REVISTA COLOMBIANA DE COMPUTACIÓN
Volumen 10, número 1
Págs. 7 - 25

Las Máquinas de Turing Como Modelo General de la Computación. ¿Hacia un Cambio de Paradigma?

Graciela Elisa Barchini *

Fecha de Recibido: 31/03/2008 Fecha de Aprobación: 28/11/2008

Resumen

La teoría de la computabilidad está actualmente fundamentada en el paradigma de la Máquinas de Turing, en las funciones recursivas y en la tesis de Church-Turing. El modelo de Turing captura la noción intuitiva de lo que es algorítmicamente computable en un sentido amplio. La interacción es un paradigma emergente que refleja el cambio en la tecnología (redes de agentes inteligentes, interfaces de usuario gráficas, sistemas distribuidos, etc.). La computación interactiva involucra la comunicación con el ambiente durante la computación. Según muchos autores la interacción se basa en modelos que difieren de los modelos algorítmicos tradicionales. Es así como, en las últimas décadas han surgido trabajos que cuestionan los pilares de la teoría de la computabilidad. La cuestión que crece es, si la noción de computación como se entiende todavía puede describirse adecuadamente por ellos. La finalidad de este artículo es presentar cuatro propuestas que se realizaron para abordar la crisis paradigmática por la que está pasando la teoría de la computabilidad, determinar la vigencia de la MT como modelo para cualquier tipo de computación y proporcionar argumentos que permitan establecer si la tesis de Church-Turing es aplicable a la computación interactiva.

Palabras clave: *Máquina de Turi ng, Computación Interactiva, Algoritmo, Computabilidad, Tesis de Church-Turing.*

Abstract

Today the computability theory is founded on the Turing Machine paradigm, the recursive function and the Church-Turing thesis. The Turing´s model captures the intuitive notion of what could be algorithmically computed -in a broad sense. *Interaction* is an emerging paradigm that reflects the shift in technology (intelligent agents nets, graphical user interfaces, distributed systems, etc.). Interactive computing involves the communication with the environment during computing. According to many authors, interaction is based on models that are different from the Turing-machine-based algorithmic models. So, many papers that argue about the foundations of the computability theory have arisen in the last decades. The growing issue is: the computing notion –as still understood- could be described appropriately by the computability theory? The aim of this article is to present four approaches that have been developed in order to deal with the computability theory paradigmatic crisis. Another objective is to determine the stability of the Turing Machine as a useful model for any computing issue and to provide arguments to determine if the Church-Turing thesis is applicable to interactive computing.

Keywords*:* *Turing Machine, Interactive Computing, Algorithm, Computability, Church-Turing thesis.*

* Universidad Nacional de Santiago del Estero, Avenida Belgrano (S) 1912, (4200) Santiago del Estero, Argentina. E-mail: grael @unse.edu.ar

1 Introducción

La teoría de la computabilidad, tiene fuertes raíces en las matemáticas y su concepto clave es el concepto de algoritmo. Fue elaborada en las décadas de los 30 y los 40 gracias a descubrimientos de lógicos matemáticos como Turing, Gödel, Church, Kleene, Markov y otros. Entre los principales aportes se pueden citar el teorema de incompletitud de Gödel; la noción matemática de función recursiva y el cálculo lambda, de Church, la máquina de Turing (MT) como un modelo abstracto de computación efectiva y los algoritmos de Markov, como sistemas de reescritura [2].

Con el tiempo, estas formalizaciones han resultado ser equivalentes con respecto a la noción de función computable[1], sólo difieren en cómo deben ser computadas [17]. Así, en los años cuarenta, la comunidad matemática aceptó el modelo de Turing y Church como definición formal de computación efectiva y, hoy, la mayoría está de acuerdo en afirmar que, resolver un problema computacional (o mejor, una clase de problemas) significa: encontrar una MT, o bien un algoritmo de Markov o bien una función (parcial) recursiva que calcule o reconozca las soluciones [2].

Desde la década del 70 hasta nuestros días, se intentó cuestionar el paradigma computacional actual y dar alternativas para enfrentar, principalmente, la interacción. Entre otros, Milner [14], señaló que los procesos concurrentes no se podían representar mediante algoritmos secuenciales y presentó los modelos de interacción complementarios a la computación que podía llevarse a cabo con una MT [15]. Leeuwen en [13], se refiere a modelos interactivos y admite que el paradigma clásico de Turing dejaba de ser apropiado para el nuevo tipo de computación. Es así como, en las últimas décadas han surgido trabajos que cuestionan los pilares de la teoría de la computabilidad.

Wegner y Goldin en “Computation Beyond Turing Machines” [23] sostienen que las MT son inapropiadas como modelo universal para la resolución de problemas computacionales y que la ciencia de la computacion es una disciplina fundamentalmente no-matemática. Frente a la cuestión: ¿Puede un algoritmo ser interactivo?, Gurevich [11] responde afirmativamente y distingue dos tipos de interacción de un algoritmo con el entorno (interacción inter-paso e interacción intra-paso). Wegner en [22] , hace pensar lo contrario y afirma que la teoría de

1 Una función es computable si hay un algoritmo que computa el valor de la función para cualquier argumento. Si la función es parcial, el caso indefinido se representa por la no terminación (no produce ningún resultado).

la computación que nosotros hemos heredado de los 60 se focaliza en la computación algorítmica como embebida en la MT y se excluyen a otros tipos de computaciones que Turing había considerado. En trabajos posteriores Goldin y Wegner [23, 24] presentan nuevos modelos de computación que, según ellos, son más apropiados a los sistemas de computación actuales: interactivos, de redes y embebidos.

