INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

U.A: Teoría Computacional

“Investigación 1”

Grupo: 2CV13

Profesor: De la O Torres Saul

Alumno: Sánchez Becerra Ernesto Daniel

Importancia de estudiar los autómatas

Los autómatas son esenciales para el estudio de los límites de la computación.

1) ¿Qué puede hacer una computadora?:

   Esta área de estudio se conoce como “decidibilidad”, y los problemas que una computadora puede resolver se dice que son “decidibles”.

2) ¿Qué puede hacer una computadora de manera eficiente?:

   Esta área de estudio se conoce como “intratabilidad”, y los problemas que una computadora puede resolver en un tiempo proporcional a alguna función que crezca lentamente con el tamaño de la entrada se dice que son “tratables”. Habitualmente, se supone que todas las funciones polinómicas son de “crecimiento lento”, mientras que se considera que las funciones que crecen más rápido que cualquier función polinómica crecen con demasiada rapidez.

Existen dos importantes notaciones que no son las utilizadas normalmente con los autómatas, pero que desempeñan un importante papel en el estudio de los autómatas y sus aplicaciones.

a) Gramáticas:

   Sirve para procesar datos con una estructura recursiva. Uno de los ejemplos más comunes es puede ser un “Analizador Sintáctico”

b) Expresiones Regulares:

   Especifican la estructura de los datos, especialmente de las cadenas
   de texto, los patrones de cadenas de caracteres que pueden describir
   expresiones regulares son los mismos que pueden ser descritos por los autómatas finitos.

La Teoría de Autómatas se resume a lo siguiente: es el estudio de dispositivos abstractos, es decir; el estudio de las “Maquinas”.

Esto surge debido a que en la década de los treinta Alan Turing comenzó con el estudio de una maquina abstracta o máquina de cálculo. Su principal objetivo era describir de forma precisa los límites entre lo que una máquina de cálculo podía hacer y no podía hacer.

A finales de la década de los 60’s y principios de los 70’s Cook decidió profundizar más en el estudio de Turing sobre que se podía y que no se podía calcular. Esta profundización resulto en que Cook logro separar los problemas que se podían resolver de forma eficiente con una computadora, de aquellos que eran más complejos de resolver, al igual por computadora, pero que al momento de la resolución ya no era tan eficiente, se consumía mucho tiempo y las computadoras resultaban inútiles para poder realizar la solución del problema. Este tipo de problema se denominaron como “insolubles” o “NP-Difíciles”.

Toda esta teoría desarrollada durante muchos años sirve o afecta a los expertos en computadoras hacen, términos como “Autómatas Finitos” y “Determinados tipos de Gramáticas Formales” sirve para que los expertos tomen decisiones sobre el diseño y la construcción de importantes clases de software. Otros conceptos como “Maquina de Turing” ayudan a comprender que se puede esperar cada uno de nuestros softwares. Se resume a lo siguiente: si podemos enfrentar un problema y escribir un programa para resolverlo.

¿Qué es un autómata finito?

Constituyen un modelo útil para muchos tipos de software y hardware.

1) Software para diseñar y probar el comportamiento de circuitos digitales.
2) El “analizador léxico” de un compilador típico, es decir, el componente del compilador que separa el texto de entrada en unidades lógicas, tal como identificadores, palabras clave y signos de puntuación.
3) Software para explorar cuerpos de texto largos, como colecciones de páginas web, o para determinar el número de apariciones de palabras, frases u otros patrones.
4) Software para verificar sistemas de todo tipo que tengan un número finito de estados diferentes, tales como protocolos de comunicaciones o protocolos para el intercambio seguro de información.

Existen muchos sistemas o componentes, que pueden tener una serie de “estados” finitos. El propósito de un estado es el de recordar la parte relevante del historial del sistema. Puesto que sólo existe un número finito de estados, generalmente, no es posible recordar el historial completo, por lo que el sistema debe diseñarse cuidadosamente, con el fin de recordar lo que es importante y olvidar lo que no lo es. La ventaja de disponer de sólo un número finito de estados es que podemos implementar el sistema mediante un conjunto fijo de recursos.

Tipos de demostraciones:

DEMOSTRACIÓN FORMAL: Una demostración formal se define como una secuencia de fórmulas en un lenguaje formal en la cual cada fórmula es una consecuencia lógica de las precedentes.

Se supone que la definición de demostración formal está para capturar el concepto de la demostración tal como se escribe en la práctica de la matemática.

DEMOSTRACIÓN DEDUCTIVA: Una demostración deductiva consta de una secuencia de proposiciones cuya veracidad se comprueba partiendo de una proposición inicial, conocida como hipótesis o de una serie de proposiciones dadas, hasta llegar a una conclusión. Cada uno de los pasos de la demostración, hay que deducirlo mediante algún principio lógico aceptado, bien a partir de los postulados o de algunas de las proposiciones anteriores de la demostración deductiva o de una combinación de éstas.

