Qian, Soloveichik, and Winfree proposed a one way of using DNA to create a Turing-complete computation system [1]. Instead of directly implementing a Turing machine, their proposal involves a multiple stack system that is Turing complete and as efficient as multi-tape Turing machines [2]. This decision relates to DNA polymers lending themselves more readily to creating stacks than traditional Turing tapes.

The basic structure consists of one or more stack polymers where each molecule represents a particular letter . Chemical reaction networks are used as state transitions, adding and removing molecules from the stack. Importantly, this is done in a way that is reversible, which allows a particular addition or removal to be undone without adding to the energy cost [1].

Reactions rely on a DNA fuel species that is specific to a particular DNA molecule. For example, fuel species F1X applies only to attaching molecule X to the stack 1 polymer. Additionally, each time a molecule is added to the stack, it releases a confirmations molecule (Q) that is used later when querying the stack. This Q molecule is also easily changed from a stack specific molecule (Q1) to a generic form.

The stack consists of a polymer with a fixed end and a growing end. The fixed end is denoted with a special molecule that indicates an empty stack. Molecules are added to the stack when both the correct fuel and molecule are present. It is worth noting that several fuels may be attempted unsuccessfully before the correct match is made. If a fuel (specific to a particular stack) attempts to bond with that stack polymer, it will only succeed if the correct input molecule is present as well. For instance, any fuel F1Y may attempt to bond with stack 1, but will fail if molecule X instead of Y is present. This can result in many fuel attempts before F1X successfully bonds X to the growing end of stack 1.

For an example computation, consider an input polymer (S1) that is to be copied to two output polymers (S2 and S3) using an alphabet of {0,1}. The state transitions consist generally of popping S1 and copying that molecule to S2 then S3. Using the Q molecule, the top element of S1 is removed. This leads to the states where molecules representing that same element are added to S2 and S3, creating their own Q confirmation molecules. This process is repeated until S1 reaches the empty stack molecule. There is an illustration of this process in the figure from [1].

This is only one example from the overall field and there are drawbacks to this method as well as advantages. The reversible reactions are more efficient than having two irreversible forward/back reactions, which is a considerable advantage. However, the need to have unique free-floating polymers (stacks) in the same solution limits the scope and parallelization of this method. Additionally, some other methods are able to use complete material recycling outside of the computation output, while this approach consumes materials.

[1] Qian, Lulu, David Soloveichik, and Erik Winfree. 2010. Efficient turing-universal computation with DNA polymers. In Proceedings of the 16th international conference on DNA computing and molecular programming (DNA'10), Yasubumi Sakakibara and Yongli Mi (Eds.). Springer-Verlag, Berlin, Heidelberg, 123-140.

[2] Minsky, M.L.: Computation: finite and infinite machines. Prentice-Hall, Englewood Cliffs (1967)

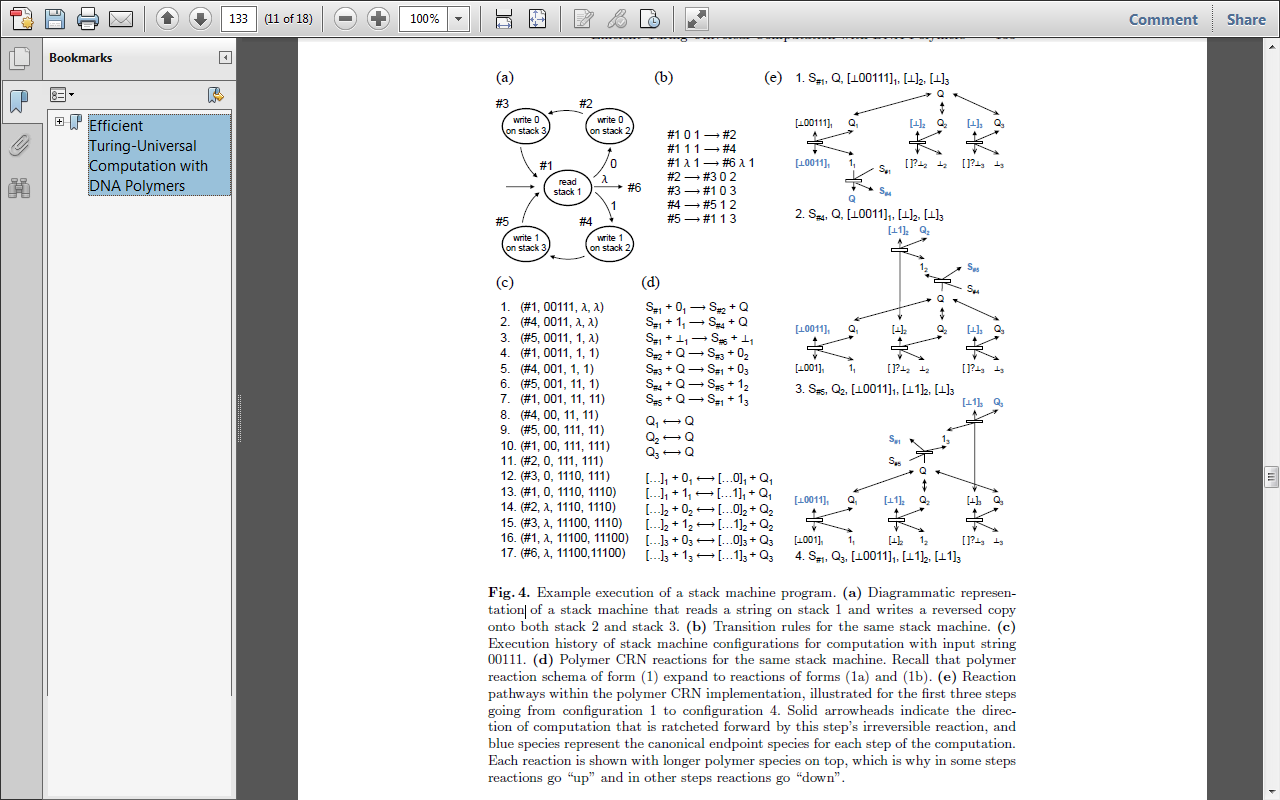


Figure courtesy <http://www.dna.caltech.edu/DNAresearch_publications.html>