EXCERPTED FROM

STEPHEN WOLFRAM
A NEW KIND OF SCIENCE

SECTION 9.10

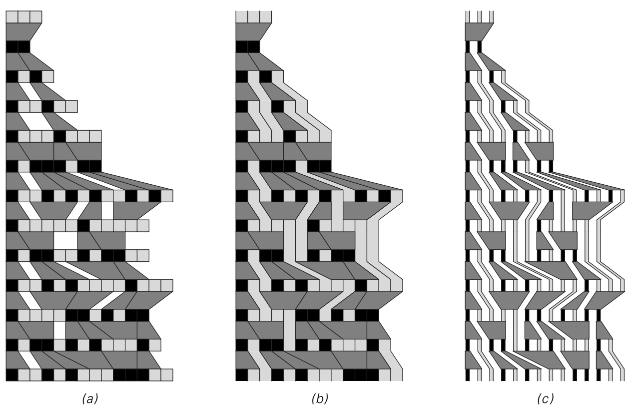
The Sequencing of Events in the Universe

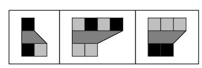
The Sequencing of Events in the Universe

In the last section I discussed one type of model in which familiar notions of time can emerge without any kind of built-in global clock. The particular models I used were based on mobile automata—in which the presence of a single active cell forces only one event ever to occur in the universe at once. But as we will see in this section, there is actually no need for the setup to be so rigid, or indeed for there to be any kind of construct like an active cell.

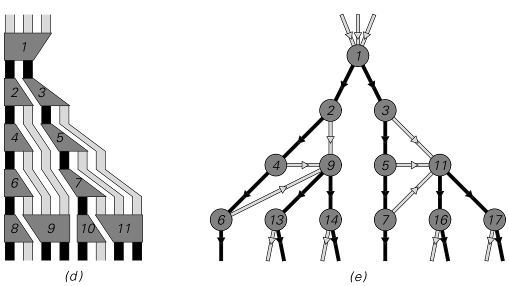
One can think of mobile automata as being special cases of substitution systems of the type I introduced in Chapter 3. Such systems in general take a string of elements and at each step replace blocks of these elements with other elements according to some definite rule.

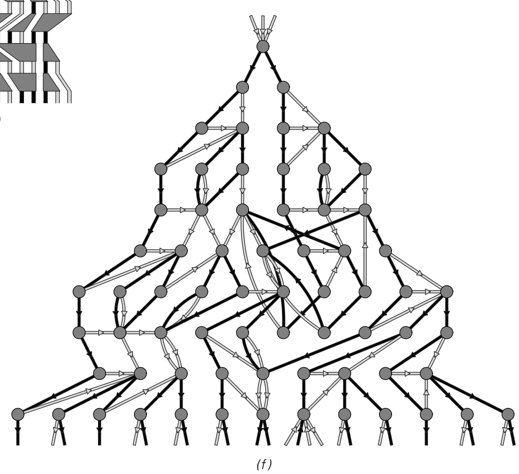
The picture below shows an example of one such system, and illustrates how—just like in a mobile automaton—relations between updating events can be represented by a causal network.