El paradigma de la interacción proporciona una nueva conceptualización de los fenómenos computacionales, poniendo el énfasis en la interacción en lugar de en los algoritmos. Sin embargo, no existen aún sólidos fundamentos y enfoques satisfactorios para la computación interactiva, de manera análoga, a las provistas para los algoritmos por la MT y el cálculo lambda.

Las afirmaciones de Wegner que "*la interacción es más poderosa que los algoritmos*" y que "*la tesis de Church-Turing es un mito*" son invitaciones abiertas a reflexionar sobre el paradigma central y desarrollar herramientas conceptuales que sustenten o rechacen estas aseveraciones.

La finalidad de este artículo es presentar algunos modelos que se propusieron, en la comunidad de las ciencias de la computación, para abordar la crisis paradigmática por la que está pasando la teoría de la computabilidad. En base a estos nuevos estudios y a enfoques referidos al tema, se aspira a proporcionar argumentos que contribuyan a determinar la vigencia de la MT como modelo de las ciencias de la computación.

2 La Teoría Clásica de la Computabilidad

2.1 Génesis de la Teoría

En 1928, David Hilbert presentó, como reto, a sus colegas matemáticos demostrar tres proposiciones de gran generalidad sobre las matemáticas: a) que eran completas (cualquier enunciado matemático verdadero debería poder probarse axiomáticamente), b) que eran consistentes, es decir, que no podría nunca derivarse de sus axiomas, con ayuda de reglas deductivas, un enunciado falso, y c) que eran decidibles (todo problema matemático podría decidirse por un sí o por un no). Hilbert conjeturaba que las tres afirmaciones eran verdaderas. [12]

El tercer reto de Hilbert, denominado el *Problema de Decisión (Entscheidungsproblem)*, se puede enunciar de una manera informal:

"*Dada una representación formal de una afirmación matemática, diseñar un algoritmo general que determine si la afirmación es verdadera (teorema) o falsa".* Este problema equivalía a buscar un algoritmo para determinar si una conclusión particular podría derivarse de ciertas premisas con el uso de reglas de prueba. A partir del análisis sobre la potencia del razonamiento matemático, con respecto al mencionado problema, surgieron distintos trabajos que constituyeron los fundamentos de la teoría de la computabilidad.

1. Gödel logró probar, en 1931, que las matemáticas no podían ser completas y consistentes al mismo tiempo, con lo que contestaba en forma negativa los dos primeros desafíos de Hilbert.

2. Alonzo Church, en 1936, demostró que la aritmética es indecidible, o sea, que no existe ningún algoritmo para saber si una expresión aritmética es verdadera. Además, identificó lo efectivamente calculable con lo recursivo (tesis de Church). De esta manera, Church conjeturó que la computación efectiva era un modo alternativo de describir la noción matemática de función recursiva y creó, en torno a esa conjetura, un modelo de automatización que llamó *Lambda Calculus*. Las funciones que pueden ser computadas mediante un algoritmo finito son efectivamente las funciones recursivas.

3. También en 1936, Alan Turing respondió negativamente, al tercer reto de Hilbert. Turing mostró que no existe algoritmo alguno para resolver el problema de decisión, y en consecuencia, que la esperanza de resolver todos los problemas matemáticos de una sola vez era inconsistente.

Para poder resolver el problema de Hilbert, Turing necesitó precisar del concepto de algoritmo. Así, en su célebre artículo "*On Computable Numbers with an application to the Entscheidungsproblem*" [19], publicado en 1937, definió un algoritmo como un conjunto de reglas explícitas que permiten, en un número finito de pasos, decidir una cuestión y dio la especificación de la máquina abstracta que pudiera hacer la tarea, que desde entonces habría de ser llamada Máquina de Turing (MT). Si bien dicho artículo tuvo como finalidad primaria mostrar la incapacidad de la MT para resolver los problemas matemáticos; sin embargo, más tarde las MT fueron adoptadas por los teóricos de las ciencias computacionales en los años sesenta como un modo de resolver todos los problemas de computación [23].

Es decir, diferentes personas hicieron propuestas más o menos al mismo tiempo, independientemente unas de otras, para identificar el concepto de función eficazmente computable con varios conceptos precisos. Se demostró que todas estas nociones son equivalentes.

La *computabilidad Turing* fue una de tres maneras equivalentes de caracterizar exactamente las funciones para las cuales existen algoritmos. Los conceptos usados en las otras dos fueron la definibilidad lambda de Church-Kleene y la recursividad general de Herbrand-Gödel. La tesis Church-Turing[2], de que la MT (o su equivalente) de hecho define lo que, matemáticamente, entendemos por un procedimiento algorítmico (eficaz, recursivo o mecánico). La tesis de Turing establece que lo computable coincide con lo Turing-computable (computable mediante la MT).

La Teoría de la Computabilidad se ocupa de dividir el universo de todos los lenguajes sobre Σ^*[3], en aquellos lenguajes que pueden ser reconocidos por algoritmos efectivos y los que no. Ello conduce a las funciones no computables, es decir, a los problemas no resolubles. La teoría clásica de la Computabilidad es la parte de la computación que estudia los problemas de decisión que pueden ser resueltos con un algoritmo o equivalentemente con una MT.

En síntesis, las MT han sido aceptadas como modelos estándares para el estudio de la computabilidad durante más de medio siglo.

