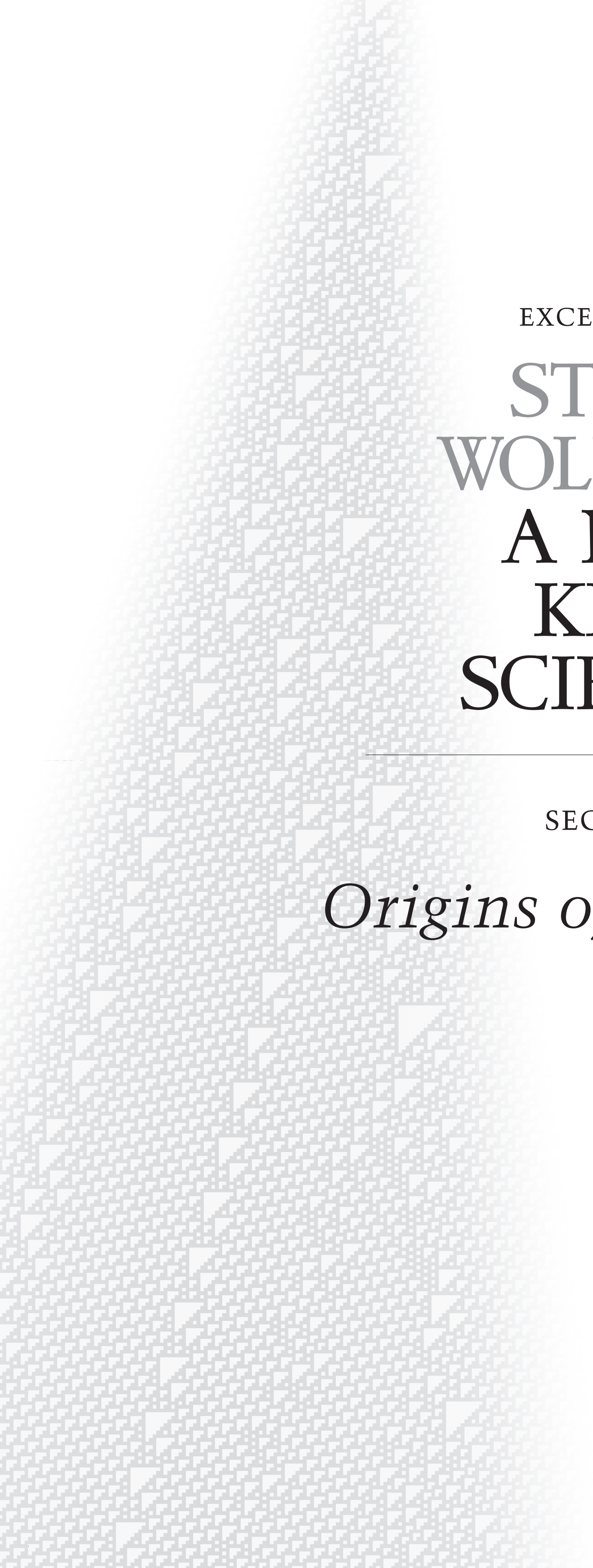


EXCERPTED FROM

STEPHEN WOLFRAM
A NEW KIND OF SCIENCE

SECTION 7.7

Origins of Discreteness

Origins of Discreteness

In the previous section we saw that even though a system may on a small scale consist of discrete components, it is still possible for the system overall to exhibit behavior that seems smooth and continuous. And as we have discussed before, the vast majority of traditional mathematical models have in fact been based on just such continuity.

But when one looks at actual systems in nature, it turns out that one often sees discrete behavior—so that, for example, the coat of a zebra has discrete black and white stripes, not continuous shades of gray. And in fact many systems that exhibit complex behavior show at least some level of overall discreteness.

So what does this mean for continuous models? In the previous section we found that discrete models could yield continuous behavior. And what we will find in this section is that the reverse is also true: continuous models can sometimes yield behavior that appears discrete.

