Queues, Stacks, and Transcendentality at the Transition to Chaos

Cristopher Moore
Computer Science Department, University of New Mexico

September 20, 1999

Summary by Bruno Salvy

Iteration of the logistic map

$$F_{\mu}(x) = 4\mu x (1-x), \quad \mu \in [0,1)$$

is a classical example of a discrete dynamical system exhibiting chaos. Depending on the value of $\mu$, the iterates of an arbitrary $x \in I = [0,1]$ are attracted to a limit cycle of size a power of 2 (see [3]). Figure 1 displays the values of $F_{\mu}^{50}(1/2), \ldots, F_{\mu}^{100}(1/2)$ as $\mu$ increases from 0 to 1, where $F^k$ denotes the $k$th iterate of $F$. Figure 2 shows an example of a trajectory with an attracting 4-cycle.

To each $x \in I$ is associated the infinite word $a(x) \in \{0,1\}^*$ whose $k$th letter is 0 if $F_{\mu}^k(x) \leq 1/2$ and 1 otherwise. The aim of Cristopher Moore and Porus Lakdawala [6] is to study the language $L$ formed by the set of prefixes of all $a(x)$ for $x \in I$ (the symbolic dynamics of $F_{\mu}$) and its evolution as $\mu$ increases from 0 to 1. For instance, the language corresponding to $\mu$ in Figure 2 is

$$L = 0^*1^*(10)^*(1011)^*.$$

This can be interpreted as follows: the first iterates can be smaller than 1/2, but apart from the fixed point at 0 (where $a(0) = 0^*$) they eventually get larger. Then, apart from the second fixed point of $F_{\mu}$ (where $a$ is 1*) the iterates are attracted by the 4-cycle, but they may first have a few iterates on the other side of 1/2, hence the $(10)^*$. One should also account for those prefixes which

![Figure 1. The period-doubling phenomenon.](image1)

![Figure 2. 100 iterates for $\mu = 0.884$.](image2)
do not end exactly at the end of a period; this is obtained by concatenating \((\epsilon|1|10|101)\) at the end of \(L\) and removing \((1|10)\) which otherwise would be counted twice. However, these modifications introduce unnecessary technicalities and will be ignored in what follows. When \(\mu\) increases further, the 4-cycle becomes repelling and gives rise to an attracting 8-cycle. This does not change \(L\) until the third element of the cycle becomes smaller than \(1/2\), and then

\[ L = 0^* 1^* (10)^* (1011)^* (10111010)^*. \]

Examples of corresponding 8-cycles are given in Figure 3 and 4.

1. Transcendality at the Transition to Chaos

This process leads to a sequence of languages

(1)

\[ L_0 = 0^*, \quad L_1 = L_0 w_0^*, \quad L_2 = L_1 w_1^*, \ldots, \]

with \(w_0 = 1\) and \(w_{n+1} = R(w_n)\) where \(R\) is the substitution

(2)

\[ R: 0 \mapsto 11, 1 \mapsto 10. \]

Each of these languages is regular. Their generating functions are obtained by translation from (1):

\[ L_0(z) = \frac{1}{1 - z}, \quad L_n(z) = L_{n-1}(z) \frac{1}{1 - z^{2^n}}. \]

The transition to chaos corresponds to letting \(\mu\) approach 1. The limiting value of \(w_n\) is the fixed point of \(R\), the Morse sequence. The limiting value of \(L\) has a generating function defined by

(3)

\[ L_\infty(0) = 1, \quad L_\infty(z) = \frac{L_\infty(z^2)}{1 - z}. \]

From this it follows that \(L_\infty(z)\) has an infinite number of singularities on the unit circle, thus \(L_\infty(z)\) is not algebraic and the corresponding language is not context-free. This generating function is classical: it is the generating function of binary partitions studied by Mahler [5] and de Bruijn [2].
who showed that the logarithm of the \( n \)th Taylor coefficient of \( L_\infty \) behaves asymptotically like
\[
\frac{1}{2\log 2} \left( \frac{\log n}{\log n} \right)^2 + \left( 1 + \frac{1}{\log 2} + \frac{\log \log 2}{\log 2} \right) \log n \\
- \left( 1 + \frac{\log \log 2}{\log 2} \right) \log \log n + F \left( \frac{\log n - \log \log n}{\log 2} \right) + o(1),
\]
where \( F \) is a periodic function with period 1 for which a full Fourier expansion is known.