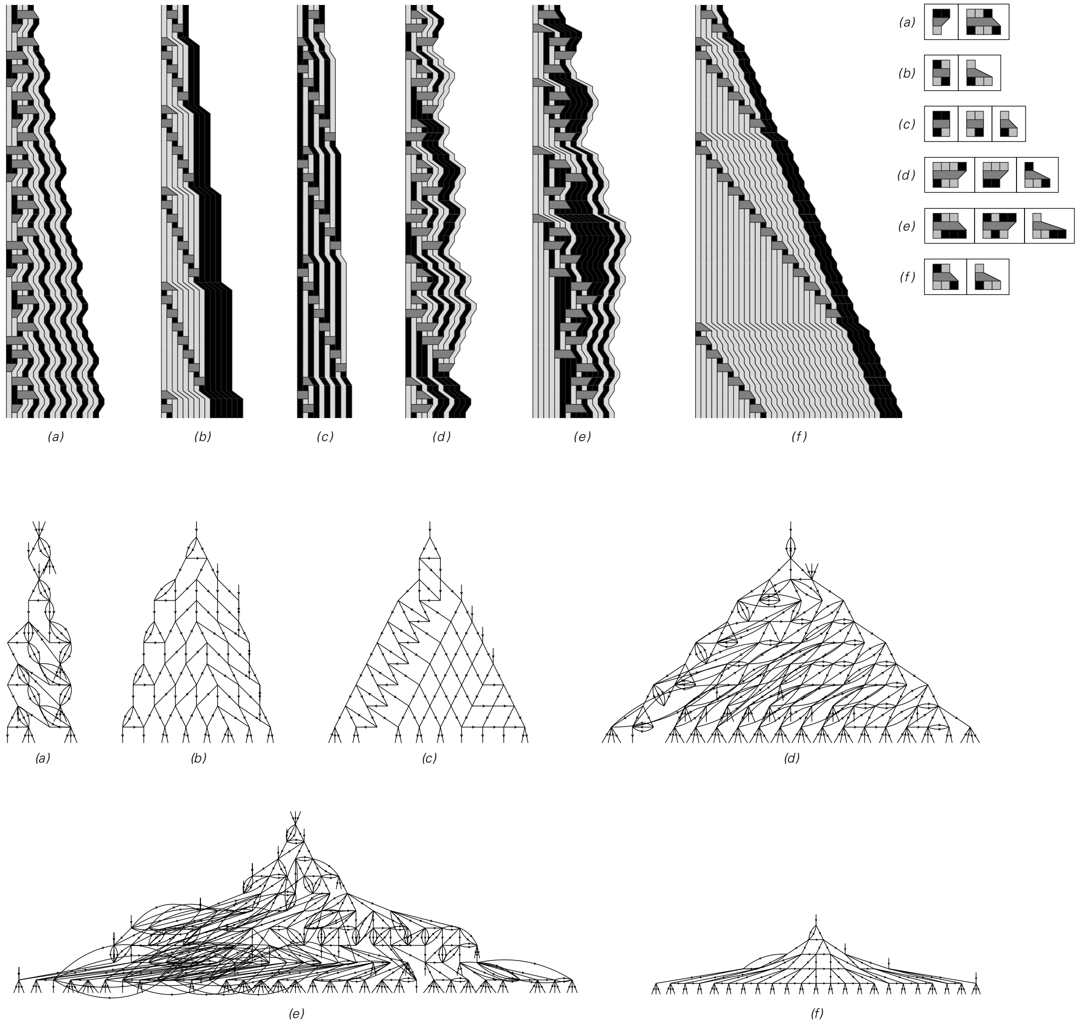




Steps in the construction of a causal network from a general substitution system. The substitution system works by replacing blocks of elements at each step according to the rule shown. Each such updating event becomes a node in the causal network. In the case shown here, all the replacements found to fit in a left-to-right scan are carried out at each step.







Examples of sequential substitution systems of the type discussed on page 88, and the causal networks that emerge from them. In a sequential substitution system only the first replacement that is found to apply in a left-to-right scan is ever performed at any step. Rule (a) above yields a causal network that is purely repetitive and thus yields no meaningful notion of space. Rules (b), (c) and (d) yield causal networks that in effect grow roughly linearly with time. In rule (f) the causal network grows exponentially, while in rule (e) the causal network also grows quite rapidly, though its overall growth properties are not clear. Note that to obtain the 10 levels shown here in the causal network for rule (e), it was necessary to follow the evolution of the underlying substitution system for a total of 258 steps.

Substitution systems that correspond to mobile automata can be thought of as having rules and initial conditions that are specially set up so that only one updating event can ever occur on any particular step. But with most rules—including the one shown on the previous page—there are usually several possible replacements that can be made at each step.