2.2. Máquinas de Turing

La importancia de la MT es triple [13]:

a) Por una parte, Turing abordó el problema decisorio (también llamado, después de Turing, problema del detenimiento[4]), perteneciente a la metamatemática, mostró que no hay algoritmo alguno para resolver el problema decisorio.

b) En segundo lugar, la MT inspiró la construcción del computador digital de propósito general, dando origen a la informática como disciplina científico- tecnológica [2]. Especificó su máquina y

---

2 Establece que: "una función de enteros positivos es efectivamente calculable sólo si es recursiva", comúnmente se utiliza otra versión equivalente: "Todo algoritmo o procedimiento efectivo es Turing-computable (computable por una MT).

3 Un **alfabeto** es cualquier conjunto finito no vacío, Σ; un **símbolo** (o letra) es un elemento del alfabeto; una **palabra** es una secuencia o cadena finita de símbolos de Σ. El conjunto infinito de todas las palabras que se pueden formar con símbolos de Σ se representa por Σ^*. Los problemas de decisión sobre Σ se representan por $f: \Sigma^* \rightarrow \{0,1\}$.

4 El problema del detenimiento o problema de la parada se puede expresar de esta manera: "Sea M una MT arbitraria con un alfabeto de entrada Σ, sea w *, ¿puede decidirse si la máquina M se detendrá con la entrada w?".

formalizó el concepto de algoritmo como procedimiento mecánico. El concepto de MT va asociado al de un determinado algoritmo, pero Turing extendió el concepto a la llamada *Máquina Universal de Turing*, es decir una MT capaz de realizar cualquier tarea que pudiese realizar cualquier MT concreta. Influyó en las ideas del matemático John von Neumann quién construyó la primera computadora semejante a los que ahora conocemos [14].

c) En tercer lugar, la MT, ofreció a los científicos cognoscitivos un modelo útil para sus investigaciones, en el sentido de que los procesos cognoscitivos del cerebro humano pueden ser explicados en términos de esa máquina. Por otra parte, el pensamiento filosófico de Turing se centraba en la cuestión si estas máquinas podrían algún día ser inteligentes. Por ello, Turing formuló un test para saber si una máquina es inteligente o no. En "*Computing Machinery and Intelligence*" [20], expresó su convicción de que las computadoras eran capaces de imitar la inteligencia humana y, que tal hazaña, la realizarían hacia el año 2000.

Una MT (figura 1) es un autómata que consiste de:

- Una ***unidad de control*** **que** opera a intervalos regulares, es decir, en pasos discretos; en cada paso realiza alguna función, según lo especificado por cada función de transición directa.

- Una memoria auxiliar que es una ***cinta infinita*** con acceso relativamente no-restringido. La cinta se considera dividida en cuadrados, cada uno contiene un símbolo.

- La ***cabeza de lectura / escritura*** de la cinta puede moverse a lo largo de la cinta en ambas direcciones leyendo y/o escribiendo el contenido de un cuadrado uno a uno.

La información de entrada viene constituida por un conjunto de símbolos de entrada (A_1, A_2, ..., A_n), colocados al comienzo de la cinta, uno por cuadrado, y los demás cuadrados tienen inicialmente el símbolo blanco. La máquina puede ir alterando sus entradas escribiendo sobre ellas nuevos símbolos y escribiendo símbolos sobre los demás cuadrados que contienen símbolos blancos. El dispositivo realiza las siguientes operaciones: leer-escribir un símbolo, borrar un símbolo, ir al cuadrado de la izquierda (I), ir al cuadrado de la derecha (D), o quedarse donde está (N). La máquina se parará solamente cuando alcanza el estado de parada o cuando no quedan símbolos por leer en la cinta.

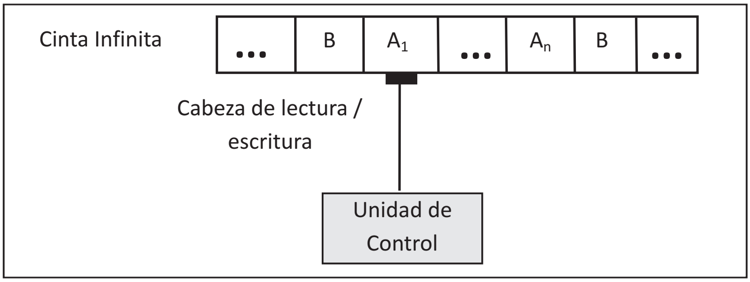

Fig 1. **Máquina de Turing**

Formalmente una MT[5] es una séxtupla:

$A = (Q, \Sigma, \tau, \delta, q_0, F)$

Q: es un conjunto finito y no vacío de estados de la unidad de control.

Σ : es el alfabeto de entrada.

Γ: es el alfabeto de cinta, donde

δ: es la función de transición directa, es una proyección

$\delta: Q \, x \, \Gamma$ subconjuntos de $Q \, x \, \Gamma \, x \, \{D, I, N\}$

$q_0 \in Q$ es el estado inicial.

$F \subseteq Q$: es el conjunto de estados finales

La entrada de una MT viene determinada por el estado actual y el símbolo leído, un par (estado, símbolo), siendo el cambio de estado, la escritura de un nuevo símbolo y el movimiento las acciones a tomar en función de una entrada.

3. COMPUTACIÓN INTERACTIVA

La interacción es un paradigma emergente de modelos de computación que refleja el cambio en la tecnología desde los grandes sistemas informáticos a las redes de agentes inteligentes, de los sistemas embebidos a las interfaces de usuario gráficas, y del procedimiento orientado a los sistemas distribuidos. La interacción se basa en modelos que difieren de los modelos algorítmicos basados en la MT de los años sesenta [5].

