**Emergent properties in reactive systems**

**Aiguier et al., 2008**

Reactive systems consist of interconnected sub-components part of structural links that define communication methods. These systems can exhibit emergent properties that are unpredictable even when complete knowledge of the systems is available. This implies complexity in the systems, such that they cannot be simplified to rules based on inferences from their properties, but knowledge of the rules of interaction between components is also necessary.

Emergent properties are often difficult to predict.

**Shape-memory polymers**

**Behl et al., 2007**

Shape-memory polymers are active polymers with the capability of dual shapes, changing from one shape to the other as a response to a particular stimulus.

**An overview of novel soft actuators for soft robotics**

**Boyraz et al., 2018**

**The advantages of generative grammatical encodings for physical design**

**Hornby & Pollack, 2015**

Genetic algorithms typically use direct encodings of solutions, but may struggle to cope with designing highly complex systems using these direct methods.

Generative encoding is a type of encoding that specifies the construction of the phenotype. It may be more scalable because of its self-similar and hierarchical structure.

**Evolving virtual creatures**

**Sims, 1994**

A genotype is a programmed representation of potential individual or problem solution.

A phenotype is set of characteristics of an individual as they result from the composite of its genotypes.

**The algorithmic beauty of plants**

**Prusinkiewicz et al, 2004**

A shape and the cascade that generated it are both called self-similar when every piece of the shape is geometrically similar to the whole.

*Chapter 1 Graphical modelling using L-systems*

L-systems were conceived as a mathematical theory of plant development. They did not originally include enough detail to completely model higher plants. They focused on topology and not geometry.

*1.1 Rewriting systems*

The main component of L-systems is rewriting.

Rewriting is defining complex objects by successively replacing parts of an initial, simple object according to a set of rules.

In L-systems grammars are applied in parallel and simultaneously replace all letters in a given word.

*1.2 DOL-systems*

The simplest class of L-systems, namely DOL-systems, are deterministic and context-free.

*1.4 Synthesis of DOL-systems*

With edge rewriting, productions replace polygon edges with figures.

With node rewriting, productions operate on polygon vertices.

*1.7 Stochastic L-systems*

Variation in productions can be attained by implementing randomization in the turtle interpretations and/or the L-systems.

*1.8 Context-sensitive L-systems*

A production’s expression may depend on predecessors’ context.

*1.9 Growth functions*

A growth function describes the number of symbols in a word in terms of its derivation length.

Growth functions of DOL-systems are independent of the order of letters in a production and derived words.

*1.10 Parametric L-systems*

Turtle interpretation of L-systems is limited due to the fact that all lines are reduced to integer multiples of the unit segment.

*Chapter 2*

Some computer simulations of branching patterns consider interactions among growing features, structures and the environment. This makes models more complex and realistic, but not all models take these interactions into account.

*Chapter 3*

Data base amplification is the generation of seemingly complex objects from very concise descriptions.

*3.1 Levels of model specification*

Partial L-systems use notation of nondeterministic OL-systems to define the different possible structures of a given type that can develop. They capture the main traits that characterize a structural type and provide a formal basis for their classification.

L-system schemata are control mechanisms that resolve nondeterminism. The topology of individual productions and temporal aspects of their development are described at this level.

Complete L-systems add the geometric aspects.

*3.3 Models of inflorescences*

Inflorescences are compound flowering structures.

*Chapter 4*

Phyllotaxis is the regular arrangement of lateral organs of a plant.

*Chapter 6*

Original L-systems are discrete in time and space, meaning model states are known only at specific time intervals and only a finite number of model states exist. Parametric L-systems allow for infinite model states due to the assignment of continuous attributes to model components.

*6.2 Selection of growth functions*

Timed L-systems capture qualitative changes in model topology corresponding to module divisions and return its age as a function of the global time.

*Chapter 7*

Map L-systems allow for the formation of cycles in a structure.

*7.1 Map L-systems*

A map is a finite set of regions, where a region is surrounded by a boundary made up of a finite, circular sequence of edges meeting at vertices. Each edge has one or two vertices associated with it; one if the edge forms a loop. Edges cannot cross without forming a vertex and there are no vertices not associated with an edge. Every edge is part of the boundary of a region, and the set of all edges is connected.

*Chapter 8*

A fractal is a set with Hausdorff-Besicovitch dimension strictly exceeding the topological dimension .

A finite curve can be considered an approximate rendering of an infinite fractal as long as their respectively interesting properties are related, i.e. self-similarity.

**Complex networks from simple rules**

**Southwell et al., 2013**

**Composing secure systems that have emergent properties**

**Zakinthinos and Lee, 1998**

Emergent properties occur when not all components of a property satisfy the property.

An emergent property is not satisfied by constituent components of a system, but is satisfied by the overall system.

If the required condition for a specific property to emerge can be determined, it might be possible to construct a system with that property from components that do not satisfy that property.

If the target property of a system is known, and the property of a component is known, then it can be determined if the other component can have a property that will result in the system having the target property. If the property exists, it can be found.