Análisis de comportamiento de datos generados por códigos con complejidad O(n)

C. Bustillo-Hernández, L.C. Cota-Gómez, J. Figueroa-Nazuno
Centro de Investigación en Computación
Instituto Politécnico Nacional
Unidad Profesional Adolfo López Mateos
{chbustillo004,lcota,jfn}@cic.ipn.mx
Paper received on 23/07/10, Accepted on 05/10/10.

Resumen. Uno de los aspectos importantes de las técnicas de prueba conocidas para la evaluación de código es su funcionamiento. Sin embargo no se puede probar todos los escenarios posibles en los que el código será utilizado; esto es, existen casos no frontera que provocan comportamiento no esperado. Este trabajo presenta resultados de códigos extraordinariamente simples con comportamiento complicado y la aplicación de diferentes técnicas de análisis, no lineal sobre éstos, utilizando una máquina de Tag. Los sistemas Tag representan el modelo de máquina de Turing universal, fundamento de la Computación Moderna; los cuales a pesar de que su codificación es de complejidad O(n), pueden generar datos con comportamiento complicado.

Palabras Clave: código, validación, técnicas de evaluación, sistemas Tag, máquina de Tag.

1 Introducción

El desarrollo de código implica una serie de actividades específicas previas, tal como el análisis y diseño del algoritmo, y aunque la funcionalidad esperada se vea reflejada en la codificación, algunas veces el resultado de la salida del programa no es el esperado; esto en algunos casos no depende precisamente de errores en el proceso de desarrollo sino en la naturaleza del problema abordado.

Las técnicas de prueba conocidas para la evaluación del código, se enfocan en que el programa cumpla con el propósito planteado, por medio de pruebas sistemáticas que comprueben la lógica interna y verifiquen los dominios de entrada y salida. En base a esto último, los métodos conocidos de ingeniería de software, no son suficientes para evaluar la calidad de los resultados arrojados por programas sumamente sencillos, cuyos datos generados tienen diferentes comportamientos con parámetros de entrada similares.

De programas muy sencillos se pueden obtener datos de salida con varias características distintas y muy complejas.

La contribución de este trabajo, es la aplicación de diferentes técnicas de análisis no lineal sobre datos producidos por la máquina clásica de Tag, que permitirán obte-

Advances in Intelligent and Information Technologies.
Research in Computing Science 50, 2010, pp. 37-45.

ner una caracterización más completa por medio de valores cuantitativos y estudiar el comportamiento no esperado de códigos simples y cortos.

2 Problema Abordado

En la literatura se ha encontrado que expresiones formales sencillas pueden producir comportamiento complicado o caótico en el sentido estricto[3], tal es el caso de las ecuaciones de Lorenz. Por otra parte en el área de electrónica lo anterior también se ha manifestádo en los circuitos muy simples como el de Chua [8,9].Así mismo Wolfram demuestra que estos comportamientos se presentan en Autómatas Celulares clasificándolos como estácionarios, cíclicos o "muy complicados"[7]. Por otro lado, De Mol ha realizado experimentaciones exhaustivas basadas sólo en ajustes de parámetros [2]. Sin embargo ninguno de los trabajos anteriores efectúa un análisis métrico cuantitativo.

Los sistemas Tag, han sido formalmente demostrados como equivalentes a una máquina de Turing universal (fundamento teórico de la Computación Moderna)[11] y son útiles para el estudio de problemas sobre Decibilidad, Halting, etc.

