Parts closure in a kinematic self-replicating programmable constructor

William M. Stevens
Department of Physics and Astronomy
Open University
Milton Keynes, United Kingdom, MK7 6AA
william@stevens93.fsnet.co.uk
www.srm.org.uk

Abstract

A self-replicating automaton capable of being programmed to construct a wide range of objects in addition to a replica of itself may be termed a self-replicating programmable constructing automaton (SRPC).

An open problem in the field of self-replication is the design of a physical SRPC made from simple parts taken from a small set and using for its supply of parts (from which to construct things) a disorganised collection of parts distributed in a random fashion in its environment.

A machine capable of doing this will need to contain a subsystem capable of taking an unknown part from its environment and determining which kind of part it is. The architecture of a discriminating subsystem of this kind is of course highly dependant on the precise nature and size of the set of parts used.

General principles of this type of disciminating system are discussed, and a specific design for such a system in an abstract discrete-space kinematic simulation environement is presented.

1 Introduction

The concept of parts closure in self-replicating systems was first clearly defined in [1]. In the context in which the term was first used, it referred to the ability of a system to manufacture all of the parts from which it is made. The definition assumed that a system would process raw materials from its environment to make parts. When applied to the class of automaton that this paper is concerned with, in which parts are not manufactured from raw materials but are distributed at random in the automaton's environment, the term 'parts closure' can be defined as the ability of such an automaton to recognise and classify all parts from which it is made. The kinematic self-reproducing automaton concept described by von Neumann in [3] depends upon this ability. Other proposed or actual kinematic self-replicating systems depend upon having known parts in known locations, for example [2] and section 5.6.4 of [1].

2 General Principles

In its most general form, a part discriminating subsystem D of an SRPC takes an unidentified part p from a set P as an input and generates some indication of which type of part p is. Since p will be used later by the self-replicating system, part p is also output from D. Since D is part of a self-replicating programmable constructor, D is itself made from parts taken from the set P.

There are various ways in which D may identify p. An obvious and intuitive way is to use visual inspection; this is the way that many large animals recognise the objects that we encounter. An advantage of using visual inspection is that it is a very general method: it can be used to distinguish between any parts which differ in external appearance. There are several disadvantages of this method, however. Firstly D must contain a camera, a light source and a computer sufficiently powerful enough to be able to process images from the camera. This adds considerable complexity to D and requires that P be rich enough to support these devices.

To make the job of D easier, parts can be deliberately designed to be easily distinguishable. For example, for easy recognition by a visual inspection system, parts could have different reflectivities. Bar codes, shape-fitting and having different masses for different parts from P are other possible methods of providing for easy recognition.

If we choose not to use a visual method for identification, nor deliberately augment parts from P with

features that make them easily distinguishable, then parts must be identified by examination of their geometrical or physical properties, or by their function. This examination may be done explicitly. For example, if the part is prodded in a particular way and as a result exhibits a particular behaviour, then this behaviour may distinguish it from other parts. Alternatively identification could result from the interaction of parts from P with some substrate. For example, in chemical chromatography, different molecules are identified by their speed of passage through a medium which selectively hinders molecules according to their shape or nature.

The discriminating process need not be deterministic. Figure 1 is a schematic diagram of a discriminating system D in which a particular type of part p enters through input i and will emerge from exit o_n of D with probability $F(n, p)$.
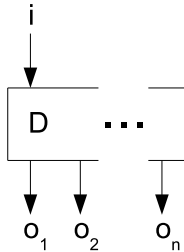

Figure 1: A schematic diagram of a non-deterministic discriminator.

Assuming that $F(n, p)$ is independent of previous tests of p, and also that for different types of part p and q there is at least one n such that $F(n, p) \neq F(n, q)$, p can be identified with any required degree of certainty by passing p through D as many times as necessary to reach that degree of certainty. The logic required to assess whether a particular degree of certainty has been reached need not be sophisticated, and could consist of a set of resettable counters to count the number of times p emerges from each O_n and a set of comparators to determine when counts reach a level at which p can be reliably identified. A non-deterministic discriminator of this kind is likely to be easier to implement than one which must always give a definite identification. Indeed, for some systems almost any vertically mounted conduit with one entrance and two or more exits containing obstructions to parts dropped into the top may function as a non-deterministic discriminator. Such a discriminator is depicted in figure 2. The function $F(n, p)$ may need to be obtained empirically by dropping known parts through the discriminator.
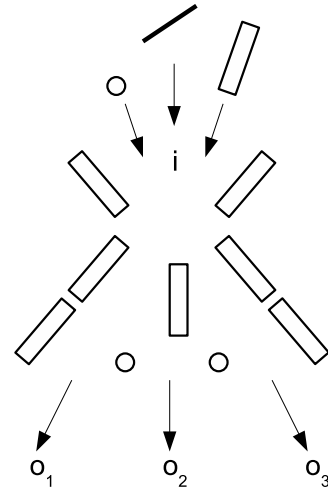

Figure 2: A non-deterministic discriminator.

