* Log Signatures directly via BCH
  + Concatenation with single displacement
  + Use coefficients calculated and distributed in the paper I presented (what Jeremy has too). The Lie brackets of basis elements can be expressed as combinations of other basis elements so easy recursion.
  + Jeremy uses the Lyndon basis for this, but the paper I studied claims that the classic Hall basis has fewer terms, so if I had to bet on which produced better performance I would go for the second one.
* Log Signatures from Signatures
  + Compute the logarithm of the signature in Tensor space effectively (Mike Giles?)
  + The Dynkin projection t2l can be represented by matrices M\_m, each one compressing level m of the expanded log-signature x\_m (in Tensor space) to its value c\_m in the Hall basis. Each column of M\_m is labelled by a basis element of the free Lie Algebra and each row is labelled with one of the d^m words of length d.
  + Jeremy solves the overdetermined system of equations M\_m\*c\_m = x\_m
  + Looking at some other numerical experiments, I think you are right, the projection Dynkin projection matrix is “sparse by block”. When we expand a bracketed expression of words into words, the words in the expansion are anagrams of the foliage of the bracketed expression (Reuteneur talks about “fine homogeneity”). So rearranging columns and rows of each M\_m to put anagrams close together we get diagonal matrices. I think Jeremy solves a separate linear systems for each class of anagrams of words of length m, by pre-determining all the mapping matrices between anagram classes of the log-signature and its expansion beforehand, and then by simply calculating inverses in one go. Finally, he uses Lyndon here because the first word in the expansion of a bracketed expression in Lyndon basis elements for the log-signature is always the foliage, so when he removes all rows corresponding to non-Lyndon words, he is left with a mapping from an anagram class to Lyndon elements in Tensor space, which is just lower a squared, lower triangular matrix with one on the diagonal! So super quick to invert.
* Other than that, Reutenauer talks about a unique linear map defined on the truncated tensor space that matches the logarithm function exactly on grouplike elements. He gives an expression for it (which is hard because it involves inner products of shuffled elements) and an expression of its dual, which is easy to calculate. With the adjoint operator, the expression for the log-signature element corresponding to the Hall basis element labelled by the Hall word w, is simply the adjoint operator evaluated at the corresponding dual Lie basis element P\*\_w. Reutenauer gives an explicit construction of this dual basis when the initial basis is the Poincare-Birkhoff-Witt basis. I need to re-read this construction because I did not follow the first time. Not sure it is easy to implement on a computer…