```
size(13cm);
unitsize(1cm);

// Colors: T = outer triangle, M = Morley triangle
// Use the "invisible" pen to remove any attribute
pen legend = orange; //blue;
pen sketch = orange + opacity(0.4); //paleblue;
pen sketch2 = green + opacity(0.4);;
pen T_edge = orange + 1bp; //blue + 1bp;
pen M_edge = invisible; //blue + 1bp;
pen T_beam = interp(red, orange, 0.5) + opacity(0.25);
pen M_fill = interp(mediumred, white, 0.9);
pen T_fill = interp(white, deepgray, 0.9);
pen T_back = deepgray;

// Names of triangle vertices, angles and rotations
string[] T_vertex = {"$A$", "$B$", "$C$"};
string[] T_corner = {"$\alpha$", "$\beta$", "$\gamma$"};
string[] M_vertex = {"$a$", "$b$", "$c$"};
string[] S_vertex = {"$a'$", "$b'$", "$c'$"};
string[] rotation = {"$f$", "$g$", "$h$"};
string[] rotation3 = {"$f^3$", "$g^3$", "$h^3$"};

// Outer triangle
path T = (8.83, 47.89)--(72.74, 8.55)--(90.72, 86.63)--cycle;

// Compute inner angles of outer triangle
real[] T_angle;
for (int k=0; k<3; ++k) {
    pair CA = dir(T,k,-1), AB = dir(T,k,+1);
    T_angle[k] = angle(-CA/AB);
}

// Compute Morley triangle
guide m;
for (int k=0; k<3; ++k) {
    int l = (k+1)%3;
    pair A = point(T,k), B = point(T,k+1), AB = dir(T,k,+1), BC = dir(T,k+1,+1);
    // Compute Morley vertices
    pair a = extension(A, A + AB*expi(1/3*T_angle[k]), B, B +
BC*expi(2/3*T_angle[l]));
    m = m--a;
}
path M = m--cycle;

// Draw outer triangle
fill(T, T_fill);
for (int k=0; k<3; ++k) {
    pair A = point(T,k), B = point(T,k+1), AB = dir(T,k,+1), AC = -dir(T,k,-1);
    // Draw vertices
    dot(T_vertex[k], A, -(AB+AC)/2, legend);
    // Draw inner angles
    pair rr = (B-A)/12;
    pair z1 = rr*expi(0), z2 = rr*expi(T_angle[k]);
    draw(arc(A, A+z1, A+z2), arrow=Arrow(TeXHead), L=Label(T_corner[k],
position=MidPoint, legend), legend);
/*
    for (int i=-1; i<4; ++i) {
        pen p = (0 <= i && i < 3) ? legend : sketch;
        pair z1 = rr*expi(i/3*T_angle[k]), z2 = rr*expi((i+1)/3*T_angle[k]);
        draw(arc(A, A+z1, A+z2), arrow=Arrow(TeXHead), L=Label(T_corner[k],
position=MidPoint, p), p);
    }
*/
```

```
}
draw(T, T_edge);

// Draw Morley triangle
for (int k=0; k<3; ++k) {
    pair A = point(T,k), B = point(T,k+1), C = point(T,k+2), a = point(M,k), c =
point(M,k+2);
    // Compute symmetrical Morley vertices
    int l = (k+1)%3;
    pair s = B + (a - B)*expi(2/3*T_angle[l]);
    // Fill light beams
    fill(A--extension(A, a, B, C)--extension(A, c, B, C)--cycle, T_beam);
    // Fill Morley triangle
    fill(M, M_fill);
    // Draw trisectors
    draw(A--a--B, legend);
    draw(A--s--B, sketch);
    // Draw rotations
    position pos = Relative(0.57);
    pair a_out = (a - B)*expi(pi/2), s_out = (s - A)*expi(pi/2);
    draw(a{a_out}..s, arrow=Arrow(TeXHead, pos), L=Label(rotation[l],
position=pos, sketch), sketch);
    draw(s{s_out}..a, arrow=Arrow(TeXHead, pos), L=Label(rotation[k],
position=pos, sketch), sketch);
    // Draw vertices
    dot(M_vertex[k], a, dir(C--a), legend);
    dot(S_vertex[k], s, dir(a--s), sketch);
}
draw(M, M_edge);

// Draw f^3 g^3 h^3 = 1
pair p = point(T,2) + (point(M,2) - point(T,2))*expi(-2/3*T_angle[2]);
for (int k=2; k>=0; --k) {
    pair C = point(T,k);
    position pos = Relative(0.05);
    pair p_out = (p - C)*expi(pi/2), q = C + (p - C)*expi(2*T_angle[k]);
    draw(p{p_out}..q, arrow=Arrow(TeXHead, pos), L=Label(rotation3[k],
position=pos, sketch2), sketch2);
    dot(p, sketch2);
    p = q;
}

pair ll = min(currentpicture,true), ur = max(currentpicture,true);
label("$f^3g^3h^3=1\iff a+jb+j^2c=0$", ll+(ur.x-ll.x)/2+S*0.15cm, sketch);

// Fill background
shipout(bbox(T_back, Fill));
```