La computación interactiva involucra la comunicación con el mundo exterior o con el entorno durante la computación. En contraste con la comprensión tradicional de computación (algorítmica) que asume una

5 Existen en la literatura un abundante número de definiciones alternativas, pero todas ellas tienen el mismo poder computacional.

simple interfaz entre un agente computacional y su ambiente, consistente de formular una pregunta (la entrada) y generar una respuesta (la salida).

Wegner [22], sostiene que la computación actual da énfasis a los procesos abiertos que involucran la interacción entre las máquinas y los usuarios, en lugar de la transformación cerrada de una entrada a una salida.

Son muchos los autores que coinciden que el paradigma computacional clásico [3, 4, 5, 6, 10, 14, 22], ya no es totalmente apropiado para capturar todos los rasgos que de la computación actual.

Frente a la interacción como paradigma emergente y, teniendo en cuenta el paradigma actual, surge la necesidad de revisar el paradigma de la MT clásica. A continuación se presentan cuatro enfoques, correspondientes a diferentes autores, frente a esta problemática.

3.1. Enfoque 1: Hipermáquinas

La *hipercomputación*[6] se refiere al estudio de las máquinas que pueden computar más que las MT. Hay muchos ejemplos de hipermáquinas en la literatura, éstas son como las MT, modelos teóricos que usan recursos abstractos para manipular objetos abstractos como símbolos o números.

Toby Ord [16, 17], considera que algunas máquinas tienen mayor poder que las MT. En [16], presenta diez hipermáquinas (algunas propuestas por otros autores) y describe los requisitos y capacidades de cada una y las compara utilizando la teoría recursiva. Algunas de ellas son:

- **O-máquina**

Según Ord el análisis de la hipercomputación comenzó con el artículo de Turing en 1939, los "*Systems of Logic Based on Ordinals*" [21]. En dicho artículo, Turing introdujo la máquina oráculo (*O-máquina)* que permitiría resolver problemas que carecen de solución algorítmica. Si bien, el diseño utilizado no era práctico para "ejecutarlo" por los matemáticos humanos, su importancia como herramienta abstracta para analizar y ampliar el concepto del cómputo fue reconocida y tuvo un gran impacto en el campo de la teoría recursiva.

El oráculo[7] de la O-**máquina** es capaz de contestar preguntas sobre la pertenencia a un conjunto específico de números naturales. La

6 La hipercomputacion se refiere a varios métodos propuestos para la computación de funciones "Turing no calculables". Un término similar es "computación super –Turing".
7 La palabra oráculo representa una especie de *"caja negra"*, cuya principal característica

máquina, tiene tres estados especiales: el "*estado llamada*", el "*estado-1*" y el "*estado-0*" y un marcador especial: el símbolo μ. Para usar su oráculo, la máquina debe, previamente, escribir el símbolo μ en dos cuadrados de la cinta y, de esta forma, entra en el "estado llamada". En este estado solicita una petición al oráculo y, si el número escrito en los cuadrados de la cinta entre los símbolos "μ" son un elemento del conjunto oráculo, la máquina finaliza en el "estado-1"; caso contrario, termina en el "estado-0" (figura 2). Intuitivamente, se puede afirmar que una O-máquina puede resolver el problema de la parada. Para ello, basta con facilitarle el alfabeto correspondiente a dicho problema. Por tanto, este tipo de máquinas incrementan, de manera obvia, el poder computacional de una MT clásica. Además, las O-máquinas permiten computar todas las funciones recursivamente enumerables[8].

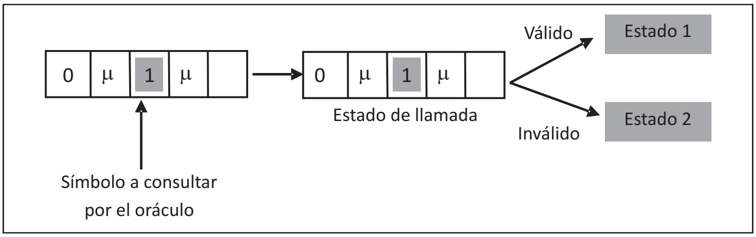

Fig. 2. **O - Máquina (MT con oráculo)**

- **Redes asincrónicas de MT**

Copeland and Sylvan, en "*Beyond the Universal Turing Machina*" [4], discuten las redes asíncronas utilizando MT que tienen una función de tiempo Ak: N → N que representan el número de unidades entre la ejecución de dos acciones consecutivas. Si dos máquinas operan sobre la misma cinta con la función de tiempo, éstas pueden resolver el problema de la parada. Esta capacidad viene claramente de las funciones que miden el tiempo y, si éstas son todas funciones recursivas, la capacidad adicional desaparece puesto que las redes se pueden simular de nuevo con una MT simple.

- **MT propensa a error**

Es una MT normal que algunas veces imprime un símbolo diferente al que está previsto. Para las MT con un alfabeto de sólo 0´s y 1´s, escribir

es que permite computar una función concreta de una sola vez..

8 Una función "f" es recursivamente enumerable por definición si y sólo si hay una MT, m, tal que para todo n perteneciente a los ℕ, la máquina dará como salida 1 si f(n) =1 y dará salida 0 o un bucle infinito (la máquina diverge), en cualquier otro caso.

un símbolo equivocado sería imprimir un 1 donde se requiere un 0 y viceversa. Este comportamiento erróneo se puede definir por una función de error, $e: N \rightarrow \{0, 1\}$ donde la máquina escribe su n-ésimo símbolo incorrectamente si y sólo *si* $e(n)=1$. Una máquina como ésta podría computar la función de parada y otras funciones no recursivas, dependiendo de su función de error.

