Most of these questions require familiarity with built-in functions, or at least familiarity with looking for them in the Help browser. Some are open ended, or at least have open ended variants. Try to answer them using *Mathematica*, even though some could be handled by paper and pen. For most of them, there is no single right way of doing it.

Basic Evolutions

For the following, also write graphics functions to visualize the evolutions at the same time as writing functions to perform the evolutions.

1. (p.24) A cellular automaton updates a one-dimensional grid in parallel. The next color in a cell depends on its current color and those of its immediate neighbors.
Write a function equivalent to CellularAutomaton[110,{0,0,1,0,0},2] using the functions Partition, ReplaceAll (/.),NestList and the update rule for 110

```
{{1, 1, 1} → 0, {1, 1, 0} → 1, {1, 0, 1} → 1, {1, 0, 0} → 0,
 {0, 1, 1} → 1, {0, 1, 0} → 1, {0, 0, 1} → 1, {0, 0, 0} → 0}
```

where the above rule gives the CA function by {left neighbor, center cell, right neighbor} → next color for the center cell.

2. (p.71) A mobile automaton updates a one-dimensional grid one cell at a time. The next color of the active cell and the next position of the active cell depend on its current color and those of its immediate neighbors.
Write a function that implements the evolution of a mobile automaton given as follows.

```
{{1, 1, 1} → {0, -1}, {1, 1, 0} → {0, -1}, {1, 0, 1} → {0, -1}, {1, 0, 0} → {0, -1},
 {0, 1, 1} → {0, 1}, {0, 1, 0} → {1, -1}, {0, 0, 1} → {1, 1}, {0, 0, 0} → {1, -1}}
```

where the above rule determines the MA function by {left neighbor, center cell, right neighbor} → {next color for the center cell, relative position of the next active cell}

3. (p.88) A sequential substitution system uses a replacement rule at the first position possible to update at every step.

a. Write a function for sequential substitution using strings as states, using the function StringReplace. The rules have the form

```
{"AAB" → "BA", "B" → "ABA"}
```

b. Write a function for sequential substitution using Replace where the states are represented by lists of 0s and 1s. The rules may have the following form (or use something similar)

```
{{x___, 0, 0, 1, y___} :→ {x, 1, 0, y}, {x___, 1, y___} :→ {x, 0, 1, 0, y}}
```

4. (p.78) A Turing machine updates a one-dimensional grid one cell at a time, as well as updating its internal state. The next internal state of the machine, the next color of the active cell and the next position of the active cell depend on its current state and color.
Write a function that implements the evolution of a Turing machine without using the TuringMachine function, with rules given as in the following.

```
{{1, 1} → {1, 0, -1}, {1, 0} → {1, 0, 1}, {2, 1} → {2, 0, 1}, {2, 0} → {2, 1, -1}}
```

where the above rule gives the Turing machine update by {state, color of active cell} → {next state, next color for the active cell, relative position of the next active cell}

5. (p.93) A tag system updates a finite one dimensional list by reading a fixed number of entries, joining another list to the end depending on those entries, and deleting the entries which were read.
Write a function that implements the evolution of a tag system given as in the following.

```
{{1, 1} → {0, 1, 1}, {1, 0} → {1, 0, 1}, {0, 1} → {0, 0, 0}, {0, 0} → {0}}
```

where the above rule gives the tag system by {leftmost entries} → {new rightmost entries}

Basic Manipulates

For the following write an easy to use, easy to understand, dynamic visualization, with labels that make sense.

1. Write a Manipulate function that allows the user to move a disk along a sinusoidal path in the plane. Also, allow the user to vary the size and color of the disk. (More technically challenging is to make the position parameter correspond to arc-length).

2. Write a Manipulate function which allows one to walk through the evolution of any elementary cellular automaton up to 20 steps starting from a single black cell. The size of the graphic should be constant.

3. Write a Manipulate function that allows one to examine the 2 state 2 color Mobile automata, with one parameter for the rule number.

[[Note ordinary mobile automata have one state for the active cell.]]

4. Write a function which does the same thing as Table[list[[{i,j}]],{i,Length[list]},{j,1,i-1}] without using Table, and a way to display the result. Then make a Manipulate to show it at work on different lists.

Basic ImageManipulation

1. Try mixing the parameters of Pruning and ImageResolution to find new ways to discover features in ECA evolutions. Begin by empirically finding a function f[t] so that

```
New30[t_] :=
 Show[Pruning[Image[ArrayPlot[CellularAutomaton[30, {{1}, 0}, t], Frame → None],
    ImageResolution → f[t]], Infinity], ImageSize → 300]
```

shows (for a large range of integer values t) structures similar to

```
Show[Pruning[Image[ArrayPlot[CellularAutomaton[30, {{1}, 0}, 100], Frame → None],
   ImageResolution → 43], Infinity], ImageSize → 300]
```

2. Use Colorize to visualize the GCD function. Try visualizing it with ArrayPlot and these Binarize functions: Binarize, ChanVeseBinarize, MorphologicalBinarize, and RegionBinarize.

