EXCERPTED FROM

STEPHEN WOLFRAM
A NEW KIND OF SCIENCE

SECTION 5.5

Network Systems

Network Systems

One feature of systems like cellular automata is that their elements are always set up in a regular array that remains the same from one step to the next. In substitution systems with geometrical replacement rules there is slightly more freedom, but still the elements are ultimately constrained to lie in a two-dimensional plane.

Indeed, in all the systems that we have discussed so far there is in effect always a fixed underlying geometrical structure which remains unchanged throughout the evolution of the system.

It turns out, however, that it is possible to construct systems in which there is no such invariance in basic structure, and in this section I discuss as an example one version of what I will call network systems.

A network system is fundamentally just a collection of nodes with various connections between these nodes, and rules that specify how these connections should change from one step to the next.

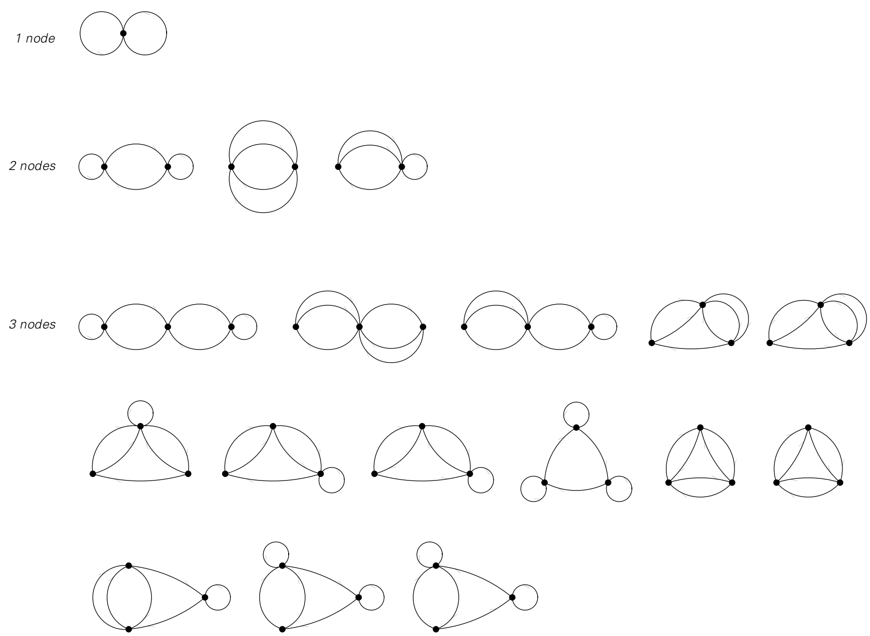
At any particular step in its evolution, a network system can be thought of a little like an electric circuit, with the nodes of the network corresponding to the components in the circuit, and the connections to the wires joining these components together.

And as in an electric circuit, the properties of the system depend only on the way in which the nodes are connected together, and not on any specific layout for the nodes that may happen to be used.

Of course, to make a picture of a network system, one has to choose particular positions for each of its nodes. But the crucial point is that these positions have no fundamental significance: they are introduced solely for the purpose of visual representation.

In constructing network systems one could in general allow each node to have any number of connections coming from it. But at least for the purposes of this section nothing fundamental turns out to be lost if one restricts oneself to the case in which every node has exactly two outgoing connections—each of which can then either go to another node, or can loop back to the original node itself.

With this setup the very simplest possible network consists of just one node, with both connections from the node looping back, as in the top picture below. With two nodes, there are already three possible patterns of connections, as shown on the second line below. And as the number of nodes increases, the number of possible different networks grows very rapidly.

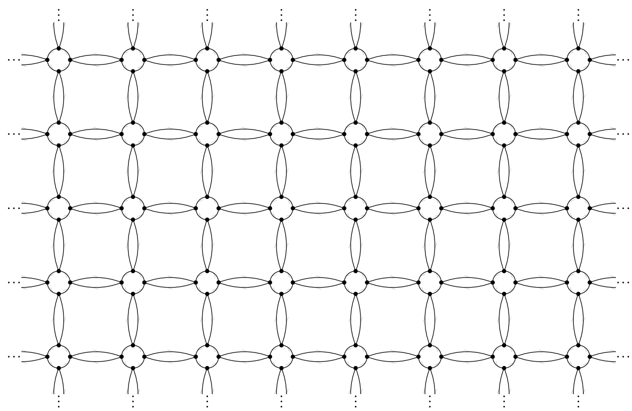


Possible networks formed by having one, two or three nodes, with two connections coming out of each node. The picture shows all inequivalent cases ignoring labels, but excludes networks in which there are nodes which cannot be reached by connections from other nodes.