- **MT aceleradas**

Estas máquinas ejecutan su primer paso en una unidad de tiempo y, cada paso siguiente en la mitad de tiempo que el anterior. Como se cumple que: $1+ 1/2 + 1/4 + 1/8 +... < 2$, este proceso podría terminar en un número infinito de pasos en dos unidades de tiempo. La única diferencia entre una MT acelerada y una MT estándar es la velocidad con la que opera cada una. Esta máquina puede resolver el problema de la parada: si muestra como salida cualquier valor después de dos unidades de tiempo.

- **MT no deterministicas "justas"**

Una MT deterministica siempre tiene a lo sumo una acción aplicable en cualquier circunstancia, una MT no deterministica puede tener muchas. En estos casos la ejecución puede ser entendida como una bifurcación donde se prueban las distintas posibilidades en paralelo. Cada rama de computación se ejecuta exactamente como lo haría una MT estándar, sólo que la salida se restringe. De esta manera, una MT no determinista puede ser usada para ejecutar procesos de forma paralela y obtener más rápidamente su resultado. Una MT no determinista no puede computar ninguna función que una MT no pueda computar. Sin embargo, si se restringe el "no determinismo" a computaciones "justas" la situación es diferente. Una computación de una MT no determinista se dice que es "justa" si, durante su computación infinita, se llega a un estado determinado y todas las transiciones posibles se han elegido. Es "injusta" si alguna de las transiciones nunca es elegida. Algunos autores, han mostrado como las MT no deterministas "justas" pueden computar la función de la parada.

Otras máquinas descriptas en [16, 17]son : MT con inscripción iniciales, MT acopladas, MT de tiempo infinito, MT con estados infinitos y la MT probabílistica

3.2 Enfoque 2: Conceptualización de los algoritmos interactivos

Yuri Gurevich [10] analiza los algoritmos secuenciales ("determinísticos y no interactivos" y "no determinísticos e interactivos") y define la máquina abstracta de estados secuenciales (en ingles ASM) y los algoritmos secuenciales**.** El mismo autor en [3, 11], define a un algoritmo como "un sistema de la computadora (real o imaginable, físico o abstracto), a un nivel de abstracción determinado, dónde su conducta (posiblemente interactiva, posiblemente paralela, etc.) se da o puede darse por un programa".

Distingue dos tipos de interacción de un algoritmo con el entorno:

- La *interacción inter-pasos*, cuando el ambiente modifica el estado del algoritmo (a un estado permitido) "antes de", "después de", o "entre" los pasos del algoritmo.

- La *interacción intra-paso* tiene lugar durante un paso de un algoritmo. Por ejemplo una llamada a un procedimiento remoto cuyo resultado se usa para formar otra llamada a un procedimiento remota.

Sostiene que un algoritmo secuencial sigue sus instrucciones y no puede, por sí mismo, solucionar una opción no determinística, pero puede ser preparado para solicitar "ayuda" desde fuera para solucionar ese problema. Similarmente, un programa orientado a objetos no puede crear un nuevo objeto por sí solo; un comando de "crear-un-nuevo-objeto" se ingresa desde fuera (estos son ejemplos de interacción del intra-paso de un algoritmo con su ambiente).

Con respecto a los mecanismos de interacción, utiliza el denominado *"llamada a un procedimiento remoto*". Uno puede pensar en una llamada del procedimiento remota como una pregunta al ambiente dónde la llamada espera por una respuesta a su pregunta de tal forma de completar un paso y continuar el cómputo. Esta forma de interacción se llama a menudo sincrónico o bloqueado. Otra forma de interacción popular es el paso del mensaje. Después de enviar un mensaje, el remitente procede con su cómputo; esta forma de interacción se llama a menudo asíncrono o no bloqueado. Lo sincrónico / asincrónico y las terminologías del bloqueo / no bloqueo pueden crear una impresión que cada interacción del intra-paso atómica está en una de las dos formas. Sin embargo, hay un amplio espectro de posibles formas de interacción.

3.3 Enfoque 3: Modelos para computación interactiva (I)

La definición de computabilidad de Peter Wegner [22, 23, 24] difiere notablemente de la acepción clásica, afirma que es necesario un cambio ya que las MT no modelan aspectos importantes del mundo real, como la interacción y tiempo real. Sostiene que las tareas interactivas, como "conducir un auto desde la casa al trabajo", no puede comprenderse a través de los algoritmos. Los algoritmos se ejecutan automáticamente sin tomar en cuenta sus ambientes.

Goldin y Wegner consideran la relación entre computabilidad e interacción [7, 8, 9]. Wegner, en su artículo de 1997 [22], introduce el concepto de las *máquinas interactivas* que son MT con entrada y salidas (esto convierte a las MT en sistemas abiertos), permitiendo la interacción dinámica con el entorno; establece, además, que el concepto de interacción es tan poderoso que modifica la tesis de Church.

Wegner sostiene que las máquinas interactivas son computacionalmente más poderosas que las MT. Él distingue varios tipos de máquinas de la interacción según las maneras de entrada y capacidades. Las "*máquinas interactivas secuenciales*" (MIS), son máquinas de transición de estados M=(S,I,m) donde S es un conjunto enumerable de estados, I es un conjunto enumerable de entradas limitadas dinámicamente y la transición de estados m: SxI→SxO establece para cada par de acción-estado, un nuevo estado y salida.