Un sistema Tag se define parte conjunto finito de símbolos $\{0,1,\ldots..,\mu\}$ y de reglas de la forma $\{0 \rightarrow \sigma_0,\ldots.,\ \mu \rightarrow \sigma_\mu\}$, donde $\sigma_0,\ldots..,\ \sigma_\mu$ es una secuencia de cadenas finitas compuestá de símbolos, incluyendo la cadena vacía, junto con un entero.[1]

El modelo aquí abordado, mismo que Post encontró como intratable y que ha sido ampliamente estudiado, es el sistema con los símbolos {0,1}, las reglas $\{0 \rightarrow 00,\ 1 \rightarrow 11001\}$ y ν =3. El algoritmo de máquina de Tag implementado, se describe a continuación:

Paso 1. Se ingresa una cadena binaria de cualquier longitud (ver Fig. 1)

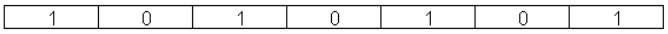

Fig. 1. Cadena binaria ingresada para ejemplificar el algoritmo.

Paso 2. Se verifica el primer elemento de la cadena y se aplica la regla correspondiente de acuerdo a: $0 \rightarrow 00$, $1 \rightarrow$ 11001.

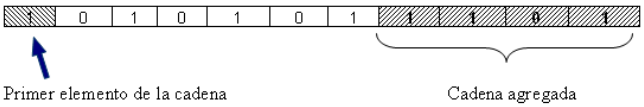

Fig. 2. Selección del primer elemento y cadena agregada.

Paso 3. Cortar ν =3 elementos al principio de la cadena como se ilustra en la Fig.3

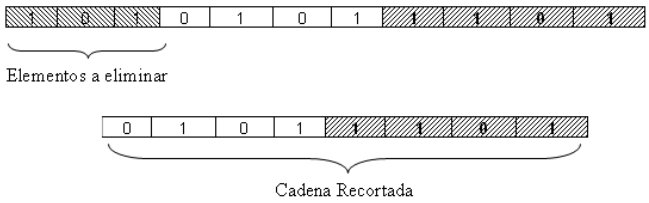

Fig. 3. Corte de la cadena.

Paso 4. Una vez cortada la cadena, se convierte a su representación decimal.

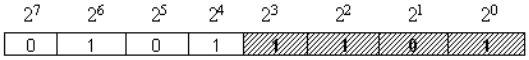

Fig. 4. Cadena recortada con su representación en decimal.

Paso 5. Se repite el paso 1, con la cadena recortada obtenida en el paso 3.

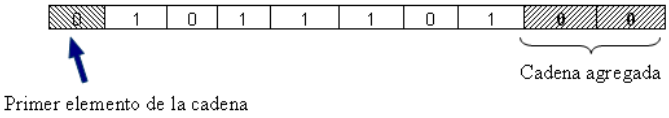

Fig. 5. Cadena obtenida, aplicando regla y corte.

En este trabajo para generar las diferentes cadenas de entrada se definió un ordenamiento sencillo para presuponer cierta estructura previa que pueda tener relación con la salida, tomando en cuenta que las reglas del sistema son fijas. En este caso se utilizó una secuencia númerica muy simple.

Algunas aspectos importantes abordados son: la aplicación de técnicas de análisis no lineal para obtener valores cuantitativos ortogonales que proporcionen una caracterización más completa sobre el comportamiento de las series generadas; la representación decimal de las cadenas generadas, en las que se pueda utilizar técnicas de análisis en el espacio continuo y no en el discreto; se hace un estudio de la representación decimal de las cadenas generadas y no sobre la longitud de éstas. [2,7]

3 Metodología

3.1. Generación de datos

Se realizaron varios experimentos de los cuales se escogió como ejemplo las cadenas de entrada 10010, 11100,100110, 110000 y 111010, correspondientes a los números 18, 38, 48 y 58 respectivamente. Se buscó un ordenamiento sencillo que procurara una relación aditiva de cantidades en 10 entre los números, en otros casos de entrada los resultados son los clásicos que se abordan en [2,7], lo cuales aquí no

se presentan. Las series de salida producidas consisten en 1000 números, convertidos a base decimal desde la cadena binaria.