2. Stacks of Stacks

Since the language \( L_\infty \) is not context-free, it cannot be recognized with a finite amount of memory. The question addressed by Moore and Lakadawala is to determine how simple a long-term memory mechanism recognizing \( L_\infty \) can be. This in turn is expected to give more precise information on the nature of the transition to chaos. Two natural candidates for the mechanism are the queue (first in–first out) and the stack (last in–first out).

Since context-free languages are those recognized by automata with a stack (pushed-down automata) [4], a stack is not sufficient to recognize \( L_\infty \). A more general class of languages is provided by indexed languages [4, p. 389], whose grammars look like context-free grammars except for string indices, which can be appended to non-terminals. Production rule involving an indexed non-terminal copies this index to all non-terminals it produces. For instance, \( \{ a^n b^n c^n \mid n \geq 0 \} \) is not context-free but it is indexed, the grammar being

\[
S \to T_f g, \\
A_f \to a A, \\
A_g \to a, \\
B_f \to b B, \\
B_g \to b, \\
C_f \to c C, \\
C_g \to c.
\]

From the start state, the first rule introduces a final \( g \), the second one stacks any number of \( f \)'s to produce \( T_f^n g \). The third rule then produces \( A_f^n g B_f^n g C_f^n g \), the rules on the second line pop these indices and the final \( g \) is popped by the rules on the third one. More generally, these languages are recognized by nested stack automata which resemble stacks of stacks.

It turns out that \( L_\infty \) can be recognized by such a grammar:

\[
S \to 0 S \mid T, \\
A_f \to A B, \\
A_g \to 1, \\
T \to A_f \mid A_g T \mid T_f, \\
B_f \to A A, \\
B_g \to 0.
\]

The first rule takes care of the initial \( 0^* \), the second one first stacks a number \( k \) of \( f \)'s at the end and then either produces an \( A_f^n g \) or an \( A_f^n g T_f k \). To this final \( T_f k \), more \( f \)'s can then be stacked by that same rule. To see that \( L_\infty \) is the end result, it is then sufficient to show why \( A_f^n g \) actually produces the word \( w_k \) from (1). This follows from productions in the second line performing the substitution \( R \) from (2).

3. Queues

Automata with \( k \) queues can simulate the \( k \) tapes of a multi-tape Turing machine. However, restricting the way the queues are accessed by imposing a bound on the number of transitions performed for each symbol of the input string leads to the class of quasi-real-time queue automata [1]. The corresponding grammars are breadth-first grammars. In these grammars, a production of the form \( A \to s B \) where \( s \) is a string of terminals and \( B \) a string of non-terminals rewrites a string \( xA_f \) into \( xsyB \) and the rule has to be applied to the leftmost non-terminals first. Thus the string of
non-terminals represents the queue and the string of terminals represents the part of the input that has been read so far.

By storing the current $w_n$ on the queue and applying $R$ when necessary to expand it, Moore and Lakdawala show that $L_\infty$ is recognizable by a real-time deterministic queue automaton with one queue.

4. Stacks

Again, with no time restriction, two stacks are sufficient to simulate a universal Turing machine. Exploiting the fact that $w_n$ is a palindrome except for its last symbol, it can be shown [6] that $L_\infty$ can be recognized by a real time automaton with two stacks.

The conclusion [6] is therefore that since one queue is sufficient while two stacks are necessary, the long-term memory of the system has more of a FIFO character. It is unclear however how much of this work can be generalized to other dynamical phase transitions.

Bibliography