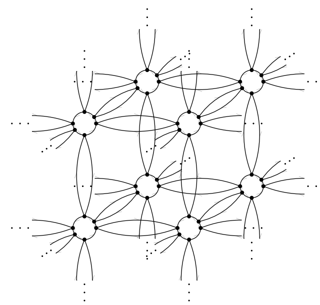
For most of these networks there is no way of laying out their nodes so as to get a picture that looks like anything much more than a random jumble of wires. But it is nevertheless possible to construct many specific networks that have easily recognizable forms, as shown in the pictures on the facing page.

Each of the networks illustrated at the top of the facing page consists at the lowest level of a collection of identical nodes. But the remarkable fact that we see is that just by changing the pattern of

(a) one dimension



(b) two dimensions



(c) three dimensions

Examples of networks that correspond to arrays in one, two and three dimensions. At an underlying level, each network consists just of a collection of nodes with two connections coming from each node. But by setting up appropriate patterns for these connections, one can get networks with very different effective geometrical structures.

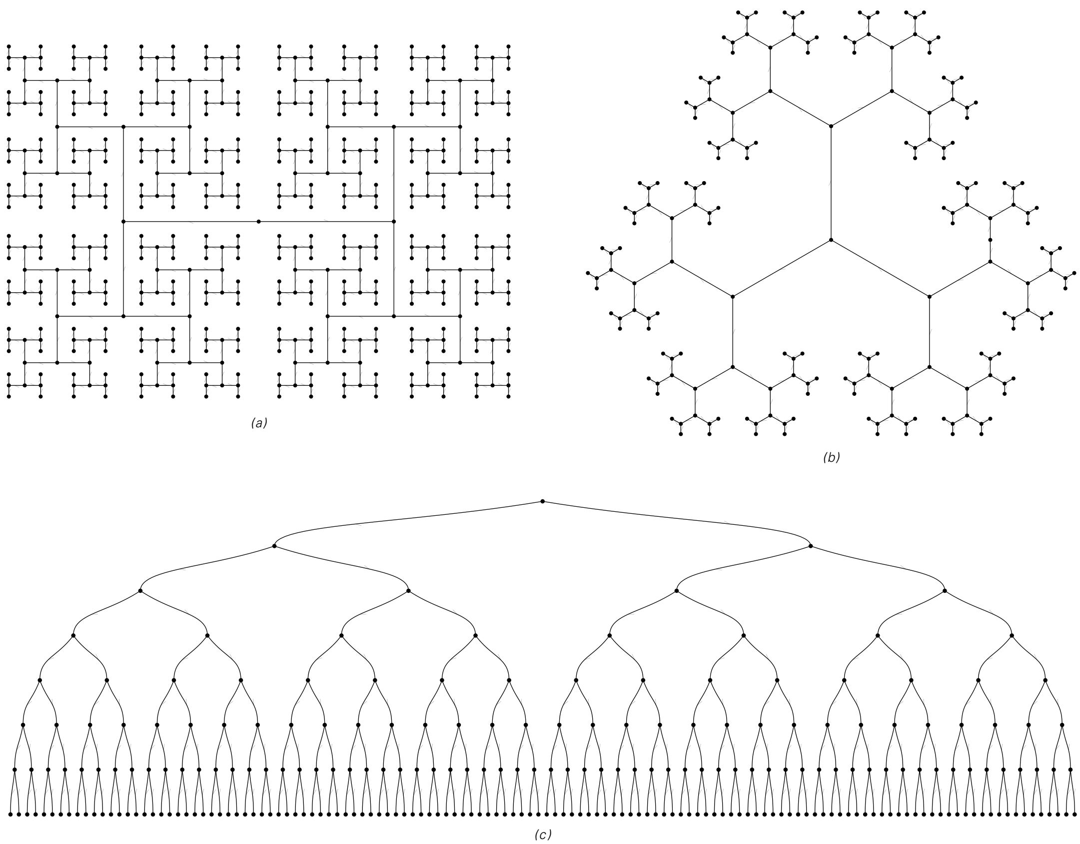
connections between these nodes it is possible to get structures that effectively correspond to arrays with different numbers of dimensions.