3.2. Implementación del Algoritmo

A continuación se presenta el pseudocódigo del programa que implementa el algoritmo de Tag anteriormente descrito.

```
for (i=1;i<n;i++)
        if(cadena[0]==0)
            concatena(cadena,00)
        else
            concatena(cadena,1101)
        end-if

    borra_elementos(cadena,v)
    convierte_decimal(cadena)

  end-for
```

3.3. Técnicas de Análisis No Lineal aplicados a los datos

Una vez generadas las series y su representación en decimal de cada número, se obtienen los parámetros de Tiempo de Retraso y Dimensión Embebida para la construcción del Espacio de Fase, utilizando las técnicas de AMI (Average Mutual Information) y FNN (False Nearest Neighbor) respectivamente.
Posteriormente se construye un Mapa Recurrente a partir de Espacio de Fase obtenido y se calculan las propiedades de Porcentaje de Recurrencia, Porcentaje de Determinismo, Entropía, Porcentaje de Laminaridad y Tendencia.

Por último, se aplicó directamente a los datos de las series, la técnica de Análisis Gramatical.

A continuación se describen brevemente cada una de las técnicas utilizadas.

3.3.1 Tiempo de Retraso

El Tiempo de Retraso es usado para calcular la Dimensión Embebida. Sin embargo escoger el Tiempo de Retraso óptimo puede ser problemático. Si el Tiempo de Retraso es muy pequeño, las coordenadas usadas para cada vector reconstruido no será lo suficientemente independiente para llevar cualquier nueva información acerca de las trayectorias del sistema en el Espacio de Fase; si el Tiempo de Retraso es muy grande, las coordenadas podrían convertirse en aleatorias con respecto una de la otra. El método más comúnmente utlilizando para estimar el Tiempo de Retraso es calcular la función de Información Mutua [3].

3.3.2 Dimension Embebida.

Los valores de la Dimensión Embebida (*m*) y el Tiempo de Retraso (*d*) son usados para la reconstrucción en del Espacio de Fase.
La regla es obtener la Dimensión Embebida adecuada, es tal que $m \leq 2*N+1$, donde *N* es el número de variables operativas. En la mayoría de los casos *N* es desconocido y solamente es estimado [4].

3.3.4 Espacio de Fase

Es una técnica que produce gráficas que representan la dinámica de una serie de datos en un espacio *n*-dimensional, una vez determinado el valor óptimo del Tiempo de Retraso y Dimensión Embebida [4].

3.3.5 Mapa Recurrente (MR)

Es una técnica que se basa en una matriz, donde cada [i],[j]-ésimo elemento es calculado como la distancia entre vectores $\vec{V}_i$ y $\vec{V}_j$ de las series de datos reconstruidas en el Espacio de Fase [4].

A través de la estructura en el Mapa Recurrente se pueden obtener diferentes medidas como son: el Porcentaje de Recurrencia, Porcentaje de Determinismo, la Entropía, Tendencia, Porcentaje de Laminaridad, entre otros.

3.3.6 Porcentaje de Determinismo (%DET)

Está métrica mide el porcentaje de [i],[j]-ésimos elementos recurrentes que llegan a formar estructuras [4]. Se puede interpretar como una medida de determinismo en la estructura de los datos y está dada por:

$$DET = \frac{\sum_{l=l_{\min}}^{N} lP(l)}{\sum_{i,j}^{N} R_{i,j}^{m,\varepsilon}} \qquad (11)$$

donde:
$P(l)$ es la frecuencia de distribución de las longitudes l de las estructuras diagonales en el MR y *N* es el número de líneas diagonales.