Cada **paso de computación** (s, i)→(s', o) de una MIS se puede ver como la computación completa de una MT, donde "i" es una entrada suministrada dinámicamente, "o" es la salida que puede afectar a las siguientes entradas, y "s' " es el próximo estado de la MIS. Los elementos S e I son finitos para cualquier instante de tiempo dado aunque su tamaño es ilimitado.

En la figura 3, se observa como un flujo de E/S es de la forma (i_1,o_1), (i_2,o_2),…, donde "o_k" se computa por "i_k" y, precede y puede influir a "i_{k+1}". La dependencia de "i_{k+1}" a "o_k" se llama **conexión de E/S y representa la "***dependencia dinámica de las entradas en función de las salidas"*, característica propia de la interacción. Las transiciones m de "s_k" a "s_{k+1}" se asocian con los pares de entrada/salida (i_k,o_k).

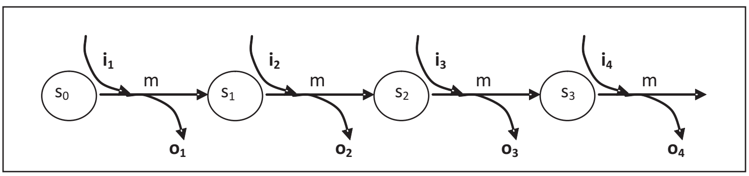

Fig. 3. **Máquina interactiva secuencial [24]**

En el artículo [6], Goldin et al. definen las *máquinas de Turing persistentes* (MTP). Una MTP es una máquina no deterministica con tres cintas infinitas: una cinta de entrada de sólo lectura, una cinta de trabajo de lectura / escritura, y una cinta de salida de sólo escritura. La cinta de trabajo es "persistente"; es decir, su contenido se conserva entre las interacciones. El estado de una MTP lo da su cinta persistente y puede ser enumerablemente infinita. Las interacciones simples de una MTP corresponden a las computaciones de una MT clásica.

El entorno interactúa con la MTP de la siguiente manera:

- pone una nueva entrada en la cinta de entrada y restablece la cabeza de la cinta de entrada a la posición inicial,
- borra la salida de la cinta de salida y restablece la cabeza de la cinta de salida a la posición inicial,
- deja la cinta de trabajo intacta (éste es el aspecto persistente de las MTP).

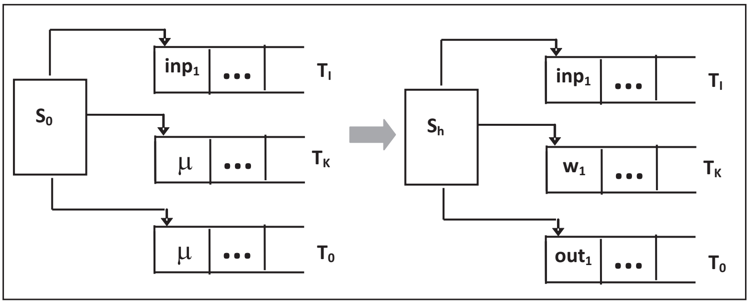

Fig. 4 . **Máquina de Turing persistente (basada en [6])**

Después de recibir un símbolo de entrada de su entorno en la cinta de entrada, una MTP computa las salidas durante algún tiempo y entonces muestra el resultado al entorno en su cinta de salida, este proceso se repite continuamente. En la figura 4, se observa como la persistencia extiende el efecto de las entradas. Un símbolo de entrada afecta la computación de los macro-pasos correspondientes, incluso la cinta de trabajo. La cinta de trabajo, a su vez, afecta los pasos de computación subsiguientes. Si la cinta de trabajo se borra, entonces el símbolo de entrada no afecta los macropasos subsiguientes, sólo su propio macropaso.

Wegner y Goldin, suscitaron muchas criticas [1, 3, 11, 18], con respecto a sus afirmaciones y modelos propuestos.

3.4 Enfoque 4: Modelos para computación interactiva (II)

Van Leeuwen y Wiedermann en [13], consideran las principales características de la computación moderna: la no-uniformidad de los programas, la interacción de máquinas, y la infinidad de operaciones posibles. Sostienen que para modelar lo que puede computarse, se necesita un modelo computacional y un escenario computacional. Proponen dos modelos de computación para capturar que algunas de las nuevas características de la computación moderna. El primer modelo, está basado en las redes de computadoras y la computación distribuida y se denomina "*máquina de sitio*" , el segundo modelo "*MT interactiva con consejo*".

Una máquina de sitio modela una computadora personal que puede, pero no necesita estar conectada a alguna red de computadoras. Informalmente, una máquina de sitio consiste en una computadora completa con un procesador y una memoria potencialmente ilimitada permanente de lectura/escritura. Ella puede ser modelada por una MT, o por una máquina de acceso aleatoria (RAM) provista con una memoria externa permanente (por ejemplo un disco), o por cualquier otro modelo similar de una computadora universal programable. La máquina está provista con varios puertos de entradas y salidas con los cuales se comunica con su ambiente. Los mensajes son secuencias finitas de símbolos escogidas de algún alfabeto finito. La máquina de sitio lee o envía mensajes símbolo por símbolo, a cada puerto un símbolo por vez. Cuando no hay símbolos para ser leídos o enviados a un símbolo especial vacío, se asume que 2 aparece en los puertos respectivos.

La MT interactiva con consejo es una MT equiparada con tres nuevas características: consejo, interacción y funcionamiento infinito. El escenario computacional de una MT interactiva es como sigue. La máquina comienza su computación con las cintas vacías. Se maneja esencialmente por una MT clásica. A cada paso, la máquina lee los símbolos que aparecen en sus puertos de entrada. Al mismo tiempo escribe algunos símbolos en sus puertos de salida. Basada en el contexto actual; es decir, en los símbolos leídos en los puertos de entrada, en sus cintas, y en el estado actual, la máquina imprime los nuevos símbolos con sus cabezas, se mueve a la izquierda o a la derecha o los deja como ellos están, y pasa a un nuevo estado. Para algún tiempo $t > 0$ se permite a la máquina pedir un consejo, pero sólo para, a lo sumo, t valores.

