Theoretical Computer Science 551 (2014) 116–131

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Universal computation is 'almost surely' chaotic

Nabarun Mondal [a,*,1], Partha Pratim Ghosh [b,2]

[a] D.E. Shaw & Co. India, Hyderabad, India
[b] Microsoft India, Hyderabad, India

ARTICLE INFO

Article history:
Received 17 December 2011
Received in revised form 19 July 2012
Accepted 6 July 2014
Available online 15 July 2014
Communicated by D. Corne

Keywords:
Turing machines
Universal computation
Chaos
Metric space
Measure space
With probability one
Aleph numbers

ABSTRACT

Fixed point iterations are known to generate chaos, for some values within the range of their parameters. It is an established fact that Turing machines are fixed point iterations. However, as these machines operate in integer space, the standard notion of a chaotic system is not readily applicable for them. Changing the state space of Turing machines from integer to rational space, the condition for chaotic dynamics can be suitably established, as presented in the current paper. Further it is deduced that, given a random input, computation performed by a universal Turing machine would be '*almost surely*' chaotic.

According to the Church–Turing thesis, there is an abstract computing device, called Universal Turing Machine (UTM), which can simulate any effective computation. It is well known that a UTM is a type of iterative map or dynamic system. Dynamic systems can exhibit chaotic behavior, and in this paper condition of chaos in a UTM has been derived. It has been formally proven that, given a random computation to simulate on any UTM, the resulting dynamics will be 'almost surely' (with probability 1) chaotic.

1. Introduction

Alan Turing's abstract computing device, *Turing machine*, is in effect a *fixed point iteration*. While computation is believed to be of predictable nature, generic fixed point iterations do not always behave similarly. Some iterations do indeed converge to a fixed point, while others diverge to infinity, yet a few oscillate between multiple points. But in some cases, the iterations neither converge, nor diverge, and do not even oscillate between points. This *deterministically random* behavior (Section 2) is informally termed as *chaos*.

* Corresponding author.
E-mail addresses: mondal@deshaw.com (N. Mondal), parthag@microsoft.com (P.P. Ghosh).

[1] Dedicated to my parents and late Professor Dr. Prashanta Kumar Nandi.
Big thanks to: Abhishek Chanda, Samira Sultan and Shweta Bansal: You all have been constant support.
In memory of: Dhrubajyoti Ghosh. Dear Dhru, rest in peace.

[2] Dedicated to my parents and family.

http://dx.doi.org/10.1016/j.tcs.2014.07.005
0304-3975/

Many theoreticians believe absence of direct connection between computation and chaos. However, many dynamic systems which are capable of simulating *universal computation* for some configurations, also exhibits chaotic behavior for some other configuration [1]. Most prominent example of this behavior is cellular automata [1].

In Section 3 a brief discussion establishes Turing machines as fixed point iterations, and lays the foundation of our work. Section 4 demonstrates the concept of chaos in the universal Turing machines. Further, it follows (Section 5) that *universal computation, 'almost surely', will be chaotic*.

As a closure (Section 6) three philosophical implications of this finding are discussed.

2. Fixed point iteration and chaos

Let us define a function, $f : X \to X$, where its domain and range are the same. Starting with an initial input 'x_0' to the function, let 'f' produce the output 'x_1', such that,

$$x_1 = f(x_0).$$

The output of the function in each step can be taken as the input to next step and hence,

$$x_n = f(x_{n-1}); \quad n \geq 1 \tag{2.1}$$

Eq. (2.1) is called a *fixed point iteration*, and is the simplest example of a *dynamic system* [2,3].

Iteration (2.1) may result in x_n converging to a limit point. It may diverge to infinity as well or exhibit chaotic behavior. Banach Fixed Point Theorem A.1, presented in Appendix A, details the condition under which the iteration would converge.

For a fixed point iteration of type (2.1), defined over some interval $f : I \to I$, the inverse of the Banach Theorem A.1 has an interesting implication. *Iterates will not converge to a limit point* if at least one of the following conditions is true.

(1) Function 'f' has *no fixed point solution*, x^*, in I: $\nexists x^* \in I$ s.t. $f(x^*) = x^*$.
(2) The interval I is *not complete*.
(3) '*f' is not a contraction mapping* satisfying (A.1).

The iterate (2.1) under such circumstances will not converge. It will either diverge or exhibit chaotic bahavior.

However, *chaos* is a tricky term to define. It is much easier to list down properties that a *chaotic system* must possess, than giving a precise definition of chaos. Appendix A has the related definitions.

Definition 2.1 *(Characteristics of chaotic dynamics).*

(1) Sensitivity to the initial condition of the system where the neighborhood of the initial point can quickly lead the system into very different final states.
(2) Having a dense (Definition A.9) collection of points with periodic orbit (Definition A.5).
(3) Topologically mixing (Definition A.10).

(3) and (2) in Definition 2.1 imply the *sensitive dependence on initial conditions* (1).

Some dynamical systems, like the one-dimensional logistic map (Eq. (2.3)) with $r = 4.0$, are chaotic everywhere. However in many cases chaotic behavior is found only within a subset of phase space.

We present three functions exhibiting *chaotic dynamics* in the following subsection.

2.1. Chaos in the fixed point iterations

2.1.1. The chaotic search for $i = \sqrt{-1}$

The Babylonian method employs an iterative map to calculate $\sqrt{S}$ with arbitrary precision.

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{S}{x_n}\right); \quad n \geq 0;\ x_n \in \mathbb{R} \tag{2.2}$$

When $S < 0$, there is no $x^* \in \mathbb{R}$ that would satisfy $x^{*2} = S$. The iteration can clearly not converge. It is discussed in [4] that the iteration eventually becomes chaotic.

2.1.2. Chaos in the logistic map

Logistic map is a very well known example of an iterative map demonstrating chaotic dynamics:

$$x_{n+1} = r x_n(1 - x_n); \quad r \in [0, 4];\ x \in (0, 1) \tag{2.3}$$

When $r = 3$, the system bifurcates into two fixed values. If r is increased further, the bifurcations continue further, till reaching a chaotic state at $r = 4$. There are numerous discussions in [5] of this behavior.