Example (a) shows a network that is effectively one-dimensional. The network consists of pairs of nodes that can be arranged in a sequence in which each pair is connected to one other pair on the left and another pair on the right.

But there is nothing intrinsically one-dimensional about the structure of network systems. And as example (b) demonstrates, it is just a matter of rearranging connections to get a network that looks like a two-dimensional rather than a one-dimensional array. Each individual node in example (b) still has exactly two connections coming out of it, but now the overall pattern of connections is such that every block of nodes is connected to four rather than two neighboring blocks, so that the network effectively forms a two-dimensional square grid.

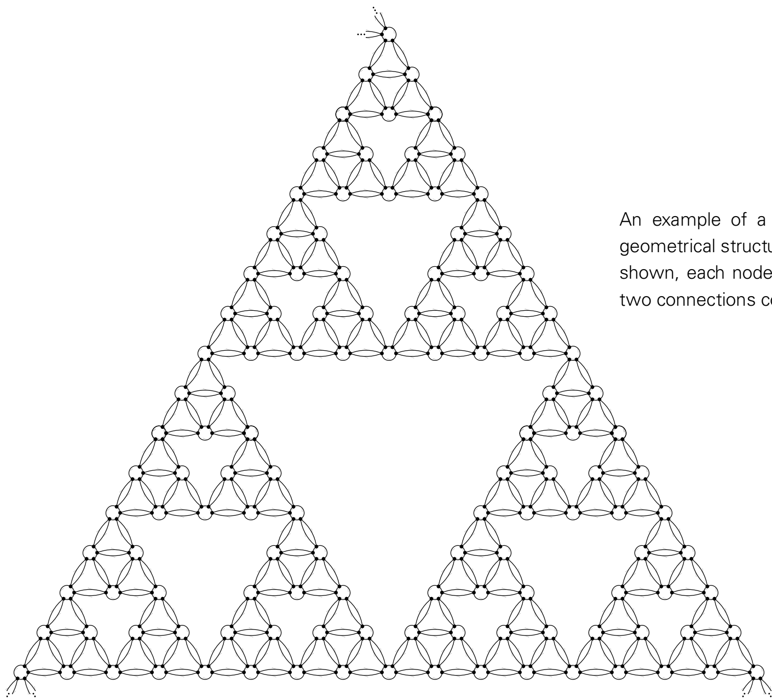
Example (c) then shows that with appropriate connections, it is also possible to get a three-dimensional array, and indeed using the same principles an array with any number of dimensions can easily be obtained.

The pictures below show examples of networks that form infinite trees rather than arrays. Notice that the first and last networks shown actually have an identical pattern of connections, but they look different here because the nodes are arranged in a different way on the page.



Examples of networks that correspond to infinite trees. Note that networks (a) and (c) are identical, though they look different because the nodes are laid out differently on the page. All the networks shown are truncated at the leaves of each tree.

In general, there is great variety in the possible structures that can be set up in network systems, and as one further example the picture below shows a network that forms a nested pattern.

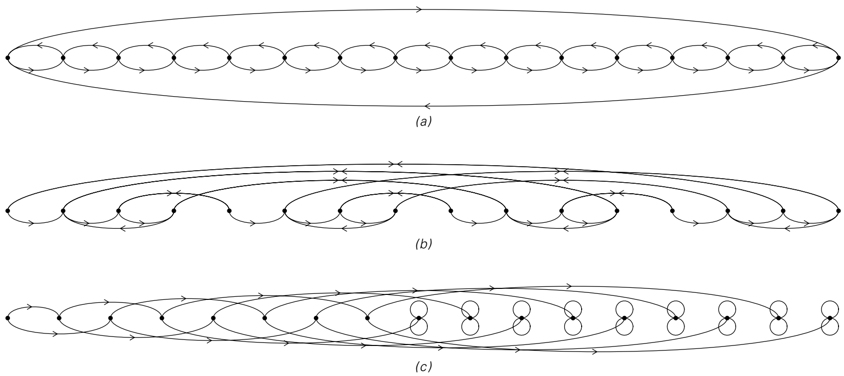


An example of a network that forms a nested geometrical structure. As in all the other networks shown, each node here is identical, and has just two connections coming out of it.

