**The improved Permutation Algorithm**

Background

The permutation algorithm that is currently implemented in CSM takes several factors into account. It does not, however, take the connectivity into account. The connectivity is used to create the equivalency classes and that’s it.

In a few recent brainstorming sessions Gil and Inbal came up with a way to enumerate on a far smaller number of permutations, which should speed up CSM considerably.

The basic idea

The idea behind the algorithm is simple. If two atoms X and Y were connected before the applying the permutation, they should be connected after applying it. If X and Y were not connected in the original molecule, they should not be connected in the permuted molecule.

Implementation

CSM generates permutations one by one. There is a “generator” that builds permutations cycle by cycle. We will change the permutation and have it rule out illegal permutations as they are being built. We can rule out specific cycles, without having to wait for the entire permutation to be generated.

When a cycle is generated we can ensure that the connectivity inside the cycle is not violated. If everything inside the cycle is consistent, we can check the connectivity between this new cycle and all the previously generated cycles.

When we get to the last cycle in the permutation, if everything checks out, we can calculate the CSM of this permutation.