

Presentación del Curso

Fabio Martínez Carrillo

Autómatas y Lenguajes Formales
Escuela de Ingeniería de Sistemas e Informatica
Universidad Industrial de Santander - UIS

31 de enero de 2018



Agenda

- 1 Información del Profesor
- 2 Sistema de Evaluación
- 3 Información Contenido del Curso
- 4 Proyectos Realizados
- 5 Algo de Historia
- 6 Bibliografía

Información del Profesor

- Nombre: Fabio Martínez Carrillo
- Correo (contacto y comunicación): famarcar@saber.uis.edu.co
- Página Web: contenido del curso
<https://sites.google.com/saber.uis.edu.co/famarcar/automatas>
- Asistente docente: Gustavo Garzón
(gustavo.garzon@saber.uis.edu.co)

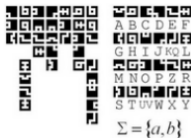
AUTÓMATAS Y LENGUAJES FORMALES



Virtual Machine

1. ALFABETOS Y LENGUAJES

- Alfabetos y lenguajes
- Introducción a Notebook, JFLAP
- Conceptos Básicos



Pres_0_Intro



Pres_1_Intro



Pres_1_2_Alfab



Notebook_MarkDown_Intro.i
nynh



Notebook_Python_Intro
1a invnh



Notebook_Python_Intro
1h invnh



Diccionarios, Funciones, Auto
matas invnh



Horarios de Clase y atención

| Horas | Martes | Miercoles | Jueves | Viernes |
|--------------|---------------------|------------------|---------------|-----------------------|
| 08-10 | | | | SEMILLERO MACV |
| 10-12 | Automatas (D1 y D2) | | | Estudiantes |
| | | | | |
| 14-16 | | | Automatas D1 | Estudiantes |
| 16-18 | | | Automatas D2 | Estudiantes |

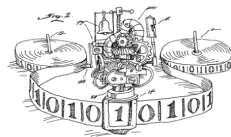
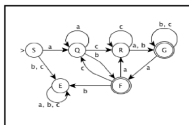
Agenda

- 1 Información del Profesor
- 2 Sistema de Evaluación**
- 3 Información Contenido del Curso
- 4 Proyectos Realizados
- 5 Algo de Historia
- 6 Bibliografía

Agenda

- 1 Información del Profesor
- 2 Sistema de Evaluación
- 3 Información Contenido del Curso**
- 4 Proyectos Realizados
- 5 Algo de Historia
- 6 Bibliografía

Definición de Automatas



Definición

La teoría de autómatas es el estudio de dispositivos de cálculo abstractos (teóricos), es decir, de las “máquinas” que:

- Describen de forma precisa los límites entre lo que una máquina puede y no puede hacer
- Son sistemas que pueden encontrarse siempre en uno de una serie de “estados” finitos

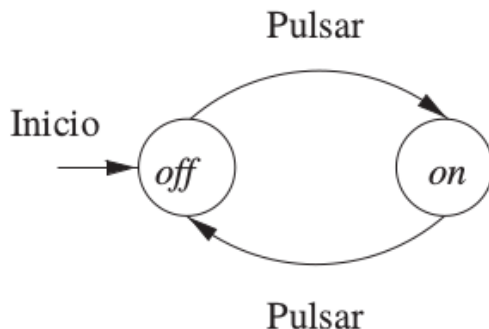
Que es un Autómata

Dispositivo mecánico o electrónico o biológico

- En un punto de tiempo está en un estado
- Dado una razón (por ejemplo una señal de entrada) cambia de estado

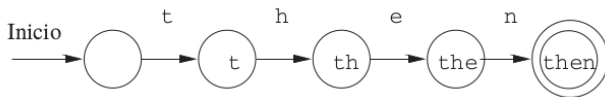
Ejemplos: reloj mecánico o electrónico, máquina para lavar, todo un ordenador

Autómata finito mas simple



Interruptor (encendido/apagado *on/off*)

