Proposed subtopic for GC7: "Journeys in Non-Classical Computing"

Computation with Dynamics and Attractors

Susan Stepney, University of York

Introduction

The existing GC7 Journey of *Non-Classical Interactivity – Open Dynamical Networks* includes the following rather cryptic paragraph:

> *Computation as a dynamical process*. What are the various attractors of a dynamical computation? How can we encourage the system to move to a "better" attractor? How can we map the route through intermediate attractors that it should take?

This paper expands a little on that paragraph, arguing for the development of a novel model of computation.

Dynamical Computation

Physical dynamical processes are characterized by motion in a phase space, controlled or directed by various attractors (so called because they "attract" the trajectory of the system to them). As various parameters of the system change, the shape of the resulting attractor space can also change, and so the trajectory may find itself being attracted to a different region of the space. [Kel], for example, uses these and related ideas to explain many features of organisms' behaviour, from gait patterns to learning and recognition tasks.

One might like to think of this dynamical behaviour in computational terms, with the attractors as "states" in the phase space, and the trajectories between them as "state transitions". This is a suggestive analogy, yet the conventional state transition model has a rather static feel to it. States and their transitions tend to be predefined, and the execution of the transitions has to be explicitly implemented by the computational system. Contrastingly, the attractors are natural consequences of the underlying dynamics, and new attractors and resulting trajectories are natural consequences of changes to that underlying dynamics. A dynamical system is relatively robust (a small perturbation to the trajectory will usually leave it moving to the same attractor), and computationally efficient (the computation is a natural consequence of the physical laws of the system, and does not need any further implementation beyond that of the dynamical system itself).

The Challenge

The challenge is thus: **to develop a new computational paradigm expressed in dynamical terms of attractors and trajectories**. Does the state transition analogy hold? Can a computation be expressed as a trajectory amongst various attractors, each changing as the result of some parameter/input? What are the programming primitives and higher level languages? What are the logics, reasoning approaches, and refinement calculi? What are the compilers and other development tools? What kinds of algorithms are most suited to this paradigm? What are the implementation mechanisms? How can we simulate these systems on classical machines?

References

[Kel] J. A. Scott Kelso. *Dynamic Patterns: the self-organization of brain and behavior*. MIT Press. 1995