En [13] proponen un modelo para las redes de máquinas de sitio, tales como Internet y otros sistemas evolutivos de agentes, llamadas "*máquinas Internet",* sostienen que las máquinas internet pueden ser simuladas por las MT interactivas con consejo.

4 Conclusiones

En este artículo se han presentado sintéticamente cuatro enfoques para abordar la computación interactiva.

- Con respecto al primer enfoque, la propuesta de utilizar hipermáquinas, se basa en la formulación de modelos temporales y espaciales de las MT. Todos estos modelos intentan proveer de mayor capacidad a la MT original; ya sea en el almacenamiento, en el tiempo de ejecución (finito, infinito, transfinito) o en el modo de operación (secuencial o paralelo). La mayoría de ellas intentan solucionar el problema de la parada.

  Con una visión anticipatoria, el mismo Turing, conciente de las limitaciones de la MT clásica (apropiada sólo para computar funciones recursivas) diseño otras máquinas, como la "C-máquina" (choice-machine) y la "O-máquina", que permiten computar las funciones no computables por la MT estándar.

- En el segundo enfoque, Gurevich conceptualiza a los algoritmos interactivos, según el tipo de interacción (inter-paso e intra-paso) y la forma de interacción (sincrónica o asincrónica) que los algoritmos tienen con el entorno. Se refiere, además a los mecanismos de interacción.

- Peter Wegner, en el tercer enfoque presentado, plantea un problema interesante para las ciencias de la computación y el desarrollo del software: "¿Cómo encaja la interacción en la teoría de computabilidad?. Esta pregunta y sus posibles respuestas son ciertamente desafiantes y atrayentes para la comunidad de las ciencias de la computación.

  Wegner y Eberbach presentan, la máquina interactiva secuencial como un modelo más poderoso que la MT; sin embargo, ellos restringen, a tener todas sus entradas a priori en la cinta, a la salida de cada computación. Por tanto, cada transición está representada por una MT convencional; es decir, una máquina interactiva secuencial, puede verse como una secuencia de MT (M_1, M_2, M_3, ..., M_n), dónde la entrada de M_{i+1} depende de la salida de M_i. Por consiguiente, la máquina interactiva, por sí misma, no posee los capacidad computacional mayor que una MT.

La MT persistente de Wegner y Goldin, es el modelo canónico de la máquina interactiva. Esta máquina puede retener los datos en su cinta de trabajo para que pueda usarse en las computaciones posteriores, no se limita a preestablecer en la cinta de entrada, pero puede manejar potencialmente los infinitos flujos de entrada. Ellos afirman que la idea de una memoria persistente estaba ausente en las MT estándares; sin embargo esto contradice la idea tradicional de la persistencia (permanencia) que se tiene en computación.

Por otra parte, la conclusión de Wegner que la interacción contradice la tesis Church-Turing y que ésta es un "mito", carece de serios fundamentos.

- En el cuarto enfoque, Van Leeuwen y Wiedermann se refieren a las principales características de la computación moderna (programas no uniformes, interacción de máquinas, y operaciones infinitas). Los dos modelos de computación propuestos son la máquina de sitio y la MT interactiva con consejo. Si bien, para cada modelo considera los escenarios computacionales y las características de la máquina, una vez más, asistimos a variantes de la MT.

Los distintos intentos, directos o indirectos, de crear modelos distintos a las MT no han logrado conseguir un mayor poder expresivo que las distinta versiones de MT.

Indudablemente, la interacción proporciona una nueva conceptualización de fenómenos computacionales; sin embargo, un enfoque fundamental unificado para la computación interactiva, aún está ausente, análogo al provisto para los algoritmos por las MT y el cálculo lambda. Si los modelos propuestos no pueden computar funciones no-recursivas, entonces la tesis de Church aún estaría vigente.

La computación interactiva requiere un nuevo esquema conceptual, no sólo un planteo de modificaciones a lo que nosotros encontramos natural para la computación tradicional.

El trabajo de Turing ha fundamentado, sólidamente, la teoría de computabilidad durante casi 70 años, desechar este paradigma exige una rigurosa formalización de los modelos propuestos. Hasta ahora, sólo hay propuestas, algunas de ellas interesantes; pero, que no consiguen descartar la tesis de Church-Turing, que permanece, por ahora, firme.

Evidentemente, cuando la cantidad o la importancia de problemas no resueltos dentro de un paradigma es muy grande, puede sobrevenir una crisis y cuestionarse la validez del paradigma.

Utilizando la terminología de Kuhn, se puede afirmar que aún no se ha producido una "revolución científica", estamos en la etapa de "ensayar " teorías nuevas. Si, con el tiempo, se acepta un nuevo paradigma que sustituya al actual, estaremos en condiciones de ingresar en un período nuevo (volviendo al estado de ciencia normal) en donde se investiga el alcance del nuevo paradigma.

Por otra parte, deberíamos preguntarnos si ¿es necesario desechar el paradigma actual basado, principalmente en la MT y en la tesis de Church-Turing y sustituirlo por otro paradigma, por ejemplo el de la interacción?. O es necesario, que el paradigma que está emergiendo, se afiance, sin descartar el paradigma clásico?. La concepción tradicional de los algoritmos y las funciones recursivas, no necesariamente tienen que ser desplazada. Además, es beneficioso que coexistan más de un paradigma.