In the pictures above we have seen various examples of individual networks that might exist at a particular step in the evolution of a network system. But now we must consider how such networks are transformed from one step in evolution to the next.

The basic idea is to have rules that specify how the connections coming out of each node should be rerouted on the basis of the local structure of the network around that node.

But to see the effect of any such rules, one must first find a uniform way of displaying the networks that can be produced. The pictures at the top of the next page show one possible approach based on always arranging the nodes in each network in a line across the page. And although this representation can obscure the geometrical structure



Networks from previous pictures laid out in a uniform way. Network (a) corresponds to a one-dimensional array, (b) to a two-dimensional array, and (c) to a tree. In the layout shown here, all the networks have their nodes arranged along a line. Note that in cases (a) and (b) the connections are arranged so that the arrays effectively wrap around; in case (c) the leaves of the tree are taken to have connections that loop back to themselves.

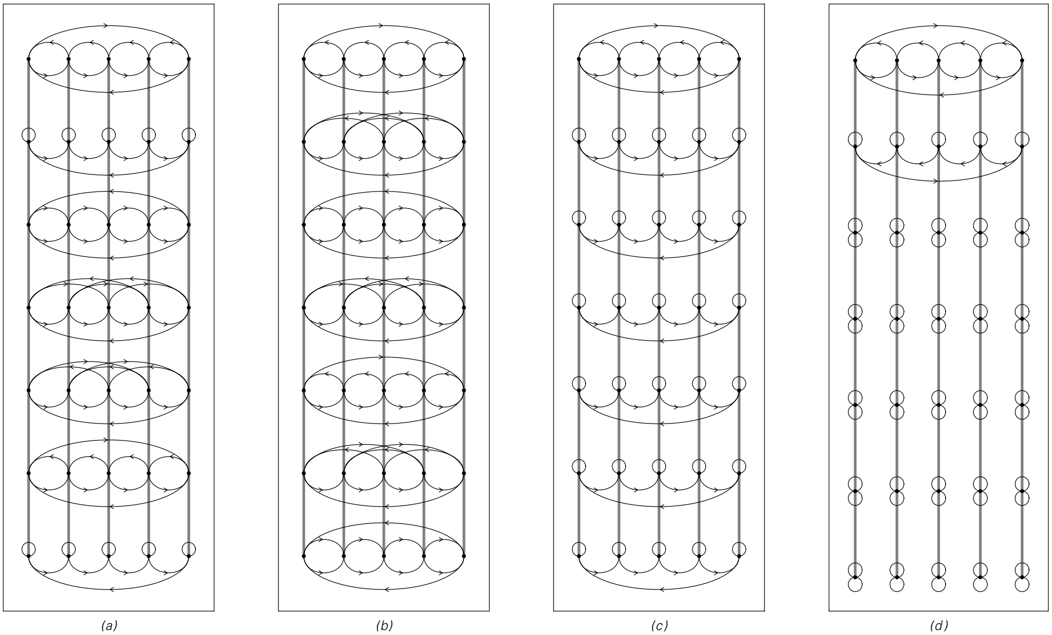
of a particular network, as in the second and third cases above, it more readily allows comparison between different networks.

In setting up rules for network systems, it is convenient to distinguish the two connections that come out of each node. And in the pictures above one connection is therefore always shown going above the line of nodes, while the other is always shown going below.

The pictures on the facing page show examples of evolution obtained with four different choices of underlying rules. In the first case, the rule specifies that the "above" connection from each node should be rerouted so that it leads to the node obtained by following the "below" connection and then the "above" connection from that node. The "below" connection is left unchanged.

The other rules shown are similar in structure, except that in cases (c) and (d), the "above" connection from each node is rerouted so that it simply loops back to the node itself.