- El dispositivo *recuerda* si el estado es *on* o *off*
- Los estados están representados mediante círculos
- Los arcos son “entradas” externas que influyen en el sistema
- Estado inicial indicado por la palabra *inicio*



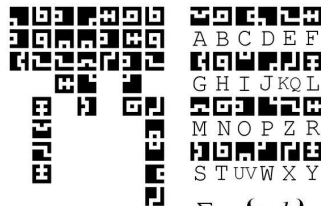
Analizador léxico

Reconoce la palabra clave *then*

- Cada estado corresponde a los prefijos de las palabras
- Estado de aceptación se representa con un círculo doble

Alfabetos y Lenguajes

- Introducción a Autómatas Finitos
- Demostraciones formales
- Alfabetos, cadenas de caracteres y lenguajes



$$\Sigma = \{a, b\}$$

$$\Sigma = \{a, b\}$$

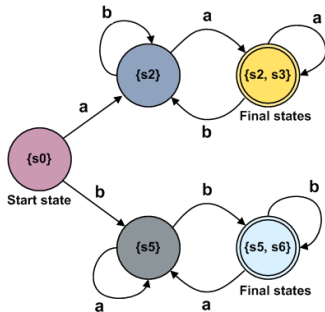
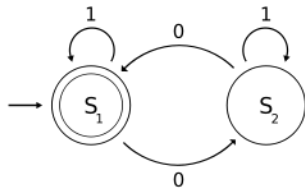
$$u = ab$$

$$v = bbbaaa$$

$$w = abba$$

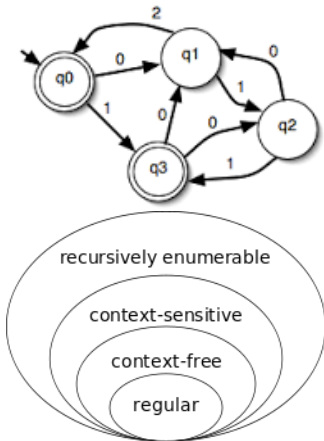
Autómatas Finitos

- Definición de autómatata
- Automata finito determinista
- Automatas finitos no deterministas
- Automatas finitos con transacciones.



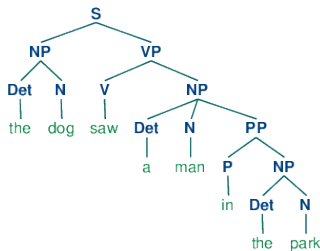
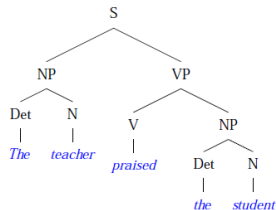
Lenguajes Regulares

- Automatas finitos y expresiones regulares
- Algebra de las expresiones regulares
- Propiedades de los lenguajes regulares.



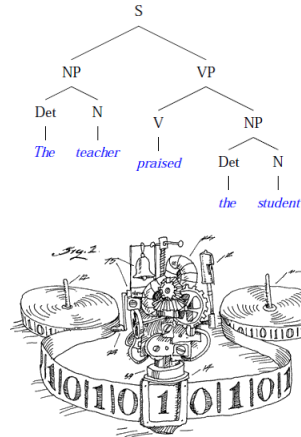
Lenguajes independientes del contexto

- Gramática independiente del contexto
- Árboles de derivación
- Automatas de Pila



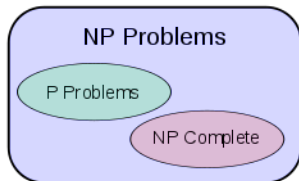
Maquinas de Turing (MT)

- Introducción a la MT
- Técnicas de programación para la MT
- MT restringidas
- Indecidibilidad



