Half-Edge Mesh Data Structure

romywilliamson

August 2024

1 Introduction

I have been thinking deeply about the half-edge mesh data structure and how it can be understood abstractly, in terms of Group Actions.

I realised out that the topology of any (halfedge) mesh is completely determined by:

1. the number of halfedges
2. knowing the action of the free group $F_{\{opposite,next\}}$ on the set of halfedges.

To clarify what I mean by 2., recall that $F_{\{a,b\}}$ consists of group elements such as $aaabbabb$, so if a and b represent 'opposite' and 'next' then 'knowing the group action of $F_{\{opposite,next\}}$ on the set of halfedges, means that for a given halfedge-id, we know which halfedge-id we end up at if we follow any combination of 'opposite' and 'next' pointers.

Intuitively, knowledge of this 'group action' tells us how the halfedges are 'put together' to make a mesh - which halfedges are neighbours, which halfedges form a loop, etc. For a closed mesh (which we assume, for simplicity), every 'loop' - made by following 'next' until we get back to where we started - corresponds to a face.

This perspective allows us to understand faces, vertices and **dual meshes** in a purely algebraic way. The faces can be identified with the distinct orbits under the group action of the subgroup $F_{\{next\}}$, and the vertices can be identified with the distinct orbits under the group action of the subgroup $F_{\{opposite \circ next\}}$. (If this isn't clear, it's because repeated applications of 'next' on one halfedge make you traverse all the halfedges around one face, and repeated applications of 'opposite.next' on one halfedge, make you traverse around all the halfedges belonging to a particular vertex.)

Now, $\phi : next \mapsto opposite \circ next, opposite \mapsto opposite$ induces a homomorphism of G. (We should check that it is an automorphism! Oh wait. It is NOT an automorphism. Because everything in the output has 'opposite' in the string.)

Clearly, $\phi(F_{\{next\}}) = F_{\{opposite \circ next\}}$.

Also, $\phi(F_{\{opposite \circ next\}}) = F_{\{opposite \circ opposite \circ next\}}$.

We would *like* to say that $= F_{\{next\}} = F_{\{opposite \circ opposite \circ next\}}$. Unfortunately, that isn't true. This seems to suggest that maybe we should consider the large group acting on the halfedges to be not $G = \langle opposite, next \rangle$ but

$$G = \langle opposite, next | opposite \circ opposite \rangle \,. \tag{1}$$

If we use this definition of G, then I believe ϕ would be an automorphism.

So... if we keep the halfedges and the group action exactly the same as before, but we relabel the 'next' and 'opposite' group elements, this simply has the effect of switching the face-orbits and vertex-orbits. So applying ϕ to the group gives us the dual mesh, without any extra computation, just relabelling. In practice, this is achieved just by replacing the 'next' attribute of each halfedge by 'opposite.next' (filling in the 'face' and 'vertex' attributes is even more trivial and I would argue that these attributes aren't even essential for defining a mesh structure, since they can be deduced from the 'next' and 'opposite' attributes).

Except for providing a clean and efficient algorithm for constructing a dual mesh, what insight does this give us? I think, the automorphism makes the 'sameness' of a mesh and its dual more concrete. That's all, for now.