3   A specific design

This section describes a specific design for a discriminating system, based around a set of 6 simple parts that in some sense span the range of elementary operations that a physical self-replicating programmable constructing machine needs to be capable of. The environment in which the design exists is a three dimensional discrete space (and discrete time) kinematic environment which, although not directly physically realistic, has more physical realism than the cellular automaton environments that have been used to investigate self-replicating programmable constructing automata in the past. The design has been fully simulated in this environment.

Parts are moveable cubes, each type of part performs a particular function. The six part types used are described briefly here with reference to figure 3.

A *wire* part receives a boolean input value at its base and propagates this value to all five of its outputs after one time unit.

A *nor-gate* part receives boolean input values from all five of its inputs, and propagates the logical NOR of these values to its top-most output after one time unit.

A *slide* part receives a boolean input value from its base. If this value is true, then one time unit later it will move any part immediately above it one unit to the right.

A *rotate* part receives a boolean input value from its base. If this value is true then one time unit later it will rotate any part directly above it counterclockwise
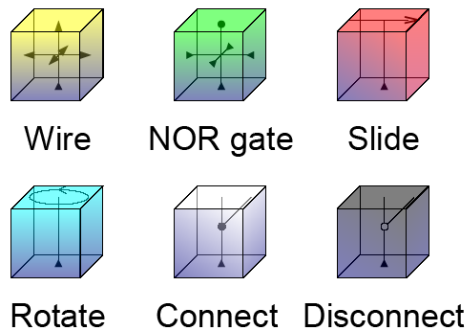
Figure 3: Six types of part.

about the axis joining the centres of the rotate part to the part above it.

A *connect* part receives a boolean input value from its base. If this value is true then one time unit later it will connect together the parts lying immediately above it, and above right.

A *disconnect* part receives a boolean input value from its base. If this value is true then one time unit later it will disconnect the parts lying immediately above it, and above right.

The discriminating system consists of a series of six filters, each of which tests for a specific part in a particular orientation. The part being tested passes from one filter to another, unless identified by a particular filter, in which cases it is put to one side and a new unknown part is obtained from the environment. If a part passes through all filters undetected, its orientation is altered and it is sent through the set of filters again. A part will be identified after at most 24 passes through the set of filters.
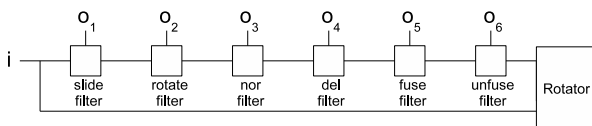

Figure 4: The logical structure of the discriminator.

The logical structure of the discriminating system is shown in figure 4. *Wire* and *slide* parts can be used to make paths along which individual parts can be transported, an example of which is shown in figure 5. In figure 5, when a signal is applied to the input of the left-most *wire* part, any part above the left-most *slide* part will move rightwards along the path.

Figure 6 shows a *nor* filter. A single path leads

into the filter, and two paths lead out. The filter is capable of detecting a *nor* part, orientated such that its output points south (see axes in figure 6). A part p part travels into the filter along path A, passing above the input to part B as it does so, and arrives at point C. Since a southward pointing nor part is the only possible part that can cause a signal to enter part B as it passes above it, a signal taken from B to path D can be used to divert part p if and only if p is a southward pointing nor part. If p is any other part, or is a *nor* part in any other orientation, it will exit the filter when part E slides it out of the filter (after having waited at C for long enough for any possible diversion to happen).
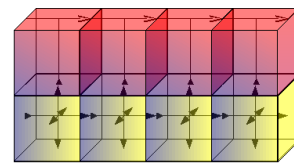

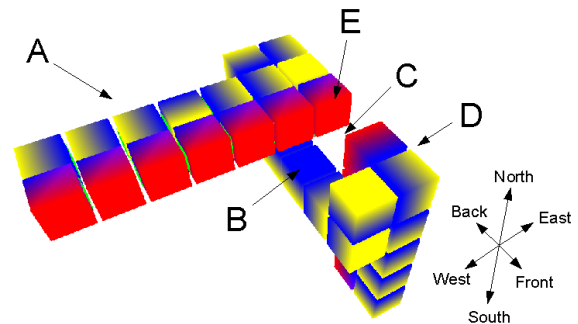Figure 5: A path along which other parts can travel.

