MACM 202 Assignment 3, Fall 2005

Michael Monagan

This assignment is worth 10% of your grade. It is due Monday October 24th at 12:00noon. A late penalty of 20% will apply for each day late. Do question 1, either 2 or 3, and question 4. Do each question in a separate Maple worksheet and hand in a printout of each worksheet. Note: we intend to mark all (three) questions.

Question 1: Stephen Wolfram's Experiment (30 marks)

Consider a one-dimensional boolean cellular automaton where the value of cell i at time $t+1$ is a function of the values of cells $i-1, i, i+1$ at time t. Wolfram investigated what happens for all 256 boolean functions with three inputs. I want you to repeat his experiment. First, write a loop which constructs and prints out all 256 boolean functions. Perhaps something like this

```
> for i from 0 to 255 do
>   ...
>   f(0,0,0) := ... ;
>   f(0,0,1) := ... ;
>   ...
>   f(1,1,1) := ... ;
>   lprint(i, (0,0,0)=f(0,0,0), (0,0,1)=f(0,0,1), ..., (1,1,1)=f(1,1,1));
> od:
```

That is, you can define a function in Maple for a discrete set of values. Now, using the networks package, for each of the 256 boolean functions, construct a cellular automaton with 101 nodes arranged in a loop. Run your network for 50 time steps with initial state $S_0 = [0, ..., 0, 0, 1, 0, 0, ..., 0]$. For each functions, draw the output using the drawrun command or the plots[listdensityplot] command. Identify which boolean functions produce a chaotic behaviour. Hand in a print out of the chaotic ones.

Question 2: Forest Fires in 2-Dimensions (40 marks)

Implement in Maple a procedure `run2dca(f,S0,m)` that runs a 2-dimensional cellular automaton on an n by n grid (with cyclic boundary conditions) where f is a function of the 8 neighbouring nodes and itself in the grid, S0 is the initial state of the cellular automaton (a Maple list of n lists of n numerical values) and $m > 0$ is the number of steps the automaton is run. Your procedure `run2dca` should output a list of $m + 1$ states where each state is a list of lists of numerical data.

Now model a forest fire in two dimensions as a 2-dimensional cellular automaton. A forest fire is an example of an excitable media. In the model you should initially assume that a quiescent state will burn (excite) if any of it's eight neighbours are burning. Moreover, a quiescent state should spontaneously excite (ignite) with low probability e.g. 0.0001. Run the model on a grid of size at least 20 by 20. You will need to run it for 40 or more steps. You can visually look at each state in the output using the plots[listdensityplot] command.

What you will find is that a forest fire will burn outward in a a *square wave* – which is not realistic – and, if there is another fire in another region, when the fires meet they will annihilate each other – which is realistic. Modify your cellular automaton so that the fire will burn outwards in a circular wave. Try to model a wind blowing from west to east.

Notes: Use the plots[listdensityplot] command to show selected states of runs of the cellular automaton.

You may generate uniform random numbers on [0,1) with

```
> R10 := rand(10^10);
> U01 := proc() Float(R10(),-10) end;
> U01();
```

Question 3: Conway's Game of Life (40 marks)

Implement in Maple a procedure `run2dca(f,S0,m)` that runs a 2-dimensional cellular automaton on an n by n grid (with cyclic boundary conditions) where f is a function of the 8 neighbouring nodes and itself in the grid, S0 is the initial state of the cellular automaton (a Maple list of n lists of n numerical values) and $m > 0$ is the number of steps the automaton is run. Your procedure `run2dca` should output a list of $m + 1$ states where each state is a list of lists of numerical data.

Run the automaton using the game of life boolean function starting from some chosen initial states and some random initial states, to find at least three initial states, not shown in class, that do not die out. Use the plots[listdensityplot] command to display selected states of a run of the cellular automaton.

Note, the rules in the text say that a cell is born (goes from 0 to 1) if exactly 4 neighboors are alive. Try using 3 instead of 4. This will produce better results. You will need to use a grid of size 20 by 20 or more. You may generate a random starting state as follows (this generates a 1 with probability 1/3).

```
> r := rand(1..3):
> B := proc() if r()=1 then 1 else 0 fi end:
> S0 := [ seq( [seq(B(),i=1..n)], j=1..n) ]:
```

Question 4: Serpinski's Gasket (30 marks)

Consider Serpinski's gasket as shown on page 112. Suppose p, q, r are the vertices of the initial triangle. Let s, t, u be the midpoints of the line segments $\bar{p}\bar{q}$, $\bar{q}\bar{r}$ and $\bar{r}\bar{p}$ respectively.

One way to draw Serpinski's gasket it is to draw all the little triangles. You can either draw the triangles in black as shown or just draw the boundaries; either way you'll get a good picture. Write a recursive Maple procedure to to this. For example, we can draw a black triangle p, q, r as a polygon POLYGONS($[p, q, r]$,COLOR(RGB,0,0,0)) or the boundary as CURVES($[p, q, r, p]$,THICKNESS(2)). We can draw the three black triangles as indicated in drawGasket[1] as either

PLOT(POLYGONS($[p, s, u]$,$[q, s, t]$,$[r, t, u]$,COLOR(RGB,0,0,0)))

or

PLOT(CURVES($[p, s, u, p]$,$[q, s, t, q]$,$[r, t, u, r]$,THICKNESS(2)))

A second way is described on pages 121–124 as the "Fractal game". Start with a point u_0 in the triangle p, q, r. Choose one of p, q, r at random. Say we choose q. Now compute $u_1 = (u_0 + q)/2$. Repeat this. Graph the points $u_0, u_1, u_2, \ldots$. You should get a picture of Serpinski's gasket.

Write a Maple procedure to do this. To generate a plot of the points use

PLOT(POINTS($u_0, u_1, u_2, \ldots, u_n$))