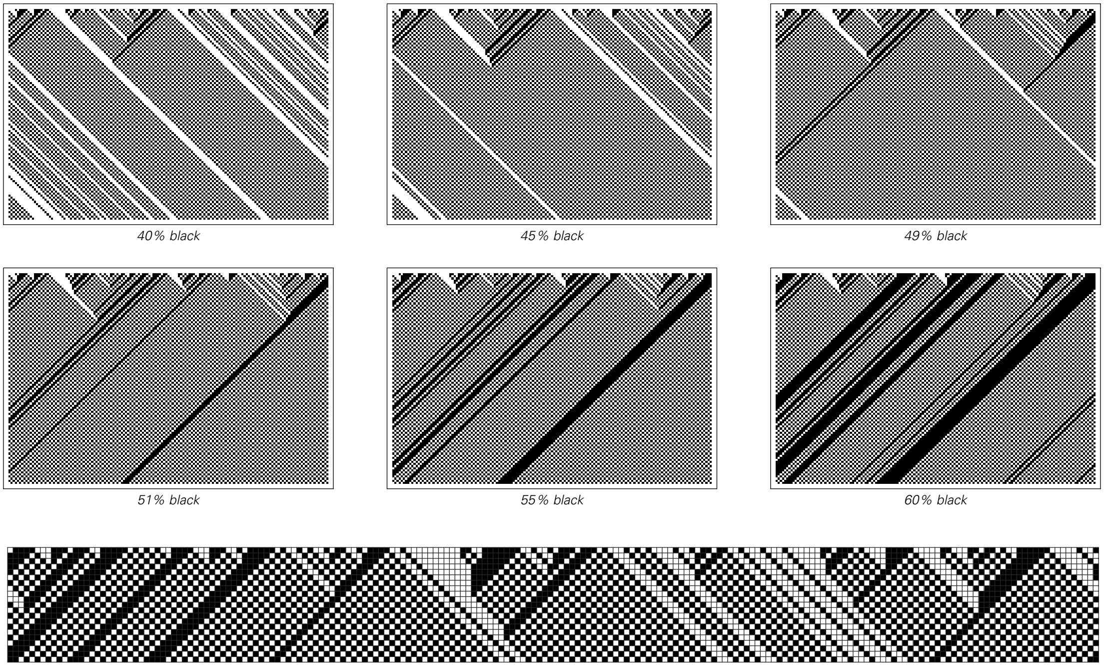
Needless to say, if one wants to study phenomena that are based on discreteness, it usually makes more sense to start with a model that is fundamentally discrete. But in making contact with existing scientific models and results, it is useful to see how discrete behavior can emerge from continuous processes.

The boiling of water provides a classic example. If one takes some water and continuously increases its temperature, then for a while nothing much happens. But when the temperature reaches 100°C, a discrete transition occurs, and all the water evaporates into steam.

It turns out that there are many kinds of systems in which continuous changes can lead to such discrete transitions.

The pictures at the top of the next page show a simple example based on a one-dimensional cellular automaton. The idea is to make continuous changes in the initial density of black cells, and then to see what effect these have on the overall behavior of the system.

One might think that if the changes one makes are always continuous, then effects would be correspondingly continuous. But the pictures on the next page demonstrate that this is not so.