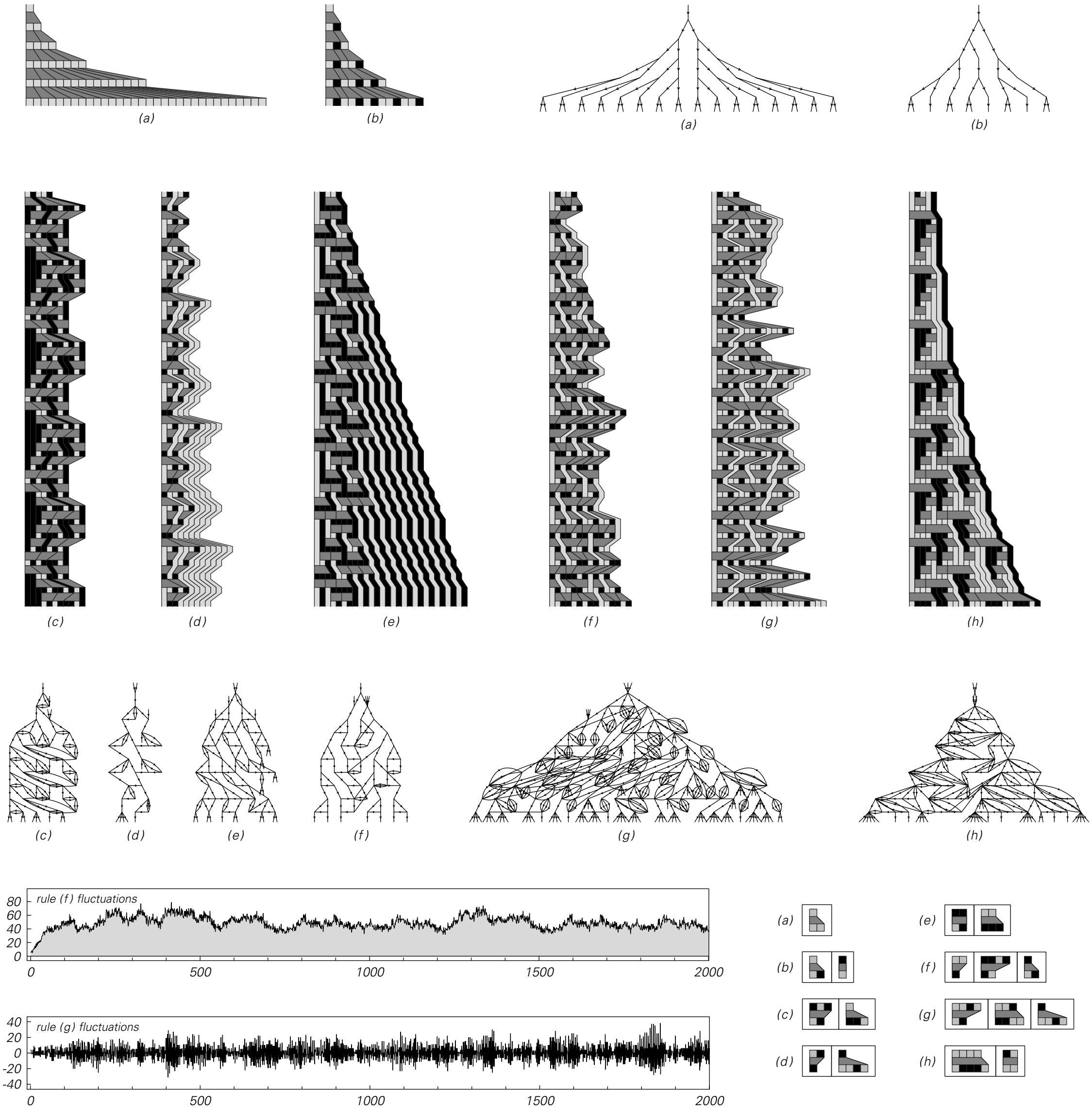
One scheme for deciding which replacement to make is just to scan the string from left to right and then pick the first replacement that applies. This scheme corresponds exactly to the sequential substitution systems we discussed in Chapter 3.