3. Find (or combine) image manipulation functions to search for structures in 2D CellularAutomata.

Basic Data

The following questions involve getting data, either using a data paclet or from the web, and calculating with it. Feel free to try out variations on the theme. Warning: you will need to check to make sure your data is clean, without unexpected or bizarre values.

1. Investigate the 100 brightest stars.

a.) Make a list, e.g. using AstronomicalData["StarBrightest100"] and eliminate the "Sun" from that list.

b.) Filter out those where the “Mass” is not known. Find the values for {DistanceLightYears,Declination,RightAscension,Mass,AbsoluteMagnitude} and make scatter plots. For instance for distance as a function of mass, make a Graphics with Points where the x values are mass and the y values are distance. Use appropriate options to make it presentable.

c.) Make some hypotheses and test them. Some patterns are in there, so use Fit or a related function to model them and check your errors. Maybe some look random, so try testing their randomness.

2. Investigate carbon and hydrogen molecules.

a.) Find a list of all known molecules that only contain carbon and hydrogen, e.g. using ChemicalData.

b.) Make a scatterplot showing the number of carbons against the number of hydrogens.

c.) Find a sublist for which the data is available (i.e. numbers) on melting point, density and combustion heat.

d.) Make scatter plots for pairs of these numbers, including the count of carbon atoms and hydrogen atoms. Find patterns and relationships. Find examples which don’t fit the patterns.

3. Investigate common phrases, idioms and clichés in a text corpus.

a.) Find some text as a string, either from ExampleData, or some other source such as an online data repository for literature.

b.) Use StringReplace, StringTake, or other similar functions to clean up your source so it is just the sequence of words.

c.) Use Tally, Gather, Parition or other similar functions to find common phrases.

d.) Try this with other sources and compare. What about text on webpages or blogs or something else? What are the most common things for people to say?

4. Convert some information into the form of a data paclet. Ideally, something that you care about or use often. At the very least your XXXData function would be defined using a "Name" for each entity, and there would be also definitions for each property.

Problems

1. Find the rule number for the identity rule, the rule which keeps any input exactly the same for (k=2, r=1), the ECA's. Find the identity rule number for three colors, (k=3, r=1).

2a. Determine which of the first hundred 3-color totalistic Cellular Automata are not ultimately repetitive starting from the initial condition of a single gray cell, {{1},0}, by considering only the first 100 steps of their evolutions.

b. Of those, determine how fast their pattern grows for the first hundred steps. For several of these a numerical value can only be approximate.

c. Find which one of the interesting first 100 totalistic 3 color rules dies out eventually.   Find those whose patterns grow but are essentially cyclic.

d. How many distinct evolutions are there starting from the single gray cell among the first 100 rules?

3. Find the number of occurrences of {0,1,1,0,1} in each step of

```
CellularAutomaton[30, {{1}, 0}, 100]
```

use

```
ListLinePlot
```

to display your result.

4. Find the subsequences of length 6 which occur and the number of each occurrence for rule 110 after evolving for 40 steps, starting with {{1},0}. Give your answer by representing the subsequences using

```
FromDigits
```

5. Write a function which takes the cyclical union of binary lists, where two lists are considered the same if they can be cyclically permuted to be the same, say using

```
RotateRight
```

Consider different possibilities without using the function

```
Union
```

using invariants under rotation, and test their performance using

```
Timing
```

6. Find all possible tilings ways to fit four copies of the following tile in a 7 by 7 grid, without any holes. Make a graphics function to display the results.

```
{{1, 0, 1, 0}, {1, 1, 1, 0}, {0, 0, 1, 1}}
```

Here are some rough suggestions for visualization.

```
ArrayPlot[{{1, 0, 1, 1}, {1, 1, 1, 0}, {0, 0, 1, 1}}, Mesh → True]
```

```
Graphics[{Green, Polygon[{{0, 1}, {2, 1}, {2, 0}, {4, 0}, {4, 1},
    {3, 1}, {3, 2}, {4, 2}, {4, 3}, {2, 3}, {2, 2}, {1, 2}, {1, 3}, {0, 3}}],
  Red, Line[{{0, 1}, {2, 1}, {2, 0}, {4, 0}, {4, 1}, {3, 1}, {3, 2},
    {4, 2}, {4, 3}, {2, 3}, {2, 2}, {1, 2}, {1, 3}, {0, 3}, {0, 1}}]}]
```

7. Find the minimal representations of the first 30 numbers in terms of the function h and the primitives 0 and 1.

```
h[x_, y_] := x^2 - y + 1
```

8. Given a triangle, write a function which splits it into three triangles, using the center as the average of the three vertices (centroid). Write a function to display the results after 4 steps starting from an equilateral triangle.