3.3.7 Porcentaje de Recurrencia (%REC)

Es definida como una medida global de [i],[j]-ésimos elementos recurrentes contenidos en el MR [4], está dada por:

$$\%REC = \frac{1}{N^2}\sum_{i,j=1}^{N} R_{i,j}^{m,\varepsilon} \quad (2)$$

donde:
N es el número total de datos.
i, j son los índices de los elementos del MR.
m es la dimensión embebida
ε es el radio (umbral)

3.3.8 Entropía (ENT)

Está medida se refiere a la entropía de Shannon de la distribución de frecuencias en la longitud de las líneas diagonales [4] y está dada por:

$$ENT = -\sum_{l=l_{\min}}^{N} P(l) In P(l) \quad (3)$$

$$P(l) = \frac{P(l)}{\sum_{l=l_{\min}}^{N} P(l)} \quad (4)$$

donde:

$P(l) = \{l_i; i = 1,...., N\}$ es la frecuencia de distribución de las longitudes l de las estructuras diagonales del MR.

N es el número de líneas diagonales.

3.3.9 Porcentaje Laminar (%LAM)

Está medida se define como el porcentaje de [i],[j]-ésimos elementos recurrentes que forman las líneas verticales [4], y está dada por:

$$LAM = \frac{\sum_{v=v_{\min}}^{N} vP(v)}{\sum_{v=1}^{N} P(v)} \quad (5)$$

Donde $P(v)$ representa la frecuencia de distribución de las longitudes de línea verticales. Cada línea vertical indica que un estádo no cambia o cambia muy lentamente, es decir, el estádo permanece constante durante algún tiempo.

3.3.10 Tendencia (TEN)

Es una medida que indica que tan rápido desaparecen y ocurren cambios desde la diagonal principal. La tendencia, como su nombre sugiere, ayuda a detectar la no estácionalidad en los datos:

$$TREND = \frac{\sum_{i=1}^{\tilde{N}} (i - \tilde{N}/2)(RR_i - RR_i)}{\sum_{i=1}^{\tilde{N}} (i - \tilde{N}/2)^2} \quad (6)$$

Donde $\tilde{N} < N$ y N es el número de áreas diagonales. Si los [i],[j]-ésimos elementos recurrentes están homogéneamente distribuidos a través del MR, el valor de tendencia deberá aproximarse a cero. En cambio, si los [i],[j]-ésimos elementos recurrentes están distribuidos heterogéneamente el valor de tendencia será muy lejano a cero . La tendencia es calculada como la pendiente de la regresión de los mínimos cuadrados del porcentaje local de recurrencia como una función de desplazamientos ortogonales desde la diagonal central [4].

3.3.11 Análisis Gramatical

Está técnica encuentra patrones dentro de una serie de datos, mismos que quedarán representados como reglas de producción dentro de una gramática. Es muy eficaz para la caracterización de una serie de datos, ya que mientras más compleja sea, más reglas de producción se necesitan para construir su gramática. Por lo tanto, el número de producciones es un valor cuantitativo de su complejidad [5].

4. Resultados.

En la Tabla 1 se muestran los valores obtenidos con las técnicas de análisis no lineal.

Se obtiene el valor de Tiempo de Retraso, indispensable para la obtención de la Dimensión Embebida (*m*). En los resultados se caracteriza de forma estricta el comportamiento de los datos, donde si *m*<=3 el fenómeno se clasifica como caótico, en caso contrario se trata de un sistema complejo. De las métricas obtenidas de Mapa Recurrente se puede observar que el Porcentaje de Recurrencia en el conjunto de datos estudiado presentan la misma estructura de repetibilidad, pero sin ser estacionarias.

Los valores altos de porcentaje de Determinismo indican que los datos generados tienen definida una estructura, pero esto no implica que sean predecibles. Los resultados de Entropía presentan valores bajos y de esto se interpreta que hay poco "ruido" en los datos.