The pictures on the facing page show a few examples of what can happen. The behavior one gets is often fairly simple, but in some cases it can end up being highly complex. And just as in mobile automata, the causal networks that emerge typically in effect grow linearly with time. But, again as in mobile automata, there are rules such as (a) in which there is no growth—and effectively no notion of space. And there are also rules such as (f)—which turn out to be much more common in general substitution systems than in mobile automata—in which the causal network in effect grows exponentially with time.

But why do only one replacement at each step? The pictures on the next page show what happens if one again scans from left to right, but now one performs all replacements that fit, rather than just the first one.

In the case of rules (a) and (b) the result is to update every single element at every step. But since the replacements in these particular rules involve only one element at a time, one in effect has a neighbor-independent substitution system of the kind we discussed on page 82. And as we discovered there, such systems can only ever produce rather simple behavior: each element repeatedly branches into several others, yielding a causal network that has the form of a regular tree.

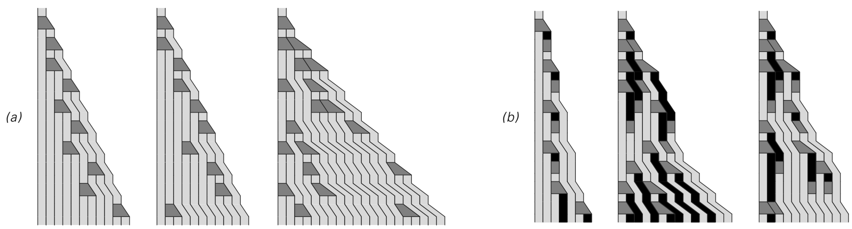
So what happens with replacements that involve more than just one element? In many cases, the behavior is still quite simple. But as several of the pictures on the next page demonstrate, fairly simple rules are sufficient—as in so many other systems that we have discussed in this book—to obtain highly complex behavior.



Examples of general substitution systems and the causal networks that emerge from them. In the pictures shown here, every replacement that is found to fit in a left-to-right scan is performed at each step. Rules (a) and (b) act like neighbor-independent substitution systems of the type discussed on page 84, and yield exponentially growing tree-like causal networks. The plots at the bottom show the growth rates of the patterns produced by rules (f) and (g). In the case of rule (f) the pattern turns out to be repetitive, with a period of 796 steps.

One may wonder, however, to what extent the behavior one sees depends on the exact scheme that one uses to pick which replacements to apply at each step. The answer is that for the vast majority of rules—including rules (c) through (g) in the picture on the facing page—using different schemes yields quite different behavior—and a quite different causal network.

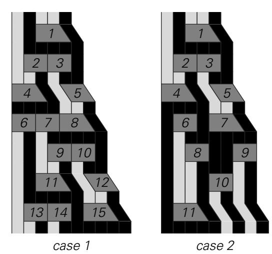
But remarkably enough there do exist rules for which exactly the same causal network is obtained regardless of what scheme is used. And as it turns out, rules (a) and (b) from the picture on the facing page provide simple examples of this phenomenon, as illustrated in the pictures below.



The behavior of rules (a) and (b) from the facing page when replacements are performed at random. Even though the detailed patterns obtained are different, the causal networks in these particular rules that represent relationships between replacement events are always exactly the same.

For each rule, the three different pictures shown above correspond to three different ways that replacements can be made. And while the positions of particular updating events are different in every picture, the point is that the network of causal connections between these events is always exactly the same.

This is certainly not true for every substitution system. Indeed, the pictures on the right show how it can fail, for example, for rule (e) from the facing page. What one sees in these pictures is that after event 4, different choices of replacements are made in the two cases, and the causal relationships implied by these replacements are different.



