp(MV3,kk);
                PrintMV(MV4);
                printf(" \n");
            }
        }
    }
```

Fivespace (4,1)

Five dimensional space-time has 32 blades. The (4,1) metric designation indicates that four basis square to one, while one basis vector squares to negative one. The scalar blade e_q squares to one, and commutes wth all blades. The vector blades e_w, e_x, e_y, and e_z, square to one, while e_t squares to negative one. These five vector basis blades mutually anti-commute. The bivector blades e_{wx}, e_{wy}, e_{wz}, e_{xy}, e_{xz}, and e_{yz} square to negative one. However, due to the t component, bivector blades e_{wt}, e_{xt}, e_{yt}, e_{zt} square to positive one. The trivectors blades e_{wxt}, e_{wyt}, e_{wzt}, e_{xyt}, e_{xzt}, e_{yzt} square to positive one, while e_{wxy}, e_{wxz}, e_{wyz}, and e_{xyz} square to negative one. The quadvectors e_{wxyt}, e_{wxzt}, e_{wyzt}, and e_{xyzt} square to negative one, while e_{wxyz} squares to positive one. The pseudoscalar e_{wxyzt} square to negative one, and commutes with all geometric algebra elements, mimicing the behavior of $i = \sqrt{-1}$.

Complements

By direct enumeration, I find 64 complements in fivespace. My independent variables are q, w, x, y, z, and wx. I have derived sign logic

$$
\begin{aligned}
t &= q \wedge w \wedge x \wedge y \wedge z; \\
wy &= x \wedge y \wedge wx; \\
wz &= x \wedge z \wedge wx; \\
wt &= q \wedge w \wedge y \wedge z \wedge wx; \\
xy &= w \wedge y \wedge wx; \\
xz &= w \wedge z \wedge wx; \\
xt &= q \wedge x \wedge y \wedge z \wedge wx; \\
yz &= w \wedge x \wedge y \wedge z \wedge wx; \\
yt &= q \wedge z \wedge wx; \\
zt &= q \wedge y \wedge wx; \\
wxy &= zt; \\
wxz &= yt; \\
wxt &= yz; \\
wyz &= xt; \\
wyt &= xz; \\
wzt &= xy; \\
xyz &= wt; \\
xyt &= wz; \\
xzt &= wy; \\
yzt &= wx; \\
wxyz &= t; \\
wxyt &= z; \\
wxzt &= y; \\
wyzt &= x; \\
xyzt &= w; \\
wxyzt &= q;
\end{aligned}
$$

Blade Substitutions

I find 360 magic blade substitutions.

Blade Products

As with GA4E, the combination of blade post-multiplication and sign conjugation also includes the pre-multiplication and sandwich product terms. Compositing all these transforms, we have 720K (737280) distinct discrete determinant preserving transforms.

```
//  list composite Magic, Blade and Comp transforms

MV1 = GA5_4_1(a, b,c,d,e,f, g,h,j,k,l, m,n,p,r,s, S,R,P,N,M, L,K,J,H,G, F,E,D,C,B, A) ;
    for (ii=0;ii<360;ii++) { // 720K variations
        MV2 = Magic(MV1,ii);
        for (jj=0;jj<32;jj++) { // blade prefactor
            MV3 = Blade[jj]*MV2;
            for (kk=0; kk<64; kk++) { // Comps
                MV4 = Comp(MV3,kk);
                PrintMV(MV4);
                printf(" \n");
            }
        }
    }
```