Referencias

[1] S. Anand Bhupinder. "Can Turing machines capture everything we can compute?". Disponible en URL<http://arxiv.org/ftp/math/papers/0304/0304379.pdf>.[Consultada en mayo de 2007].

[2] G. Barchini. "Informática. Una disciplina bio-psico-socio-tecno-cultural". Revista Ingeniería Informática. La Revista Electrónica del DIICC. ISSN : 0717 – 4195. Edición Número 12. Abril 2007. Disponible en URL: <http:// www.inf.udec.cl/revista>.

[3] A. Blass and Y. Gurevich. "Algorithms: A Quest for Absolute Definitions," Bull. Euro. Assoc. for Theor. Computer Science Number 81, October 2003, pages 195{225. Disponible en URL< http://research.microsoft.com/~ gurevich/Opera/164.pdf>. [Consultada en marzo de 2007].

[4] B. J. Copeland and R. Sylvan. "Beyond the Universal Turing Machine". Australasian Journal of Philosophy, 77:46-66, 1999.

[5] E. Eberbach; D. Goldin and P. Wegner. "Turing's Ideas and Models of Computation". Disponible en URL: <http://www. cse. uconn.edu/%7Edqg/papers /turing04 .pdf>.[Consultada en septiembre de 2007].

[6] D. Goldin; S. Smolka and P. Wegner. "Interactive Computing: A New Paradigm". Springer Verlag, Heidelberg, 2006.

[7] D. Goldin. " Interaction: Conjectures, Results, Myths". Disponible en URL:<http://www.cse.uconn.edu/%7Edqg/papers/ #interaction>. [Consultada en marzo de 2007].

[8] D. Goldin and P. Wegner. "The Church-Turing Thesis: Breaking the Myth". Disponible en URL: <http://www.cse.uconn.edu/%7Edqg/papers/ cie05.pdf>. [Consultada en marzo de 2007].

[9] D. Goldin and P. Wegner. "Principles of Interactive Computation". Disponible en URL< http://www.springer.com/cda/content/document.pdf >. [Consultada en septiembre de 2007].

[10] Y. Gurevich. "Sequential Abstract-State Machines-Capture Sequential Algorithms". ACM Transactions on Computational Logic, Vol. 1, No. 1, July 2000, Pages 77–111.

[11] Y. Gurevich. "Interactive Algorithms 2005 with Added Appendix" Proceedings of the 2005 conference on Mathematical Foundations of Computer Science, Springer Lecture. Disponible en: http://research.microsoft.com/~gurevich/Opera/174.pdf>. [Consultada en marzo de 2007]

[12] C. Gutiérrez. "Epistemología de la Informática". Editorial UNED. Costa Rica, 1993.

[13] J. van Leeuwen and J. Wiedermann. "The Turing Machine Paradigm in Contemporary Computing". Disponible en URL< http://www.cs.uu.nl/research/techreps/repo/CS-2000/2000-33.pdf>. [Consultada en septiembre de 2007].

[14] R. Milner. "Elements of interaction: 1991 Turing award lecture". Commun. ACM 36, 1 (Jan. 1993), 78–89.

[15] R. Milner. "Turing, Computing and Communication". Octubre 1997. Disponible en URL< http://www.fairdene.com/picalculus/milner-infomatics.pdf>. [Consultada en marzo de 2007].

[16] T. Ord. "Hypercomputation: computing more than the Turing machine". Disponible en URL< http://arxiv.org/ftp/math/papers/0209/0209332.pdf>. [Consultada en septiembre de 2007].

[17] T. Ord. "The many forms of hypercomputation". Applied Mathematics and Computation 178 (2006) 143–153. Disponible en URL<http://www. amirror clear. net/academic/papers/many-forms.pdf>. [Consultada en mayo de 2007].

[18] M. Prasse and P. Rittgen. "Why Church's Thesis Still Holds. Some Notes on Peter Wegner's Tracts on Interaction and Computability". Disponible en URL< http://www. adm .hb.se/~pri/cj98.pdf>. [Consultada en septiembre de 2007].

[19] A. M. Turing. "On computable numbers, with an application to the Entscheidungsproblem". Proceedings of London Mathematical Society, series 2, vol. 42 (1936–1937), 230–265; correction, ibidem, vol. 43, 544–546. Reprinted in [13, 155–222]. Disponible en URL< http://www.abelard.org/turpap2/tp2-ie.asp.>. [Consultada en septiembre de 2007].

[20] A. M. Turing. "Computing machinery and intelligence". Mind, 59, 433-460. Disponible en URL<http://www.loebner.net/Prizef/TuringArticle .html>. [Consultada en octubre de 2007].

[21] A. M. Turing. “Systems of Logic Based on the Ordinals”. Proceedings of the London Mathematical Society, 45:161-228, 1939.

[22] P. Wegner. “Why Interaction is More Powerful than Algorithms", Communications of ACM, May 1997, 81-91. Disponible en URL< http://portal.acm.org/citation.cfm?id =253801&coll= portal&dl=ACM>. [Consultada en septiembre de 2007].

[23] P. Wegner and D. Goldin. “Computation Beyond Turing Machines”. Communications of the ACM, 46 (4) 2003.

[24] P. Wegner and D. Goldin. “Interaction, Computability, and Church's Thesis”. Disponible en URL< http://www.cs. brown.edu/people/pw/papers/bcj1.pdf>. [Consultada en septiembre de 2007].