Problemas Intratables

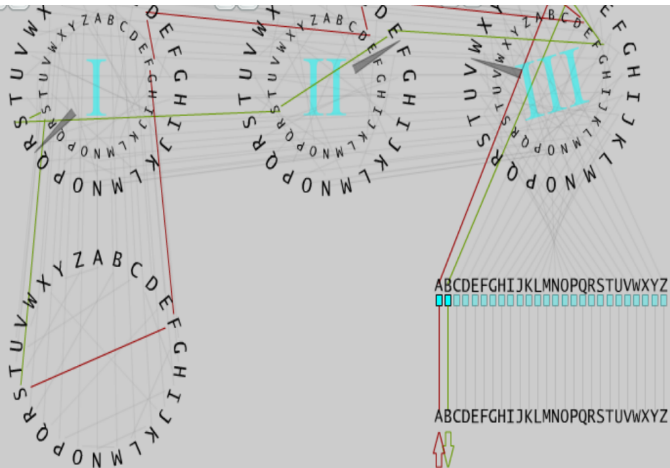
- Las clases P y NP
- problemas NP completos
- Problemas resolubles en espacio polinómico
- Problemas PS-completos



Agenda

- 1 Información del Profesor
- 2 Sistema de Evaluación
- 3 Información Contenido del Curso
- 4 Proyectos Realizados**
- 5 Algo de Historia
- 6 Bibliografía

Proyecto Enigma



Input:

A

Output:

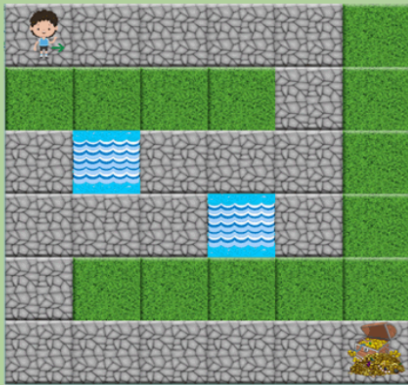
B

PEDRO

```
codigo{  
  repetir(4){  
    mover;  
  }  
  girar(abajo);  
  repetir(2){  
    mover;  
  }  
  girar(izquierda);  
  mover;  
  mover;  
  girar(abajo);  
  mover;  
  girar(izquierda);  
  mover;  
}
```



LEARN TO CODE



Autómatas y Lenguajes Formales

PROYECTO DE AUTÓMATAS JUEGO DEL AHORCADO

ALUMNOS:

- ANDRES CAMILO HERNÁNDEZ ARIAS. COD:2130284
- JULIANA ANDREA JIMENEZ BUITRAGO. COD:2140408

Profesor: Fabio Martínez

```

+---+
|   |
0   |
/|\  |
=====
''' , '''

```

```

+---+
|   |
0   |
/|\  |
/   |
=====
''' , '''

```

```

+---+
|   |
0   |
/|\  |
/   |
=====
''' ]

```

```

claves = []
for sentence in (generate(grammar)):
    claves.append(sentence)

def obtenerClaveAzar (claves):
    indiceClave = ' '.join(claves[random.randint(0, len(claves) -1)])
    return indiceClave

def mostrarTablero(imagenes_juego, letrasIncorrectas, letrasCorrectas, palabraClave):
    print(imagenes_juego[len(letrasIncorrectas)])
    print ""
    print 'Letras incorrectas: '
    for letra in letrasIncorrectas:
        print letra,
    print

```

MÁQUINA DE TURING PARA DERIVAR MONÓMIOS Y AUTÓMATA FINITO DETERMINISTA ÉPSILON QUE MUESTRA EL ORDEN DE DERIVACIÓN DE UNA EXPRESIÓN