40% black

45% black

49% black

51% black

55% black

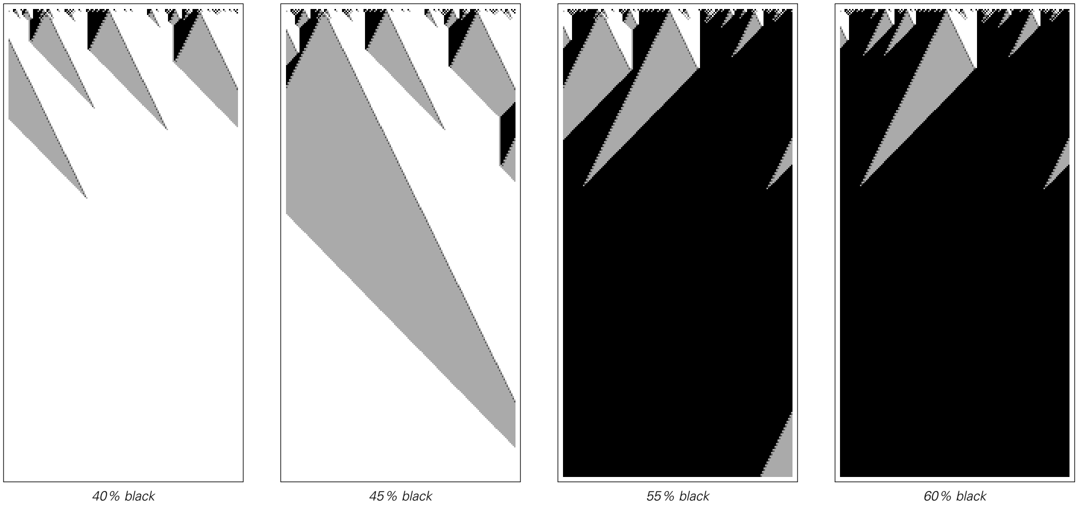
60% black

A one-dimensional cellular automaton that shows a discrete change in behavior when the properties of its initial conditions are continuously changed. If the initial density of black cells is less than 50%, then only white stripes ultimately survive. But as soon as the density increases above 50%, the white stripes disappear, and black stripes dominate. The underlying rule for the cellular automaton shown takes the new color of a cell to be the color of its right neighbor if the cell is black and its left neighbor if the cell is white. (This corresponds to rule 184 in the scheme described on page 53.)

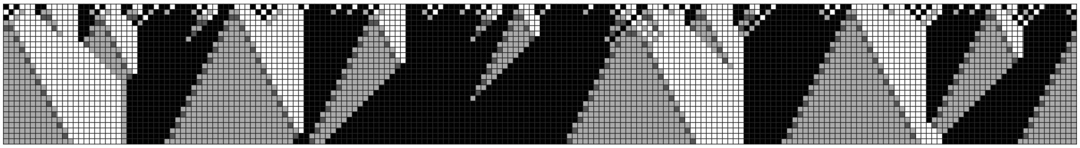
When the initial density of black cells has any value less than 50%, only white stripes ever survive. But as soon as the initial density increases above 50%, a discrete transition occurs, and it is black stripes, rather than white, that survive.

The pictures on the facing page show another example of the same basic phenomenon. When the initial density of black cells is less than 50%, all regions of black eventually disappear, and the system becomes completely white. But as soon as the density increases above 50%, the behavior suddenly changes, and the system eventually becomes completely black.

It turns out that such discrete transitions are fairly rare among one-dimensional cellular automata, but in two and more dimensions