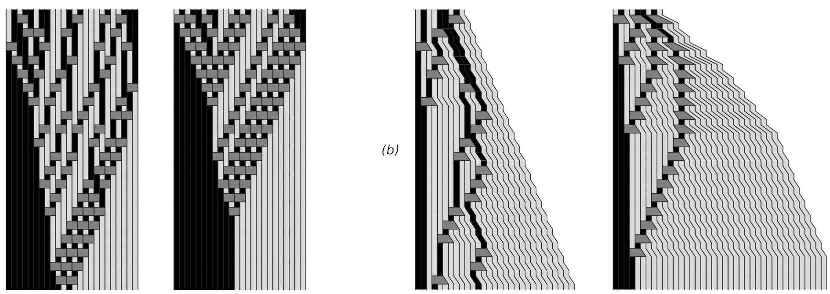
Examples of two different ways of performing replacements in rule (e) from the facing page, yielding two different causal networks.

So what could ensure that no such situation would ever arise in a particular substitution system? Essentially what needs to be true is that the sequence of elements alone must always uniquely determine what replacements can be made in every part of the system. One still has a choice of whether actually to perform a given replacement at a particular step, or whether to delay that replacement until a subsequent step. But what must be true is that there can never be any ambiguity about what replacement will eventually be made in any given part of the system.

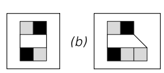
In rules like the ones at the top of page 500 where each replacement involves just a single element this is inevitably how things must work. But what about rules that have replacements involving blocks of more than one element? Can such rules still have the necessary properties?

The pictures below show two examples of rules that do. In the first picture for each rule, replacements are made at randomly chosen steps, while in the second picture, they are in a sense always made at the earliest possible step. But the point is that in no case is there any ambiguity about what replacement will eventually be made at any particular place in the system. And as a result, the causal network that represents the relationships between different updating events is always exactly the same.

(a) (b)



(a) (b)



Examples of substitution systems in which the same causal networks are obtained regardless of the way in which replacements are performed. In the first picture for each rule, the replacements are performed essentially at random. In the second picture they are performed on the earliest possible step. Note that rule (a) effectively sorts the elements in its initial conditions, always placing black before white.

So what underlying property must the rules for a substitution system have in order to make the system as a whole operate in this way? The basic answer is that somehow different replacements must never be able to interfere with each other. And one way to guarantee this is if the blocks involved in replacements can never overlap.

In both the rules shown on the facing page, the only replacement specified is for the block □■. And it is inevitably the case that in any sequence of □'s and ■'s different blocks of the form □■ do not overlap. If one had replacements for blocks such as ■■, □□ or ■□■ then these could overlap. But there is an infinite sequence of blocks such as ■□, ■□□ or ■■□□ for which no overlap is possible, and thus for which different replacements can never interfere.

If a rule involves replacements for several distinct blocks, then to avoid the possibility of interference one must require that these blocks can never overlap either themselves or each other. The simplest non-trivial pair of blocks that has this property is ■■□□, ■■□■□, while the simplest triple is ■■■□□, ■□■□□, ■□■■□□. And any substitution system whose rules specify replacements only for blocks such as these is guaranteed to yield the same causal network regardless of the order in which replacements are performed.

In general the condition is in fact somewhat weaker. For it is not necessary that no overlaps exist at all in the replacements—only that no overlaps occur in whatever sequences of elements can actually be generated by the evolution of the substitution systems.

And in the end there are then all sorts of substitution systems which have the property that the causal networks they generate are always independent of the order in which their rules are applied.

So what does this mean for models of the universe?

In a system like a cellular automaton, the same underlying rule is in a sense always applied in exact synchrony to every cell at every step. But what we have seen in this section is that there also exist systems in which rules can in effect be applied whenever and wherever one wants—but the same definite causal network always emerges.

So what this means is that there is no need for any built-in global clock, or even for any mechanism like an active cell. Simply by choosing the appropriate underlying rules it is possible to ensure that any sequence of events consistent with these rules will yield the same causal network and thus in effect the same perceived history for the universe.