Figure 6: A filter for detecting *nor* parts.

A *wire* filter can be constructed in a very similar way to the *nor* filter, except that in this case a signal must be fed into the part being tested so that any possible output from it can be detected. Since a southward pointing wire is the only part that will output a signal one time unit after receiving a signal on its input, a part from which such a signal is detected can be diverted and identified as a southward pointing *wire*.

Filters for *connect* and *disconnect* parts are more complex. Both work in a similar way, so only the *connect* filger will be described.

Figure 8 shows a *connect* filter and figure 9 shows an expanded view of the filter, so that the internal parts of the filter can be seen. A part P is fed into the filter
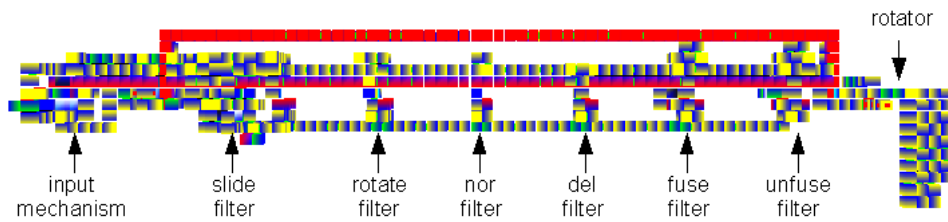
Figure 7: The complete discriminating system.

at A at the same time that a signal is fed into the filter at B. p is transported to location C. A signal derived from B is applied to *disconnect* parts G so that there is no connection at joints F. After this, a signal derived from B is applied to the northward face of p by part D. If and only if p is a southward-facing *connect* part will the rightmost of the joints F become connected. *Slide* parts I will slide the structure beneath them back and forth, as a result of which, the signal output by *nor* part G will arrive at H, but only if p was a southward-facing *connect* part. A signal derived from H can then be used to divert p along J, otherwise p will exit from the filter at L.
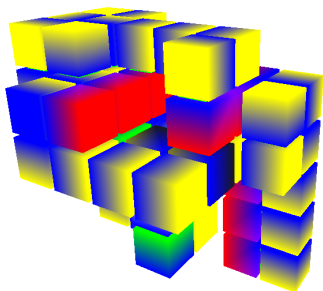

Figure 9: Expanded view of figure 8.

Figure 8: A filter for detecting *connect* parts.

The *slide* filter is the most complex of all the filters and is the first one that parts encounter when entering the discriminator. The main reason for the complexity arises from the need to detect a *slide* part in any of the four possible southward pointing orientations before passing the part onto successive filters. If such a part were to enter any other filter the result would be unpredictable because if a test signal were applied to the part it would cause the part beneath it to move, and possibly cause the discriminator to move one unit east, west, back or forth in an undesirable way.

A full description of the *slide* filter is omitted here due to space limitations. The *rotate* filter is relatively
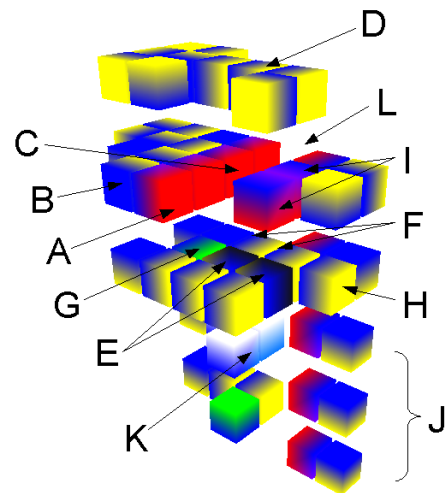
simple, but again a full description is omitted here.

Figure 7 shows the discriminating system in its entirety.

References

[1] Robert A Freitas Jr and William P Gilbreath. Advanced automation for space missions. *NASA Conference Publications*, CP-2255 (N83-15348), 1982. Online at http://www.islandone.org/MMSG/aasm/.

[2] Matthew Moses. A physical prototype of a self-replicating universal constructor. Master's thesis, Department of Mechanical Engineering, University of New Mexico, 2001.

[3] John von Neumann and Arthur W Burks. *Theory of Self-Reproducing Automata*, pages 81–82. University of Illinois Press, Urbana, Illinois, 1966.