DEMOSTRACIÓN INDUCTIVA: Existe una forma especial de demostración, denominada “inductiva”, que es esencial a la hora de tratar con objetos definidos

de forma recursiva. Muchas de las demostraciones inductivas más habituales trabajan con enteros, pero en la teoría de autómatas, también necesitamos demostraciones inductivas, por ejemplo, para conceptos definidos recursivamente como pueden ser árboles y expresiones de diversas clases, como expresiones regulares. En la teoría de autómatas, existen varias estructuras definidas recursivamente sobre las que es necesario demostrar proposiciones. Los familiares conceptos sobre árboles y expresiones son ejemplos importantes de estas estructuras. Como las inducciones, todas las definiciones recursivas tienen un caso base, en el que se definen una o más estructuras elementales, y un paso de inducción, en el que se definen estructuras más complejas.

OTROS TIPOS DE DEMOSTRACIONES:

1) Demostraciones empleando conjuntos:
   Cuando se trabaja con autómatas; en ocasiones es necesario demostrar un teorema que establece que los conjuntos construidos de dos formas diferentes son el mismo conjunto. A menudo, se trata de conjuntos de cadenas de caracteres y se denominan “lenguajes”.
2) Demostraciones por reducción al absurdo:
   Otra forma de demostrar es utilizando el “si entonces”. Es decir, se comienza suponiendo que tanto la hipótesis como la negación de la conclusión son verdaderas. La demostración se completa probando que algo que se sabe que es falso se deduce lógicamente a partir la hipótesis y la conclusión.
3) Demostraciones mediante contraejemplo:
   En la práctica, no se habla de demostrar un teorema, sino que tenemos que enfrentarnos a algo que parece que es cierto, por ejemplo, una estrategia para implementar un programa, y tenemos que decidir si el “teorema” es o no verdadero. Para resolver este problema, podemos intentar demostrar el teorema, y si no es posible, intentar demostrar que la proposición es falsa.

Conceptos importantes

A continuación, se describen las definiciones empleadas en la teoría de autómatas:

- Alfabetos:
  Un alfabeto es un conjunto de símbolos finito y no vacío. Convencionalmente, se usa el símbolo Σ para designar un alfabeto. Entre los alfabetos más comunes se incluyen los siguientes:
  - Potencias de un Alfabeto:
    Si Σ es un alfabeto, podemos expresar el conjunto de todas las cadenas de una determinada longitud de dicho alfabeto utilizando una notación exponencial. Definimos Σ^k para que sea el conjunto

de las cadenas de longitud k, tales que cada uno de los símbolos de estas pertenece a Σ.

- Cadenas de Caracteres:
  Una cadena de caracteres (que también se denomina en ocasiones palabra) es una secuencia finita de símbolos seleccionados de algún alfabeto. Por ejemplo, 01101 es una cadena del alfabeto binario Σ = {0,1}. La cadena 111 es otra cadena de dicho alfabeto.
  - Cadena Vacía:
    La cadena vacía es aquella cadena que presenta cero apariciones de símbolos. Esta cadena, designada por ε, es una cadena que puede construirse en cualquier alfabeto.
  - Longitud de una Cadena:
    Suele ser útil clasificar las cadenas por su longitud, es decir, el número de posiciones ocupadas por símbolos dentro de la cadena. Por ejemplo, 01101 tiene una longitud de 5. Es habitual decir que la longitud de una cadena es igual al "número de símbolos" que contiene; esta proposición está aceptada coloquialmente, sin embargo, no
    es estrictamente correcta. Así, en la cadena 01101 sólo hay dos símbolos, 0 y 1, aunque tiene cinco posiciones para los mismos y su longitud es igual a 5. Sin embargo, generalmente podremos utilizar la expresión "número de símbolos" cuando realmente a lo que se está haciendo referencia es al "número de posiciones". La notación estándar para indicar la longitud de una cadena w es |w|. Por ejemplo, |011| = 3 y |ε |
  - Concatenación de Cadenas:
    Sean x e y dos cadenas. Entonces, xy denota la concatenación de x e y, es decir, la cadena formada por una copia de x seguida de una copia de y. Dicho de manera más precisa, si x es la cadena compuesta por i símbolos x = a1a2 · · ·ai e y es la cadena compuesta por j símbolos y = b1b2 · · ·bj , entonces xy es la cadena de longitud i+ j: xy = a1a2 · · ·aib1b2 · · ·bj.
- Lenguajes:
  Un conjunto de cadenas, todas ellas seleccionadas de un $\Sigma*$, donde Σ es un determinado alfabeto se denomina lenguaje. Si Σ es un alfabeto y L $\subseteq\Sigma*$, entonces L es un lenguaje de Σ. Observe que un lenguaje de Σ no necesita incluir cadenas con todos los símbolos de Σ, ya que una vez que hemos establecido que L es un lenguaje de Σ, también sabemos que es un lenguaje de cualquier alfabeto que sea un superconjunto de Σ. La única restricción importante sobre lo que puede ser un lenguaje es que todos los alfabetos son finitos. De este modo, los lenguajes, aunque pueden tener un número infinito de cadenas, están restringidos a que dichas

cadenas estén formadas por los símbolos que definen un alfabeto finito y prefijado.