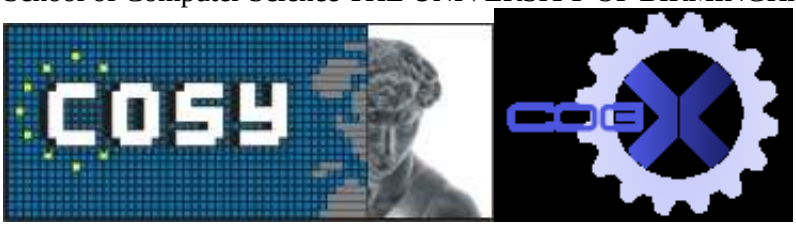
School of Computer Science THE UNIVERSITY OF BIRMINGHAM



Kinds Of Dynamical System: A request for help

Installed: 23 Nov 2010
Last updated: 23 Nov 2010; 13 Jul 2011

Common views about dynamical systems

This is a request for help/comments/criticisms.

This is an unfinished attempt to generalise some of the points made in this paper
http://www.cs.bham.ac.uk/research/projects/cogaff/misc/dynamical-systems.html

I am trying to write up some ideas about the sorts of (mostly virtual, not physical) dynamical systems that seem to run on brains and are likely to be needed in future intelligent machines.

Most discussions of dynamical systems have made a number of highly restrictive assumptions, listed below. I shall also list ways in which these assumptions need to be relaxed. The request for help is for pointers to work on the broader notion and assistance in improving the specification and perhaps beginning to develop some simple illustrative examples.

1. A Dynamical system (DS) has a single state defined by a state-vector which is a set of numbers ("scalar values") or in some cases labels, each being a member of an ordered set, ("nominal values") where the values in the vector represent physical states of components.

The set of possible state vectors is the state space.

2. The state changes of the DS are subject to a set of dynamical laws that can be expressed as differential equations or difference equations (for instance in the case of cellular automata -- like Conway's Life).

3. Changes that are discrete are synchronized (i.e. all the difference equations describe what happens in a time step).

4. For continuous changes there's a global time-frame, so that rates of change in different parts of the system are comparable.

5. A DS has (discrete, or continuous, or hybrid) trajectories defined by the changes of the state-vector.

6. A DS may be autonomous (e.g. it just starts off in some state and then runs, like most versions of Conway's life) or embedded in a larger system, which itself is a dynamical system and can influence or be influenced by some of the components of the state vector.

7. A DS has attractors: regions of the state space that a trajectory will not leave once it has entered, unless some external influence alters the trajectory.

There are other features of state-spaces arising from relations between attractor

basins and whether the behaviour is or is not chaotic etc.

8. A DS Is a fixed structure insofar as the number of components of the state-vector is fixed and the set of dynamical laws is fixed.

Requirements for cognitive dynamical systems (CDSs)

In contrast, we obviously need to talk about dynamical systems that violate almost all this constraints listed above, if we want to understand animal cognition.

I'll talk about a CDS (Cognitive Dynamical System).

1. Instead of having a single state, with a single trajectory, with global attractors, a CDS needs more or less disconnected enduring sub-systems which are separate CDSs, which are more or less closely coupled with other CDSs, and where the different CDSs may change their states at different rates and asynchronously relative to one another.

For example, a human-like (adult) cognitive system typically needs several varieties of perceptual subsystems, various action control subsystems, various subsystems concerned with generation and management of motivation, subsystems capable of reasoning, planning, problem solving, explanation-generation and self-monitoring, different kinds of learning, language understanding and language generation sub-systems: putting these all into one big mush of a dynamical system would produce something totally unmanageable -- even by evolution.

2. The changeable components of a CDS are not restricted to scalar or nominal values, but can include a variety of structured objects including logical or algebraic formulae, grammars (or something with similar functionality), parse trees or charts or other structures concerned with understanding complex inputs, (representations of) geometric shapes, fragments of shapes (e.g. a portion of a curved surface), topological structures (including ordered lists, trees, graphs, other discrete or continuous manifolds, and perhaps things that map onto things like chemical structures), and perhaps also processes (e.g. program executions, vibrating structures, a rehearsed password, fragments of tunes...)

Not all the changes are continuous, or usefully representable by differential equations, or difference equations. The possibilities for a given substructure need not form a linear space: e.g. the permutations of a list of names, the set of operators applicable to a particular equation, or the set of possible quadrilateral shapes.

For all those reasons, the state space of a particular CDS is not usefully represented as a point in a vector space of fixed dimensionality.

3. the subsystems need not have fixed numbers of components: e.g. the parse tree or graph grown by a sentence-processing subsystem can have different numbers of components at any one time (which may be borrowed from and later returned to a pool of components that are available for temporary use in different sub-systems. (I.e. a 'heap').

4. to be continued...

Maintained by Aaron Sloman
School of Computer Science
The University of Birmingham