| Slide Type | Slide |
|----------------|-------|
| Section Header | 1 |
| Text | 2 |
| Text | 3 |
| Text | 4 |
| Text | 5 |
| Text | 6 |
| Text | 7 |
| Text | 8 |
| Text | 9 |
| Text | 10 |
| Text | 11 |
| Text | 12 |
| Text | 13 |
| Text | 14 |
| Text | 15 |
| Text | 16 |
| Text | 17 |
| Text | 18 |
| Text | 19 |
| Text | 20 |
| Text | 21 |
| Text | 22 |
| Text | 23 |
| Text | 24 |
| Text | 25 |
| Text | 26 |
| Text | 27 |
| Text | 28 |
| Text | 29 |
| Text | 30 |
| Text | 31 |
| Text | 32 |
| Text | 33 |
| Text | 34 |
| Text | 35 |
| Text | 36 |
| Text | 37 |
| Text | 38 |
| Text | 39 |
| Text | 40 |
| Text | 41 |
| Text | 42 |
| Text | 43 |
| Text | 44 |
| Text | 45 |
| Text | 46 |
| Text | 47 |
| Text | 48 |
| Text | 49 |
| Text | 50 |
| Text | 51 |
| Text | 52 |
| Text | 53 |
| Text | 54 |
| Text | 55 |
| Text | 56 |
| Text | 57 |
| Text | 58 |
| Text | 59 |
| Text | 60 |
| Text | 61 |
| Text | 62 |
| Text | 63 |
| Text | 64 |
| Text | 65 |
| Text | 66 |
| Text | 67 |
| Text | 68 |
| Text | 69 |
| Text | 70 |
| Text | 71 |
| Text | 72 |
| Text | 73 |
| Text | 74 |
| Text | 75 |
| Text | 76 |
| Text | 77 |
| Text | 78 |
| Text | 79 |
| Text | 80 |
| Text | 81 |
| Text | 82 |
| Text | 83 |
| Text | 84 |
| Text | 85 |
| Text | 86 |
| Text | 87 |
| Text | 88 |
| Text | 89 |
| Text | 90 |
| Text | 91 |
| Text | 92 |
| Text | 93 |
| Text | 94 |
| Text | 95 |
| Text | 96 |
| Text | 97 |
| Text | 98 |
| Text | 99 |
| Text | 100 |

MOTIVACIÓN

Como se aprendió en el curso de Autómatas y Lenguajes Formales, usando Máquinas de Turing y Autómatas se pueden resolver todos los problemas que un computador normal resuelve, y aunque estos problemas pueden ser más o menos complejos computacionalmente en una máquina o en un autómata, es importante clarificar el funcionamiento básico de estos temas aprendidos.

La derivación es un tema difícil para los estudiantes de grados décimo y once de los colegios de la ciudad. Entonces esta herramienta nace como una solución didáctica para estos, al poder ellos a través de esta corregir los errores que tengan a la hora de derivar monómios y dándoles instrucciones precisas a la hora de derivar expresiones más complejas.

Maquina de Turing que resuelve sistemas de ecuaciones lineales de dos variables con dos incognitas

Profesor:

Fabio Martínez Carillo

Estudiantes:

David Villabona Ardila

Juan Felipe Chacón López

Escuela de Ingeniería de Sistemas e Informática

Autómatas y Lenguajes Formales

24 de Julio del 2017

Esta es una maquina de Turing que permite encontrar el valor de la variable y para un sistema de ecuaciones lineales con variables x y y .

Para encontrar el valor de la variable y se utiliza el método matematico de reducción de incognitas.

La maquina de turing está definida en un sistema unario y los simbolos de cinta son los siguientes:

"s": Indica el signo positivo del termino

"r": Indica el signo negativo del termino

"a": Simbolo que indica donde termina una ecuacion y empieza la otra

"/": Indica división entre dos números

"i": Indica el simbolo "=" en la ecuación

" " Indica el espacio en la cinta

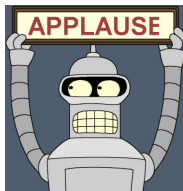
Agenda

- 1 Información del Profesor
- 2 Sistema de Evaluación
- 3 Información Contenido del Curso
- 4 Proyectos Realizados
- 5 Algo de Historia**
- 6 Bibliografía

Informática: información automática

Teoría de la computabilidad

- Origen: toda clase de problemas pudiese ser resuelto desde un esquema general: **máquina automática**
 - Que pueden hacer los computadores?
 - Cuales son los límites inherentes de las máquinas



D. Hilbert: Entscheidungsproblem (1928)

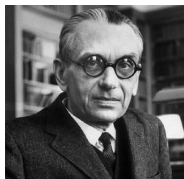


Problema de decisión

- Las matemáticas son completas?. En el sentido que pueda probarse cada aseveración matemática
- Las matemáticas son consistentes? . Puede probarse cada aseveración y su negación
- Las matemáticas son decidibles? Existe un método que pueda aplicarse y que determine la aseveración

Objetivo: crear un sistema matemático completo y consistente.