In case (d), the result of this is that the network breaks up into several disconnected pieces. And it turns out that none of the rules I consider here can ever reconnect these pieces again. So as a consequence, what I do in the remainder of this section is to track only the piece that includes the first node shown in pictures such as those

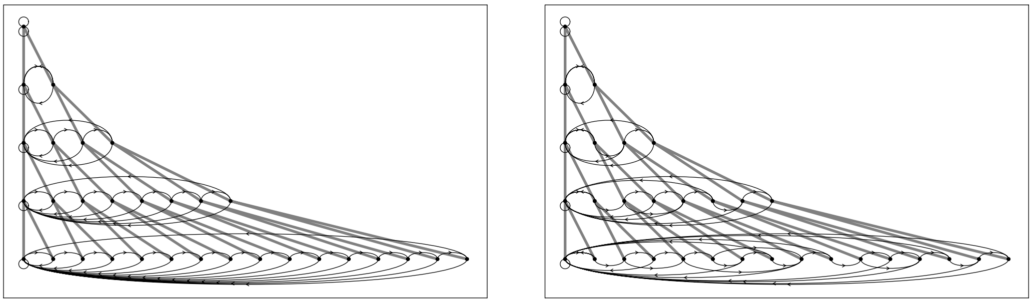


The evolution of network systems with four different choices of underlying rules. Successive steps in the evolution are shown on successive lines down the page. In case (a), the "above" connection of each node is rerouted at each step to lead to the node reached by following first the below connection and then the above connection from that node; the below connection is left unchanged. In case (b), the above connection of each node is rerouted to the node reached by following the above connection and then the above connection again; the below connection is left unchanged. In case (c), the above connection of each node is rerouted so as to loop back to the node itself, while the below connection is left unchanged. And in case (d), the above connection is rerouted so as to loop back, while the below connection is rerouted to lead to the node reached by following the above connection. With the "above" connection labelled as 1 and the "below" connection as 2, these rules correspond to replacing connections *{{1}, {2}}* at each node by (a) *{{2, 1}, {2}}*, (b) *{{1, 1}, {2}}*, (c) *{{}, {2}}*, and (d) *{{}, {1}}*.

above. And in effect, this then means that other nodes are dropped from the network, so that the total size of the network decreases.

By changing the underlying rules, however, the number of nodes in a network can also be made to increase. The basic way this can be done is by breaking a connection coming from a particular node by inserting a new node and then connecting that new node to nodes obtained by following connections from the original node.

The pictures on the next page show examples of behavior produced by two rules that use this mechanism. In both cases, a new node is inserted in the "above" connection from each existing node in the network. In the first case, the connections from the new node are exactly the same as the connections from the existing node, while in the second case, the "above" and "below" connections are reversed.



(a) (b)

Evolution of network systems whose rules involve the addition of new nodes. In both cases, the new nodes are inserted in the "above" connection from each node. In case (a), the connections from the new node lead to the same nodes as the connections from the original node. In case (b), the above and below connections for the new node are reversed. In the pictures above, new nodes are placed immediately after the nodes that give rise to them, and gray lines are used to indicate the origin of each node. Note that the initial conditions consist of a network that contains only a single node.