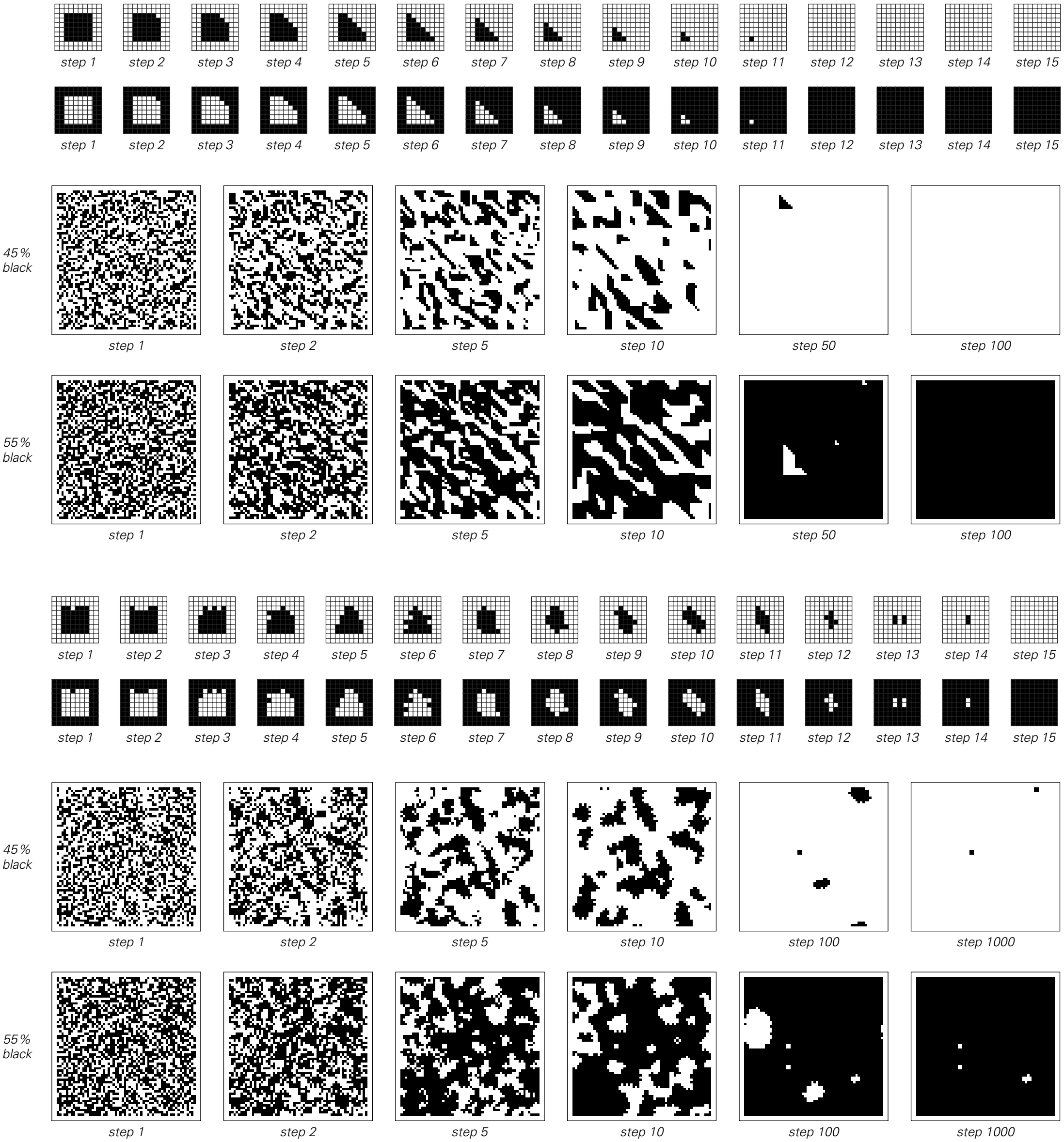
40% black *45% black* *55% black* *60% black*



A one-dimensional cellular automaton in which the density of black cells obtained after a large number of steps changes discretely when the initial density of black cells is continuously increased. With an initial density below 50%, regions of black always eventually disappear. But as soon as the density is increased above 50%, regions of black progressively expand, eventually taking over the whole system. The underlying rule allows four possible colors for each cell. The rule is set up so that whenever a region of black occurs to the left of a region of white, an expanding region of gray appears in between. The crucial point is then that if the region of white is narrower than the region of black, then the gray will reach the edge of the white before it reaches the edge of the black. And when this happens, the black expands and the gray gradually tapers away.

they become increasingly common. The pictures on the next page show two examples—the second corresponding to a rule that we saw in a different context at the end of the previous section.

In both examples, what essentially happens is that in regions where there is an excess of black over white, an increasingly large fraction of cells become black, while in regions where there is an excess of white over black, the reverse happens. And so long as the boundaries of the regions do not get stuck—as happens in many one-dimensional cellular automata—the result is that whichever color was initially more common eventually takes over the whole system.