K Godel: Teorema de la incompletitud (1931)



*“ Todo sistema de primer orden que contenga teoremas de la aritmética y cuyo conjunto de axiomas sea recursivo **no es completo**”*

- Si los axiomas de dicha teoría no se contradicen entre sí, entonces existen enunciados que no pueden probarse ni refutarse a partir de ellos.
- La conclusión del teorema se aplica siempre que la teoría aritmética en cuestión sea recursiva

Church λ -definible (1936)



λ -definible

- Propone una función calculable
- La demostración de teoremas se convierte en una transformación de una cadena de símbolos según un conjunto de reglas
- El sistema fue inconsistente



Clausura de Kleene

- una operación unaria que se aplica sobre un conjunto de cadenas de caracteres o un conjunto de símbolos
- Demuestra la equivalencia entre las funciones λ y las funciones de Godel

A. Turing (1936)



Maquinas abstractas

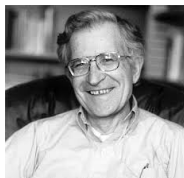
- Tesis de Turing: funciones calculables mediante un algoritmo
- Utilizo el concepto de máquina para demostrar que hay funciones no calculables
- La tesis de Turing es equivalente a la de Church

En los 60

- Se separan los conceptos de las implementaciones
- Aparecen los lenguajes de alto nivel como Fortran
- Se estudian algoritmos independientes del ordenador
- Aparece la necesidad de traducir lenguajes de alto nivel a lenguaje de Máquina

Desde los años 40 se ha intentado imitar funciones del cerebro biológico. Se han desarrollado máquinas capaces de aprender y reproducir funciones o comportamientos. Estas máquinas se llaman redes Neuronales.

Chomsky



Propone tres modelos para la descripción de lenguajes que ayudo al desarrollo de programación

- Define la diferencia entre automáatas y gramáticas

Agenda

- 1 Información del Profesor
- 2 Sistema de Evaluación
- 3 Información Contenido del Curso
- 4 Proyectos Realizados
- 5 Algo de Historia
- 6 Bibliografía**

Libros

- KELLEY, Dean. Teoría de autómatas y lenguajes formales, Prentice Hall.
- HOPCROFT, J.E., and J. D. ULLMAN. Teoría de autómatas, lenguajes y computación. Pearson 2007
- SUDKAMP, T. Languages and Machines, Addison-Wesley Publishing Company, Inc, Reading, Mass, 1988.
- BRENA RAMON. Autómatas y lenguajes. 2003

Cursos e información en linea

Stanford | ONLINE

[COURSES](#) [ABOUT](#) [ACROSS CAMPUS](#) [VPTL](#)

[Home](#) » [Courses](#) » Automata

AUTOMATA

Date:

Wednesday, June 13, 2012

[Go to Course](#)

Course number:

CS154

<http://online.stanford.edu/course/automata>

Cursos e información en línea

Home > Courses > Electrical Engineering and Computer Science > Automata, Computability, and Complexity

Automata, Computability, and Complexity

COURSE HOME <

- SYLLABUS
- CALENDAR
- LECTURE NOTES
- ASSIGNMENTS
- DOWNLOAD COURSE MATERIALS

Instructor(s)
Prof. Scott Aaronson

MIT Course Number
6.045J / 18.400J

As Taught In
Spring 2011

Level
Undergraduate

[CITE THIS COURSE](#)

<https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-045j-automata-computability-and-complexity-spring-2011/>

Muchas gracias por su atención