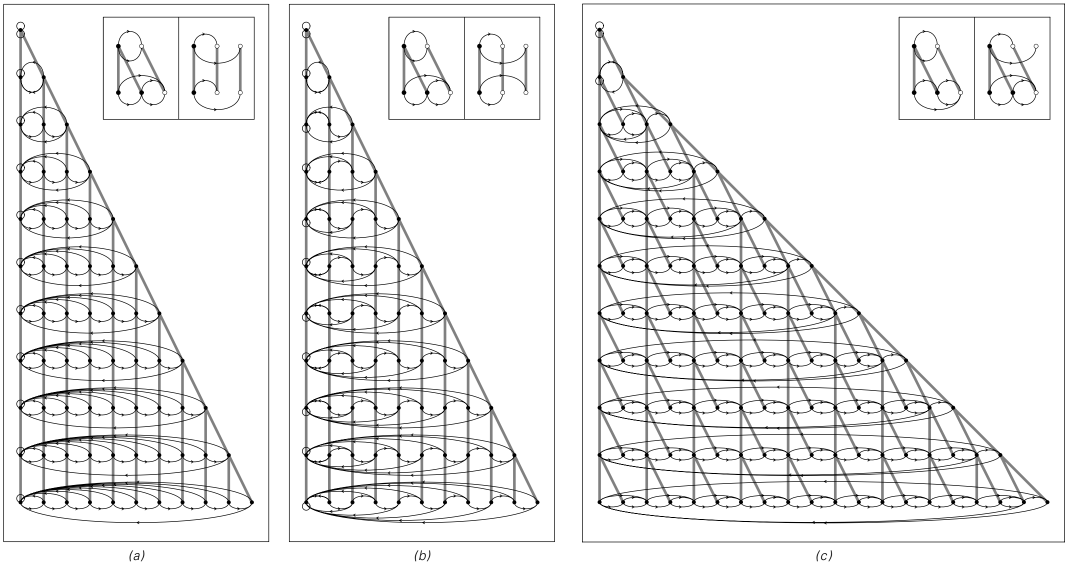
But in both cases the behavior obtained is quite simple. Yet much like neighbor-independent substitution systems these network systems have the property that exactly the same operation is always performed at each node on every step.

In general, however, one can set up network systems that have rules in which different operations are performed at different nodes, depending on the local structure of the network near each node.

One simple scheme for doing this is based on looking at the two connections that come out of each node, and then performing one operation if these two connections lead to the same node, and another if the connections lead to different nodes.

The pictures on the facing page show some examples of what can happen with this scheme. And again it turns out that the behavior is always quite simple—with the network having a structure that inevitably grows in an essentially repetitive way.

But as soon as one allows dependence on slightly longer-range features of the network, much more complicated behavior immediately