9. Find a boolean expression for Nand using rule 45 with the following function by finding an appropriate answer, a sequence of 0 and 1:

```
nand[x_, y_] := CellularAutomaton[45, Join[{x}, answer, {y}], 3][[-1, 1]]
```

Which elementary cellular automata can give Nand? Try generalizing the above function, perhaps with a completely different structure. What about larger logical operators, such as.,

```
Nand[Nand[#, #2], Nand[#3, #4]] &
```

10. Find the ECA which have inverted white triangles in a typical evolution starting from random initial conditions. Find those with inverted black triangles. Count the number of inverted white triangles in rule 30 after 100 steps.

11. Use ArrayPlot to show 20 steps of every third totalistic CA with a white background with codes 900 to 921.

12. Implement a Tag system function which takes a tag rule in the form

```
{{0, 0, x___} → Flatten[{x, replacement00}],
 {0, 1, x___} → Flatten[{x, replacement01}],
 {1, 0, x___} → Flatten[{x, replacement10}],
 {1, 1, x___} → Flatten[{x, replacement11}]}
```

where the replacements are lists of 0's and 1's. Enumerate rules of this form, make rule icons, and evolution graphics. Find the first number (under your enumeration) whose Tag system evolution from the simplest initial condition {0,0} which is not essentially cyclic.

13. Find a sequence of replacements using the rule

```
{"AB" → "A", "AAAB" → "BAB", "ABB" → "AABB"}
```

that takes "ABB" into "BABAB".

14. Make a Graphics showing the base 2 digits of the rule

```
x → (x + 2 / x) / 2
```

iterated from x==1. (Say 40 digits for ten steps)

15. Write a function to compute the number of connected components of a binary two dimensional array with cyclic boundary conditions, where two black squares are connected if they are adjacent (diagonals don't count). Compute the number of components of the evolution of code 746 (outer totalistic with 8 neighbors) from p.178, with the initial condition

```
{{{1, 1, 1, 1, 1, 1, 1}}, 0}
```

and code 6 (totalistic with 4 neighbors, e.g. p. 247) from a random 40 by 40 binary array.

16. Find a packing of 10 circles, five of radius 1 and five of radius 1/2, using

```
NMinimize
```

to attempt to minimize the total of the distances of their centers from {0,0}. Make a Graphics. Try alternative functionals, e.g., the sum of the distances from the centers of different sized circles. (Compare with the procedure from p.350)

17. Implement a Random walk in one dimension, which is the accumulated sum of random directions -1 and 1. Display the results. Do the same but where the directions are randomly chosen from square grid directions in 2D, and then do it on a hex grid. Implement a random walk through the space of totalistic CA's, displaying their evolutions from a basic initial condition {{1},0} (compare p.391)

18. Find the rule numbers for r=3/2,k=2 which preserve the number of black cells (compare p.458). Use the following CA function

```
CellularAutomaton[{rn, 2, {{-1}, {0}, {1}, {2}}}, ic, 20]
```

19. Find all expressions involving the binary operator f and the primitive term 0 which satisfy the constraint with fewer than six f's.

```
2 == # /. f → Function[{a, b}, Mod[a - b + 2, 3]] &
```

20. Apply run length encoding to each row of rule 30, run for 15 steps, finding the lengths of alternating runs of 0's and 1's (compare p.562 and related code in the Notes)

21. Find the rule number for the three color CA which emulates ECA rule 30 by treating every 2 like a 0.

22. Write a function that takes a list of circles, creating new circles inside of each. For each Circle[center, radius], it should produces *n* new circles on the inside, tangential to the original circle, each

of radius a fixed ratio to the radius of the original circle. Note that the interior circles may overlap. Make this a function of the ratio and the number *n* of new circles. Use this to produce the nested evolution after four steps.

23. Consider the following types of rules (see p.609). A rule which accepts two arguments: the first argument is a state which can be 0,1, or 2 and the second argument is an input which can be either {0,0}, {0,1}, {1,0} or {1,1}. The output is a state, either 0,1,or2.

a) How many distinct rules are there that match the specification?

b) Write a function which takes a rule number and produces a rule.

c) Write a function which makes such a rule act on two binary lists of equal length, a Fold operation where the state changes according to the rule in which the second argument is the pair of binary (0 or 1) numbers from the same position in the two lists. Take the initial state to be 0.

d.) Display the output of the rules acting on the x,y coordinates written in base 2, say on an array of size 64 by 64. Try for instance the 36118th rule (with no convention, it is possible a different numbering scheme will behave differently).

e.) Make some measurements on a random sample to estimate the proportion of such rules which give a flip symmetry, and other sorts of symmetries. (Possibly find more optimal ways of doing this.)

