EXCERPTED FROM

STEPHEN WOLFRAM

A NEW KIND OF SCIENCE

SECTION 3.10

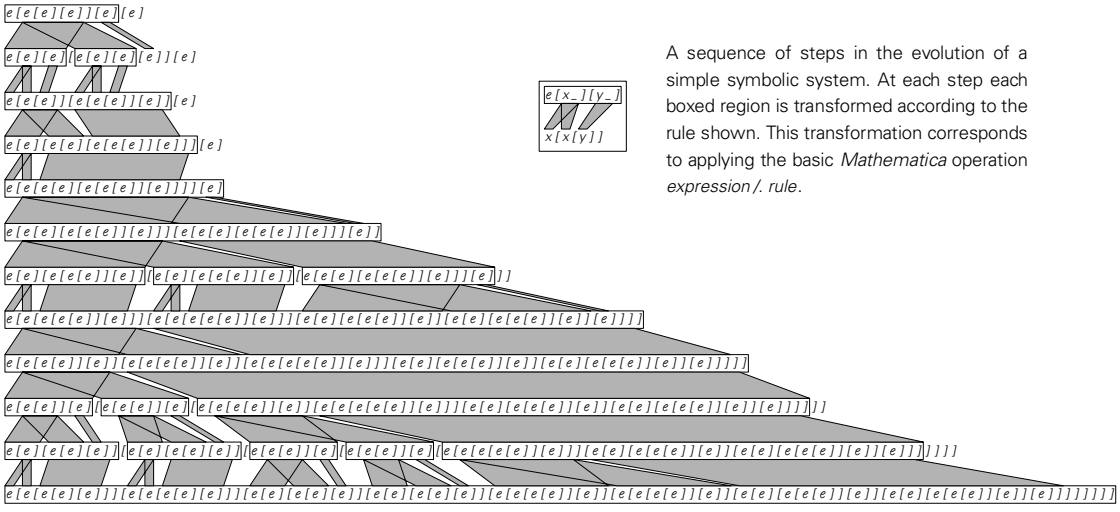
Symbolic Systems

Practical details make it somewhat difficult to do systematic experiments on such programs. But the experiments I have carried out do suggest that, just as with simple register machines, searching through many millions of short programs typically yields at least a few that exhibit complex and seemingly random behavior.

Symbolic Systems

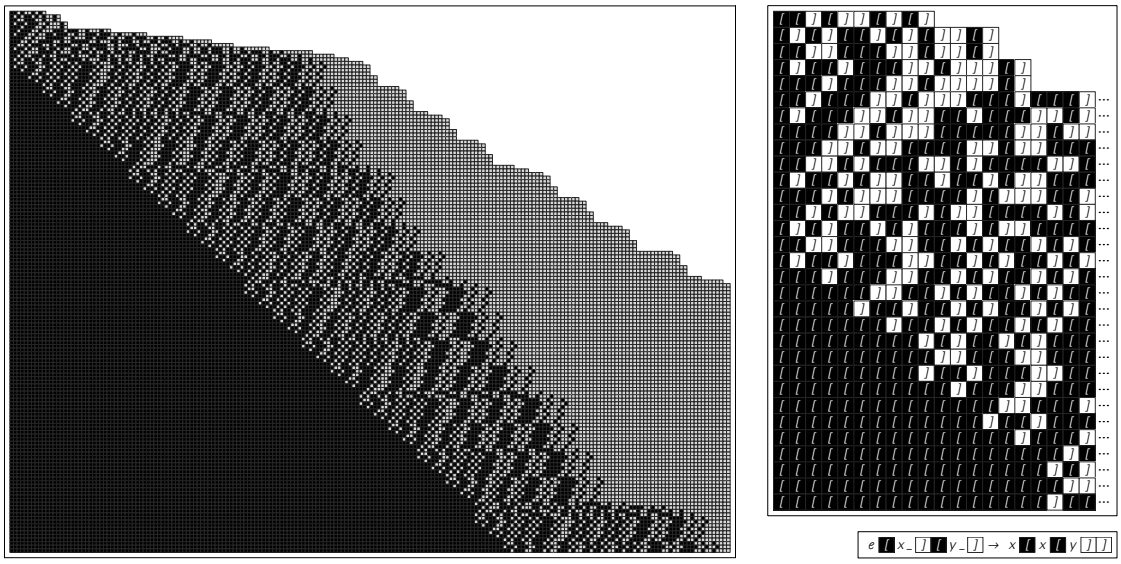
Register machines provide simple idealizations of typical low-level computer languages. But what about *Mathematica*? How can one set up a simple idealization of the transformations on symbolic expressions that *Mathematica* does? One approach suggested by the idea of combinators from the 1920s is to consider expressions with forms such as $e[e[e][e]][e][e]$ and then to make transformations on these by repeatedly applying rules such as $e[x\_][y\_] \to x[x[y]]$, where $x\_$ and $y\_$ stand for any expression.

The picture below shows an example of this. At each step the transformation is done by scanning once from left to right, and applying the rule wherever possible without overlapping.



