Acceptors

Data in the real world has structure. The structure arises because the world is not random, but has patterns that reflect an organizing principle inherent to the behavior of the world. The structure may be very fine grain (micro structure), at the level of individual data items, or may be very coarse grain (macro structure), involving a complex set of relationships among many individual data items. This study will focus on fine grain structure. The lessons and ideas apply equally well to larger, more complex structures.

Two examples illustrate the idea of fine grain structure in data. The first example of fine grain structure is the common representation of US currency, which (for our purposes) includes all of the following:

$.75
$20.50
$1,200.99
$8,245,679.13

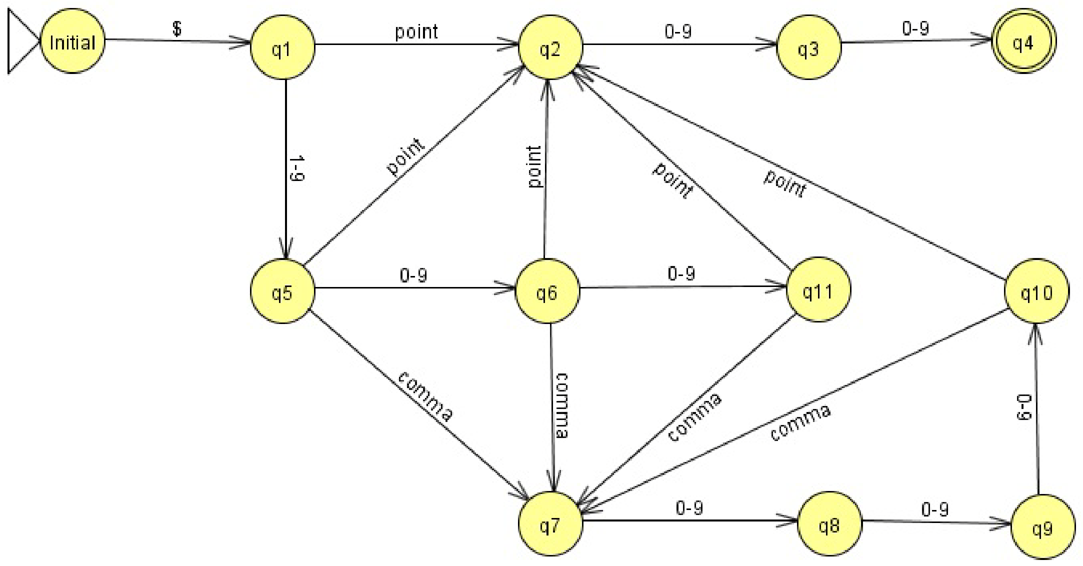
but none of the these:

.75 (missing initial "$")
$1,200.4 (decimal point not followed by two digits)
$1,200 (missing decimal point and two following digits)
$20.513 (more than two digits after the decimal)
$8,20.15 (comma not followed by three digits)
$,200.77 (no digit preceding the comma)
$0,333.44 (first digit cannot be a zero)

A second example of small scale structure in data is that of the genetic code. All living organisms have a DNA structure that can be represented as a sequence of exactly four letters: T, C, A, and G. The structure and function of plants, animals, and people derive from how this simple genetic code is used to construct genes. The genes are the basic building blocks of life. However, not every sequence of the four letters represents a gene. Every gene, however, has a pattern based on codons (three consecutive letters). In this pattern the gene always begins with the start codon ATG, ends with one of three end codons (TAA, TAG, and TGA), and in between has some sequence of codons. To be clear, every gene has this structure but there may be some structures that have this pattern that are not genes. Thus, the pattern is a necessary, but not a sufficient, condition for identifying a gene.

Finite state machines can be used as acceptors – machines that recognize if a given input satisfies a given pattern. The finite state machine will have one "initial" state and one or more "final" states. The transitions between states are labeled with a symbol that occurs in the input. An input is accepted or recognized when the finite state machine is in a "final" state at the end of the input assuming that the finite state machine began in the "initial" state. The initial state is designated by an unlabelled arc entering it that does not

come from any other state (or some equivalent notation). A final state is typically drawn as two concentric circles (a non-final state is usually drawn as a simple circle).
An acceptor for simple US currency values is shown in the figure below.



Acceptor for US Currency Values

Note that the initial state in this drawing is indicated by a triangle pointing to a state that is named “Initial”. The state named “q4” is the single, final state. Except for the initial state, the names of the states (q1, q2,…) are not inherently meaningful. The transitions labeled “point” refers to a decimal point. The transitions labeled 0-9 mean that any single digit in the range 0 through 9 will cause the transition to be made to the next state.

Acceptors are useful in a variety of contexts. One immediate use is to check that inputs provided to a system are valid. For example, a web-based banking service needs to check that amounts entered (to transfer to other accounts) are valid currency values. Gene acceptors are useful to biologists screening genomic data in the search for genes. See the assignment for an acceptor for a gene acceptor.

Exercises:

1. Test the acceptor using each of the valid and invalid US currency values given above.
2. Revise the acceptor so that the decimal point and the two digits after the decimal point are optional. That is, $1,200 is a valid currency value.

3. Revise the acceptor so that the initial “$” is optional. That is, 1,200.54 is a valid currency value.
4. Revise the acceptor so that any number of digits may follow the decimal point (e.g., for averages or small fractional amount). This is, $1.3456 is a valid currency value.