In your own words, what is a Lindenmayer system (L-System)?  Your definition and description should be complete.

An L-System (deterministic and zero context) is a starting string (state), a rule by which the string is transformed, and every subsequent state of the string. It is a variant of a formal grammar, or set of rules for forming strings in a language/syntax. This means that the L-System, like written language, is meaningless without interpretation, but also follows a set of rules as to its formation. Because of this, excepting probability where applicable (ie. Non-deterministic L-Systems), an L-System will always adhere to a pattern based on the rule and seeded with an initial state. One variation of L-Systems include non-deterministic, which allows for a degree of randomness into the L-System by probabilistically following one rule or another (or another, etc). Another variation is the context-sensitive L-System which functions like a stack in that you can push and pop instructions to make a more complicated, non-linear L-System. These variations are combinable, and this is by no means a complete list of the variations of L-Systems.

Write a pseudocode algorithm to generate a simple deterministic zero-context L-System.  You pseudocode should be clear and concise.

Transform (current, rules[])

{

for each char in current

for each rule in rules

if rule[0] == char

nextString.Append(applyRule(char, rule))

break

if still in loop

nextString.Append(char)

return nextString

}

In your own words, what is a cellular automaton (CA)? Your definition and description should be complete.

A CA is similar to an L-System, in that it has an initial, seed state and a rule. A CA’s rule is more complicated, in that it must describe the outcome of all possible states of a cell and its neighbors. For example, in a 5 neighbor system, which is to say, a cell and the cells above and below, left and right of it, there are 32 states. 16 for all configurations of the neighbors (on or off), doubled for the cell’s own state (on or off). Rules for 1-D CA’s are often expressed as numbers, such as Wolfram’s Rule 34, which is binary 0100010 and describes the various states possible in a 3 neighbor 1-D CA and their outcomes. Some CA, such as Conway’s Game of Life, have a rule which is flexible enough to be expressed more simply. Conway’s GoL specifies that if a “dead” cell has exactly three neighbors becomes “alive”, and each “live” cell which has less than one or more than four neighbors dies. This is simply a complicated rule set as described above, but for 9 neighbors (512 states).

Write a pseudocode algorithm for Conways Game of Life CA.  You pseudocode should be clear and concise.

CreateNextGrid (grid)

{

for each cell in grid

if cell is dead

if countNeighbors(cell) == 3

newGrid[cell] is alive

else if countNeighbors(cell) < 2 or > 4

newGrid[cell] is dead

else

newGrid[cell] is the same as grid[cell]

}

In your own words, what is a genetic algorithm (GA)?  Your definition and description should be complete.

A genetic algorithm is a programmatic model of evolutionary changes over generations. The elements are a genome, made up of genes, a mutation rate, parents, children, and fitness ratings. The idea is to take a seed population, which can be randomly generated, and create from that a set of genomes (50, for example) which are the original population copied with a mutation rate applied (in a single-parent system). The top members of each generation are held over without the mutation rate to the next generation, like clones, which prevents degradation of the entire population. The genome represents the solution domain, and so can for example be a string (where genes are characters), a section of code (such as a function), a recipe (<http://xkcd.com/720/>), or any other system which contains elements and can be modeled in this way. Genes are the elements which can be randomized and make up the genome (ie. elements of an array). The population is the set of all genomes in the current generation.

Write a pseudocode algorithm for a generic GA.  You pseudocode should be clear and concise.

CreateNextGeneration(population)

{

copy most fit elements from previous population

for the rest of the size of the population

{

compare two parents for fitness

apply the mutation rate to the one with higher fitness

add the resultant child to the new population

}

return the new array

}

In your own words, what is an agent based model?  Your definition and description should be complete.

An agent based model is one where instead of using a traditional top-down approach to AI, where everything is controlled by an “overmind”, each member of the model (where member can mean the ants in an ant colony, the birds in a flock, or the predators and prey in a predation model, to name a few) is a self-contained class which acts independently based on its surroundings. Each agent acts based on the same instruction set, but because it has a limited perspective (agents are limited in their perception range. They don’t know everything about the world.), it performs differently than the others. This leads to emergent behaviors: agents do things that aren’t specifically programmed in because they are reacting to both changes in the environment and to the other agents.

Act {

examineSurroundings()

analyzeSituation()

react()

}

The pseudocode here is very generic. Movement, attacking, eating, running, etc would all be reactions, while examine surroundings would be pinging the world for its immediate surroundings (range defined by the code), and analyze situation would be deciding if it should run from a predator, eat some food, follow pheromone, move randomly, etc. It is difficult to write something which makes sense in a specific use case so generically, so there you have it.

Write a pseudocode algorithm for generic agent based model.  You pseudocode should be clear and concise.  
In your own words, what does the future hold for A-Life?  What applications are there, or will there be for this discipline?  Your answer should be clear and concise, and your assretoins should be supported.  
  
2 paragraphs give or take each.  
Pseudocode can probably be a single function.