A sequence of steps in the evolution of a simple symbolic system. At each step each boxed region is transformed according to the rule shown. This transformation corresponds to applying the basic *Mathematica* operation *expression /. rule*.

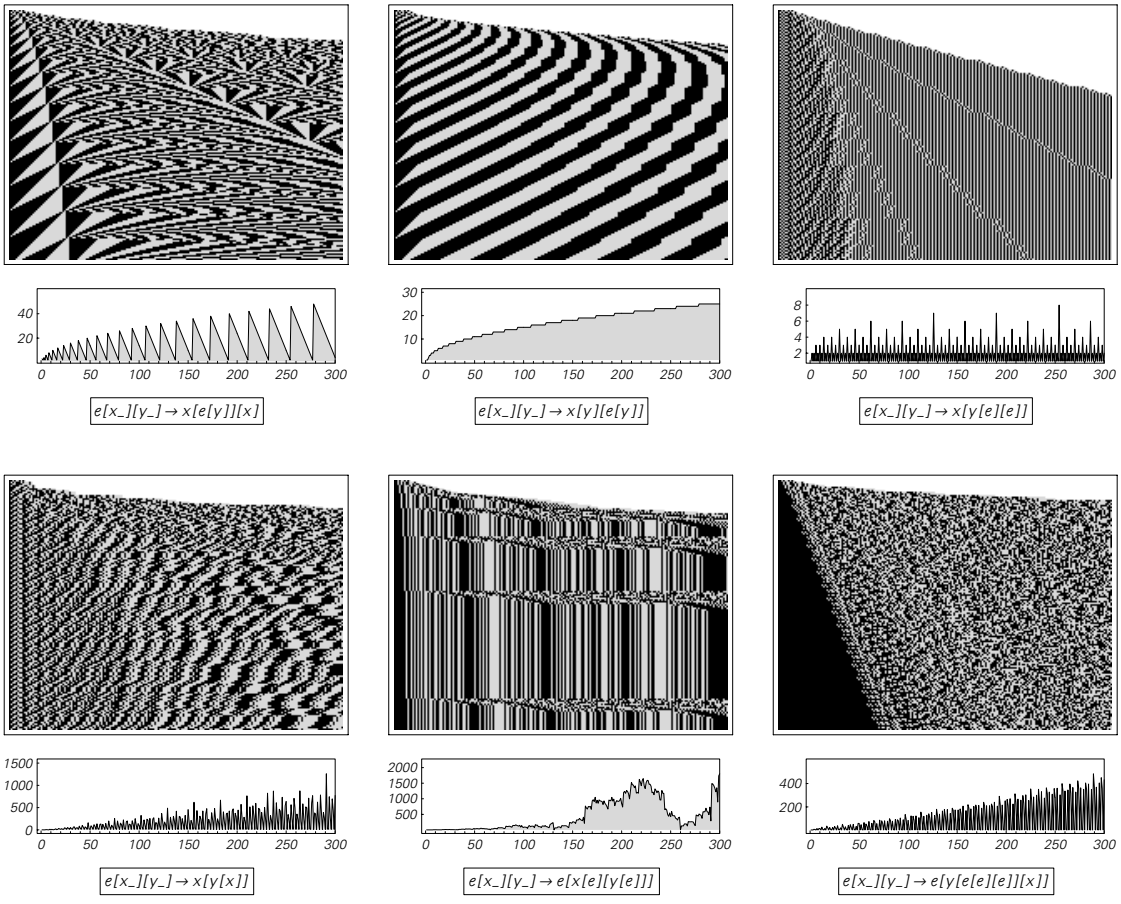
The structure of expressions like those on the facing page is determined just by their sequence of opening and closing brackets. And representing these brackets by dark and light squares respectively, the picture below shows the overall pattern of behavior generated.



More steps in the evolution on the previous page, with opening brackets represented by dark squares and closing brackets by light ones. In each case configurations wider than the picture are cut off on the right. For the initial condition from the previous page, the system evolves after 264 steps to a fixed configuration involving 256 opening brackets followed by 256 closing brackets. For the initial condition on the bottom right, the system again evolves to a fixed configuration, but now this takes 65,555 steps, and the configuration involves 65,536 opening and closing brackets. Note that the evolution rules are highly non-local, and are rather unlike those, say, in a cellular automaton. It turns out that this particular system always evolves to a fixed configuration, but for initial conditions of size n can take roughly n iterated powers of 2 (or $2^{2^{2^{\cdot^{\cdot^{\cdot}}}}}$) to do so.

With the particular rule shown, the behavior always eventually stabilizes—though sometimes only after an astronomically long time.

But it is quite possible to find symbolic systems where this does not happen, as illustrated in the pictures below. Sometimes the behavior that is generated in such systems has a simple repetitive or nested form. But often—just as in so many other kinds of systems—the behavior is instead complex and seemingly quite random.



The behavior of various symbolic systems starting from the initial condition $e[e[e]][e][e]$. The plots at the bottom show the difference in size of the expressions obtained on successive steps.