We introduce ALCH, an imperative language for describing programs in the CRN-controlled tile assembly model (CRN-TAM), as well as an ALCH compiler and simulator. ALCH supports many of the features of the C programming language and contains a nondeterministic “branching” structure that allows us to query assemblies as they are built.

We also present a strict construction of the discrete Sierpinski triangle (DST) in the CRN-TAM. It has already been shown that the CRN-TAM is as powerful as the aTAM and that it is impossible to strictly construct the DST in the aTAM; therefore our construction demonstrates that the CRN-TAM is strictly more powerful than the aTAM. ALCH allows us to describe the DST construction in a convenient high-level form. We can therefore abstract away details of chemical species and reactions and reason at the level of algorithms.

Using the ALCH language we show other shapes and fractals can be defined and then compiled into the language of molecules.

**What Students Learned**

- Molecular programming paradigms
- Molecular programming techniques
- How to write a simulator
- How to design a programming language
- How to write a compiler
- Computer science data structures
- How to write proofs about complex systems
- Computer science algorithms
- How to present research to a scientific audience

Find the entire paper from DNA 26 here.
You can also find the poster from DNA 25 here.

This research was supported in part by NSF grants 1900716 and 1545028.