Tabla 1. Resultados de diferentes medidas con las Técnicas de Análisis No Lineal, donde porcentaje de Recurrencia (%REC), procentaje de Determinismo (%DET), Entropía (ENT), porcentaje de Laminaridad (%LAM) y Tendencia(TEN)

Cadena Binaria	Número	Comportamiento	Mapa Recurrente (MR)							Análisis Gramatical
			Tiempo de Retraso	Dimensión Embebida	%REC	%DET	ENT	%LAM	TEN	Número de Reglas de Producción
10010	18	Complejo	4	9	16,103	99,789	7,304	0	-1,012	11
11100	28	Ciclico	0	0	0	0	0	0	0	9
100110	38	Caótico	4	2	16,152	99,795	7,349	0	-0,958	12
110000	48	Complejo	4	4	16,081	99,793	7,331	0	-1,077	11
111010	58	Complejo	4	4	16,081	99,793	7,331	0	-1,077	11

Así mismo, los porcentajes de Laminaridad muestran que hay homogeneidad dentro de las series de datos, esto también se observa en los resultados de Tendencia, donde se presenta un grado bajo de desorden.

Por otro lado, en el Análisis Gramatical se observa que el caso con menor número de reglas de producción corresponde a la serie de datos que se clasificó como ciclica ya que su dimensión embebida es cero.

En general, se puede observar que aunque las medidas son ortogonales entre sí, proporcionan cierta congurencia en la interpretación de los resultados.

Todos los resultados nos indican que aunque entre las series de datos se da una variabilidad de comportamientos, cada una de ellas presenta cierta estructura instrínseca.

5. Conclusiones.

La máquina de Tag es un procedimiento muy simple que puede ejemplificar el comportamiento de ciertos programas que se retroalimentan, que bajo ciertos parámetros pueden provocar salidas no esperadas. Esto se pudo caracterizar con diferentes técnicas de análisis no lineal mediante valores cuantitativos.

Se demostró que la estructura de cada una de las series generadas no tiene ninguna relación con la secuencia numérica presupuestá entre las cadenas iniciales, a pesar de que el procedimiento consiste en reglas fijas. Por lo tanto la máquina de Tag, así como otros códigos simples, pueden no ser deterministas y sin embargo no producir aleatoriedad.

Los resultados son congruentes con la Teoría del Caos moderna que ha demostrado que en sistemas matemáticos, electrónicos o de autómatas celulares, que son modelos totalmente deterministas se pueden obtener comportamiento caótico e impredecible [3].

El análisis de máquina de Tag en una de sus formas más simples, demuestra que este problema muy pequeño y determinista produce comportamiento que tiene estructura y diferentes características de comportamiento caótico.

Referencias

1. Post, E. L.: Formal Reductions of the General Combinatorial Decision Problem. American Journal of Mathematics 65(2) 197-215
2. De Mol, L.: Tracing Unsolvability: A Mathematical, Historical and Philosophical analysis with a special focus on Tag Systems. PhD thesis, University Gent (2007)
3. Kantz, H. y Schreiber, T.: Nonlinear Time Series Analysis. University Press Cambridge (2005)
4. Takens, F.: Detecting Strange Attractors in Turbulence. Lecture Notes in Mathematics 898 366-381
5. Delfín-Santiesteban, O. R.: Análisis de la Predictibilidad de Series de Tiempo usando Algoritmos de Extracción de Reglas Gramaticales. Tesis Maestría, Centro de Investigación en Computación, Instituto Politécnico Nacional (2006)
6. De Mol, L.: On the boundaries of Solvability and Unsolvability in Tag Systems. Preprint submited to Elsevier (2008)
7. Wolfram, S.: A New Kind of Science. Wolfran Media Inc. (2002)
8. E. Bilutta, P. Pantano.: A gallery of Chua Attractors, serie A. (61). World Scientific on Nonlinear Science (2008)
9. L. Fortuna, M. Frasca, M. Gabriella Xibilia,: Chua´s Circuit Implementation. Yesterday, Today and Tomorrow. serie A. (65), World Scientific on Nonlinear Science (2009)
10. De Mol L.: Tag systems and collatz-like functions. Theoretical Computer Science 390(1) 92-101 (2008)
11. Minsky M.: Recursive Unsolvability of Post's problem of 'Tag' and other Topics in Theory of Turing Machines. Annals of Mathematics 74 (3) (1961)