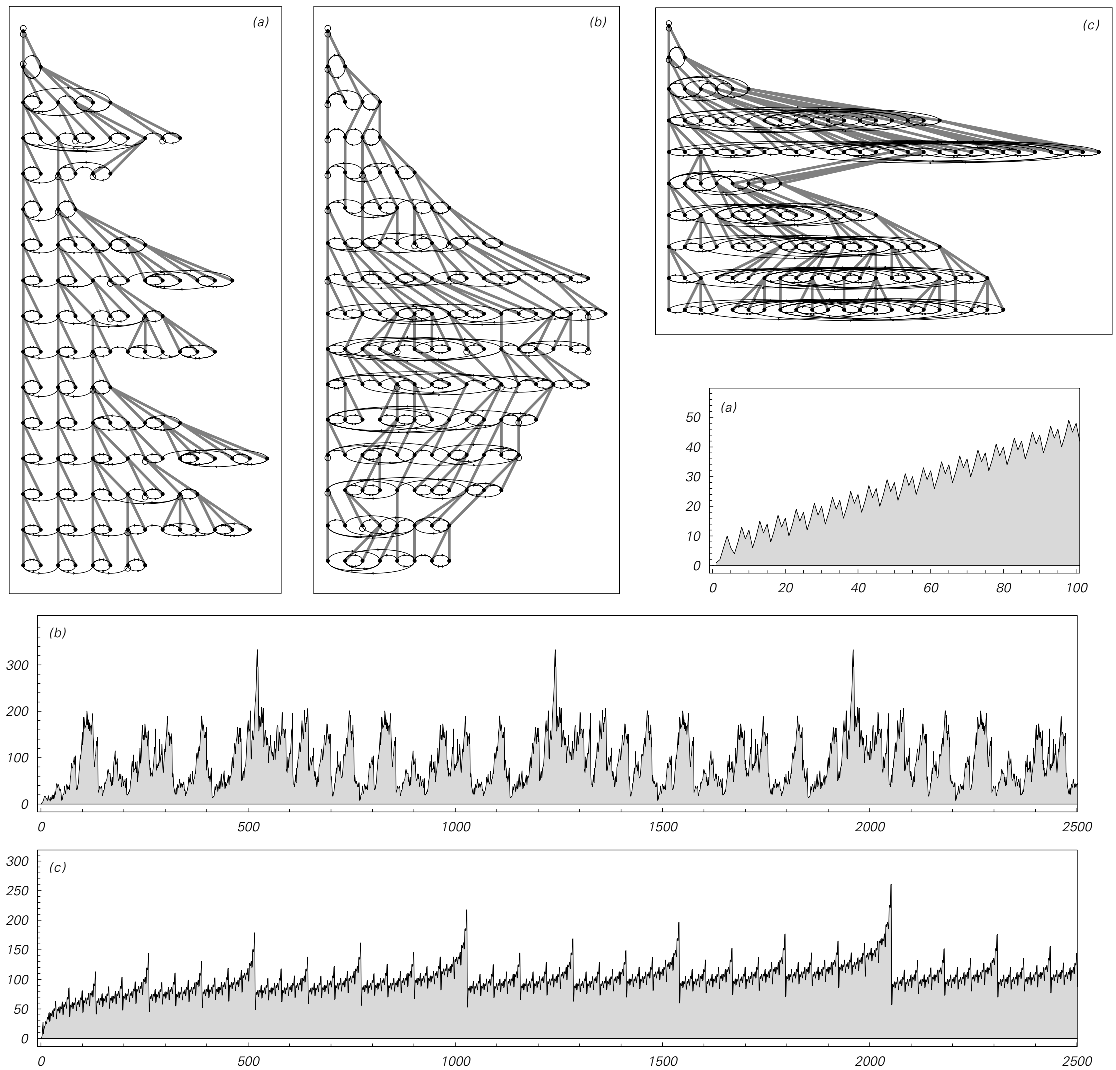


Examples of network systems with rules that cause different operations to be performed at different nodes. Each rule contains two cases, as shown above. The first case specifies what to do if both connections from a particular node lead to the same node; the second case specifies what to do when they lead to different nodes. In the rules shown, the connections from a particular node (indicated by a solid circle) and from new nodes created from this node always go to the nodes indicated by open circles that are reached by following just a single above or below connection from the original node. Even if this restriction is removed, however, more complicated behavior does not appear to be seen.

becomes possible. And indeed, the pictures on the next two pages show examples of what can happen if the rules are allowed to depend on the number of distinct nodes reached by following not just one but up to two successive connections from each node.

With such rules, the sequence of networks obtained no longer needs to form any kind of simple progression, and indeed one finds that even the total number of nodes at each step can vary in a way that seems in many respects completely random.

When we discuss issues of fundamental physics in Chapter 9 we will encounter a variety of other types of network systems—and I suspect that some of these systems will in the end turn out to be closely related to the basic structure of space and spacetime in our universe.