24. Every CA in a cyclic background is eventually cyclic in time. For rule 30 what are the possible periods for a cyclic background of size 7 ? Display the counts in a bar chart.
Automate the process of finding the lengths of periods nad answer the same question for all of the 256 elementary cellular automata.

25. Partition the ECA's according to their behavior on a single black cell.

26. Find the possible behaviors of a Turing machine with 2 states and 2 colors on a blank tape.

27. Make an array plot of a binary array which approximates a line with of slope 5/7.

28. (p.82) Implement the string substitution system "AAA"→"ABA" using StringReplace.

Implement the updates so that only the left-most instance gets updated.

29. Find all cyclic initial conditions up to size 15 for rule 30 which give the first 12 terms of the Thue-Morse sequence {1,0,0,1,0,1,1,0,0,1,1,0} as the first column in the first 11 steps of their evolution following rule 30.

30. Given a list of integer coordinates, make a Graphics object which draws squares at the coordinates and lines connecting consecutive squares, so that the lines don't enter the squares. The default would be to appear to pass under the interior of the squares.

31. (p.69,p.1017)For totalistic rule 1599 (3 color) , what can you say about which states are reversible for one step? A state s1=={0,0,1,2,1,...} is reversible for one step if there is exactly one state s0 for which

```
CellularAutomaton[{1599, {3, 1}, 1}, s0, 1, -1] == {s1}
```

32. (p.250) Show the difference patterns by altering the evolution of CellularAutomaton[30,IntegerDigits[#,2,n],11]

```
CellularAutomaton[30, IntegerDigits[12345, 2, 101], 50]
```

at a single random position during the evolution. Collect a sample of these patterns. Make an ArrayPlot where the cell is 0 for originally 0, 1 for originally 1, 2 for an altered original 0, 3 for an altered original.

33. Implement the 2D totalistic CA (9 neighbor) rule 998 where at each step any update which changes the center cell's color has a 50% chance of not happening.

34. In a system under going the Thue-Morse substitution

```
{"A" -> "AB", "B" -> "BA"}
```

what is the rule that the run length encodings follow?

35. Implement the 1D rule on lists of numbers that takes the average of a number and its two neighbors. Evolve the rule with cyclic boundaries on

```
{0, 0, 0, 1.1, 2.5, .9, 0, 0, 0}
```

and display the results after 10 steps using ArrayPlot with Mesh->True.

36. (p.278) Use GraphPlot to show the finite automata that specifies which states can occur after three steps of rule 73.

37. Find the string substitution rules lhs→rhs which transforms

```
"AAAAAAAA"
```

into

```
"BBBBBBAABA"
```

by repeating the following 10 times

```
StringReplace[#, {"AB" -> "BAA", lhs → rhs}] &
```

38. What can one say about the evolution if the squaring CA in the book (p639,p.1109) has a column that begins

```
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 6, 1, 6, 1, 6, 1, 1, 6, 1, 6, 1, 6, 1, 6,
 1, 1, 6, 1, 6, 1, 6, 1, 6, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}
```

for the first 60 steps.

```
SquareCARule = {{0, _, 3} → 0, {_, 2, 3} → 3, {1, 1, 3} → 4,
    {_, 1, 4} → 4, {1 | 2, 3, _} → 5, {p : (0 | 1), 4, _} :→ 7 - p, {7, 2, 6} → 3,
    {7, _, _} → 7, {_, 7, p : (1 | 2)} :→ p, {_, p : (5 | 6), _} → 7 - p,
    {5 | 6, p : (1 | 2), _} :→ 7 - p, {5 | 6, 0, 0} → 1,
    {_, p : (1 | 2), _} :→ p, {_, _, _} → 0};
```

39. (p.127) Find the simplest expression rhs with the primitives $\{n, -1\}$,with functions $\{f, \text{Plus}\}$ that has the following behavior

```
Clear[f]
```

```
f[n_] := f[n] = rhs; f[1] = f[2] = 1;
```

```
Array[f, 25] ==
 {1, 1, 2, 2, 3, 4, 4, 4, 5, 6, 7, 7, 8, 8, 8, 8, 9, 10, 11, 12, 12, 13, 14, 14, 15}
```

40. (p.102,p.488) Deduce the causal structure of the following symbolic system from p.103 and display it with GraphPlot.

```
e[x_][y_] → x[x[y]]
```

starting from

```
e[e[e][e]][e][e]
```

41. For the system in the above problem, rewrite it as a graph update and use another evaluation order (see BreadthFirstScan and DepthFirstScan) then compute its causal network. Try other evaluation rules.

42. Find the number of characters used in your answers to the above questions. Usually shorter answers give more efficient code. For comparison, the number of characters in the notebook I was using was 20509.