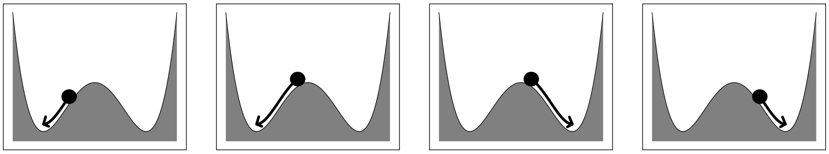


Two examples of two-dimensional cellular automata that show discrete transitions in behavior when the density of initial black cells is continuously varied. In the top rule, the new color of a particular cell is found simply by looking at that cell and its immediate neighbors above and to the right. If two or more of these three cells are black, then the new color is black; otherwise it is white. The pictures in the middle above show that with this rule blocks of opposite color are progressively destroyed, so that whichever color was initially more common eventually dominates completely. The bottom rule above is exactly the same as was shown on page 336. Whichever color was initially more common again eventually dominates, though with this rule it takes somewhat longer for this to occur.

In most cellular automata, the behavior obtained after a long time is either largely independent of the initial density, or varies quite smoothly with it. But the special feature of the cellular automata shown on the facing page is that they have two very different stable states—either all white or all black—and when one changes the initial density a discrete transition occurs between these two states.

One might think that the existence of such a discrete transition must somehow be associated with the discrete nature of the underlying cellular automaton rules. But it turns out that it is also possible to get such transitions in systems that have continuous underlying rules.

The pictures below show a standard very simple example of how this can happen. If one starts to the left of the center hump, then the ball will always roll into the left-hand minimum. But if one progressively changes the initial position of the ball, then when one passes the center a discrete transition occurs, and the ball instead rolls into the right-hand minimum.



A standard simple example of a continuous system in which there is a discrete change in behavior as a consequence of a continuous change in initial conditions. When the ball starts anywhere to the left of the center line, it rolls into the left-hand minimum. But if instead it starts on the right, then it rolls into the right-hand minimum. There are many systems in nature that follow the same general form of mathematical equations as those that describe the energy and motion of the ball.

Thus even though the mathematical equations which govern the motion of the ball have a simple continuous form, the behavior they produce still involves a discrete transition. And while this particular example may seem contrived, it turns out that essentially the same mathematical equations also occur in many other situations—such as the evolution of chemical concentrations in various chemical reactions.

And whenever such equations arise, they inevitably lead to a limited number of stable states for the system, with discrete transitions occurring between these states when the parameters of the system are varied.

So even if a system at some level follows continuous rules it is still possible for the system to exhibit discrete overall behavior. And in fact it is quite common for such behavior to be one of the most obvious features of a system—which is why discrete systems like cellular automata end up often being the most appropriate models.

The Problem of Satisfying Constraints

One feature of programs is that they immediately provide explicit rules that can be followed to determine how a system will behave. But in traditional science it is common to try to work instead with constraints that are merely supposed implicitly to force certain behavior to occur.

At the end of Chapter 5 I gave some examples of constraints, and I showed that constraints do exist that can force quite complex behavior to occur. But despite this, my strong suspicion is that of all the examples of complex behavior that we see in nature almost none can in the end best be explained in terms of constraints.

The basic reason for this is that to work out what pattern of behavior will satisfy a given constraint usually seems far too difficult for it to be something that happens routinely in nature.

Many types of constraints—including those in Chapter 5—have the property that given a specific pattern it is fairly easy to check whether the pattern satisfies the constraints. But the crucial point is that this fact by no means implies that it is necessarily easy to go from the constraints to find a pattern that satisfies them.

The situation is quite different from what happens with explicit evolution rules. For if one knows such rules then these rules immediately yield a procedure for working out what behavior will occur. Yet if one only knows constraints then such constraints do not on their own immediately yield any specific procedure for working out what behavior will occur.

In principle one could imagine looking at every possible pattern, and then picking out the ones that satisfy the constraints. But even with a 10 × 10 array of black and white squares, the number of possible patterns is already 1,267,650,600,228,229,401,496,703,205,376. And with a