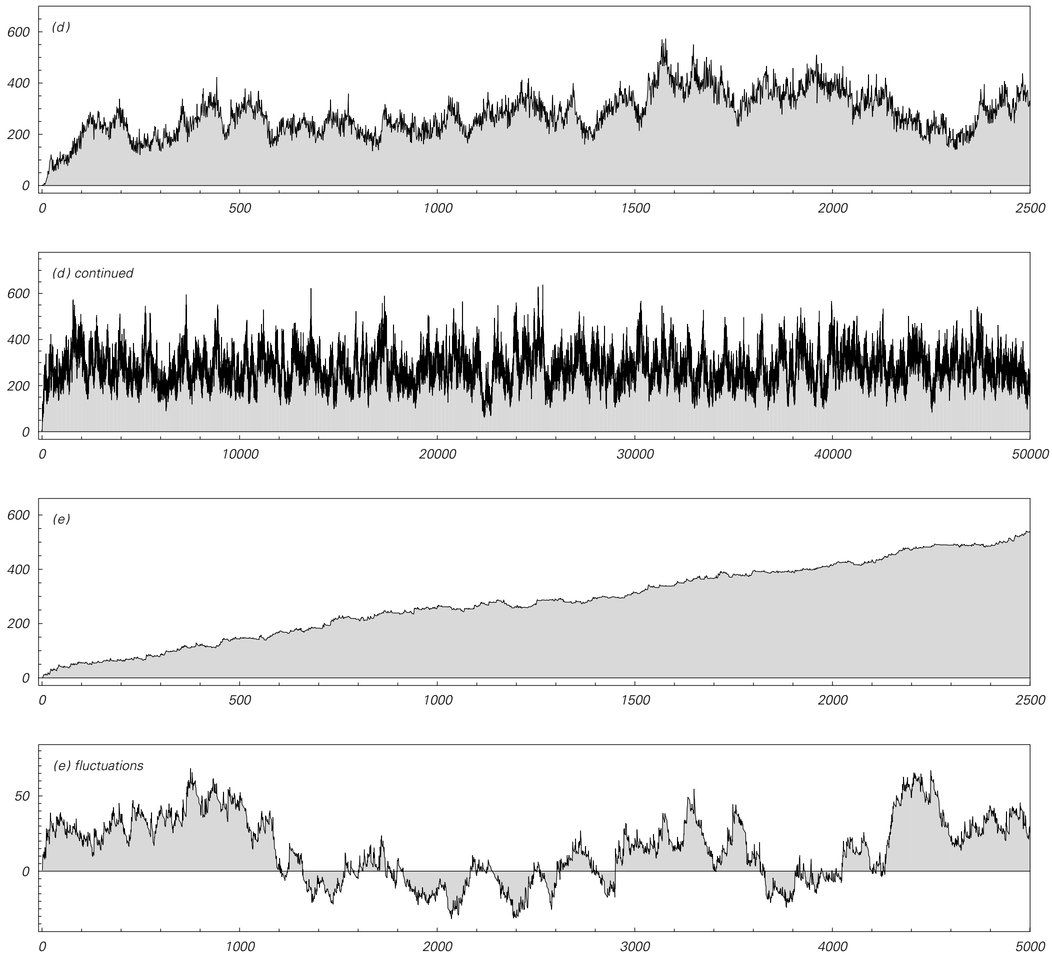


(a) {{1, 1}→{{1}, {{2, 1}, {2, 1}}}, {1, 2}→{{{}, {1, 1}}, {{1, 1}, {}}}, {2, 1}→{{{}, {}}, {{1}, {2, 1}}}, {2, 2}→{{{1, 1}, {2, 1}}, {{2}, {2, 1}}},
{2, 3}→{{{}, {}}, {2}}, {2, 4}→{{{2, 2}, {}}, {}}}

(b) {{1, 1}→{{{}, {1, 1}}, {2}}, {1, 2}→{{2}, {{}, {}}},
{2, 1}→{{2, 1}, {{}, {1}}}, {2, 2}→{{{2}, {1}}, {}}, {2, 3}→{{1, 2}, {2}}, {2, 4}→{{{1}, {1}}, {2, 1}}}

(c) {{1, 1}→{{{1, 1}, {1}}, {2}}, {1, 2}→{{{1, 2}, {2}}, {{2, 2}, {}}}, {2, 1}→{{{2, 2}, {2}}, {{1}, {}}}, {2, 2}→{{{1}, {1}}, {{2, 1}, {1, 1}}},
{2, 3}→{{2, 1}, {2}}, {2, 4}→{{{1}, {1, 2}}, {{1, 2}, {}}}}

Network systems in which the rule depends on the number of distinct nodes reached by going up to distance two away from each node. The plots show the total number of nodes obtained at each step. In cases (a) and (b), the behavior of the system is eventually repetitive. In case (c), it is nested—the size of the network at step t is related to the number of 1's in the base 2 digit sequence of t.



(d) {{1, 1} → {{{1, 2}, {1, 2}}, {}}, {1, 2} → {{2, 2}, {{1}, {1}}}, {2, 1} → {{1}, {{}, {2}}}, {2, 2} → {{1, 2}, {2, 1}}, {2, 3} → {{{2, 1}, {2}}, {1}}, {2, 4} → {{1}, {1, 1}}}

(e) {{1, 1} → {{}, {{1, 1}, {1, 2}}}, {1, 2} → {{{}, {1}}, {{1, 1}, {1, 2}}}, {2, 1} → {{2}, {}}, {2, 2} → {{{2, 1}, {1}}, {{1, 1}, {2}}}, {2, 3} → {{2, 2}, {2}}, {2, 4} → {{2, 1}, {2}}}

Network systems in which the total number of nodes obtained on successive steps appears to vary in a largely random way forever. About one in 10,000 randomly chosen network systems seem to exhibit the kind of behavior shown here.