Universality, Hardness, Engineering, and Messiness

I will review some of the ways that physical and dynamical systems can carry out universal computation, such as simulating Boolean circuits with cellular automata and sandpiles, and simulating Turing machines with iterated maps or flows in low-dimensional chaotic systems. I will then comment on the fact that every known proof that a natural system is hard to predict (undecidability, NP-completeness, etc.) relies on an ability to "build a computer" in that system. Systems for which we can do this are computationally powerful, but easy to think about — in the sense that we can do engineering in them, designing gadgets that store and transmit information in controlled ways.

In contrast, many natural systems seem very messy. Their dynamics are complicated, but complicated in a way that makes it hard to imagine building a computer out of them. Such systems could be uncomputable but not computationally universal. The challenge of synthetic biology is to engineer within the messinesss of biological systems: but many real systems might lie in this middle zone, where dynamics are complex but difficult to control.