Autómatas y Gramáticas

DOCENTES:

DR. ING. ALFREDO IGLESIAS

MG. ING. NORA COSTA

PRERREQUISITOS

Para aprovechar este curso, los estudiantes deberían conocer previamente los siguientes temas:

1. Matemática discreta.
2. Grafos.
3. Árboles.
4. Lógica.
5. Programación utilizando el lenguaje Phyton.
6. Estructuras de datos, como: pilas y colas.
7. Conceptos de recursión.

BIBLIOGRAFÍA

- **John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman**. "**TEORÍA DE AUTÓMATAS, LENGUAJES Y COMPUTACIÓN**". Pearson Addison Wesley. (Tercera Edición).
- **A. Aho, R.Sethi, J. Ullmam.** "**COMPILADORES PRINCIPIOS, TECNICAS Y HERRAMIENTAS**". Pearson Addison Wesley. (Primera y Segunda Edición).
- **Dean Kelley.** "**TEORÍA DE AUTÓMATAS Y LENGUAJES FORMALES**". Prentice Hall (Segunda Edición).

CRONOGRAMA DE CLASES 2021

15-mar	Introducción a la materia. Introducción a Autómatas Finitos
22-mar	Autómatas Finitos Deterministas. Autómatas Finitos No Deterministas. Aplicaciones.
29-mar	Lenguajes y expresiones regulares. Operaciones con expresiones regulares Autómatas Finitos y Expresiones Regulares. Aplicaciones.
5-abr	Trabajo Práctico Nº 1: Implementación con Python para validación de datos mediante Expresiones Regulares.
12-abr	Trabajo Práctico Nº 2: Autómatas de estados finitos.
19-abr	Lenguajes y Gramáticas independientes del contexto. Derivaciones. Árboles Sintácticos. Gramáticas Ambiguas
26-abr	Trabajo Práctico Nº 3: Programación de analizadores sintácticos en Phyton
3-may	Máquinas de Turing. Definición formal. Diagrama de estados. Lenguajes asociados. Restricciones. Máquinas de Turing no deterministas. Implementación de una máquina de Turing a partir de un autómata finito determinista.
10-may	Trabajo Práctico Nº 4: Máquinas de Turing.
17-may	Mesa especiales de Mayo
24-may	Feriado con fines turísticos (Día de la Revolución de Mayo)
31-may	Trabajo Práctico Nº 4: Análisis de datos con Phyton sobre una base de datos de tráfico de wifi.
7-jun	Consulta Trabajo Práctico Nº 4.
14-jun	Consulta Trabajo Práctico Nº 4.

REQUISITOS DE REGULARIDAD

Para obtener la regularidad se requiere:

1. Presentación de los trabajos prácticos 1, 2 y 3 en las fechas establecidas.
2. Fechas límites para la presentación de Prácticos:
    1. Trabajo Práctico Nº 1: 19/04/2021.
    2. Trabajo Práctico Nº 2: 03/05/2021.
    3. Trabajo Práctico Nº 3: 17/05/2021.
3. Asistencia a las clases virtuales.
4. La regularidad en la materia tiene una duración de dos años.

APROBACIÓN DE LA ASIGNATURA

El examen final de la materia consistirá en la presentación del Trabajo Práctico Nº 4 en forma sincrónica (google meet) y preguntas sobre como fue desarrollado.

Para aprobar la materia, se requiere:

1. Obtener la regularidad de la materia.
2. Desarrollar el Trabajo Práctico Nº 4, siguiendo las condiciones establecidas, presentándolo previamente en clases de consultas.
3. Inscribirse en uno de los turnos de exámenes que fije la Facultad.
4. Presentarse el día de la mesa de examen en la que se ha inscripto (google meet), explicar, en forma individual, como se desarrolló el proyecto, estructura, métodos utilizados, programación, etc., y responder las preguntas realizadas por los docentes.

DESARROLLO DE LOS TRABAJOS PRÁCTICOS

1. Los Trabajos Prácticos pueden realizarse individualmente, o en grupos de no más de tres estudiantes por grupo.
2. La conformación de estos grupos puede cambiar de un trabajo práctico a otro.
3. Tengan en cuenta que la condición de regularidad es individual, es decir solo la alcanzan aquellos estudiantes que presentaron y aprobaron los prácticos 1,2 y 3, independientemente del grupo con el que los realizaron.

INTRODUCCIÓN

LA TEORÍA DE AUTÓMATAS

La teoría de autómatas es el estudio de dispositivos de cálculo abstractos, es decir, de las “máquinas”.

LA MÁQUINA ABSTRACTA

Antes de que existieran las computadoras, en la década de 1930, A. Turing estudió una máquina abstracta que tenía todas las capacidades de las computadoras de hoy día.

- **Sugerencia:** ver película “The Imitation Game” 2014.

La historia nos narra como en plena Segunda guerra mundial, el matemático Alan Turing junto a un grupo de especialistas intentan descifrar el código secreto Alemán conocido como Enigma. Para esto Turing tendrá que lidiar con su carácter y un secreto de su vida privada.

https://www.youtube.com/watch?v=0DFqSmdbsUM

INTRODUCCIÓN

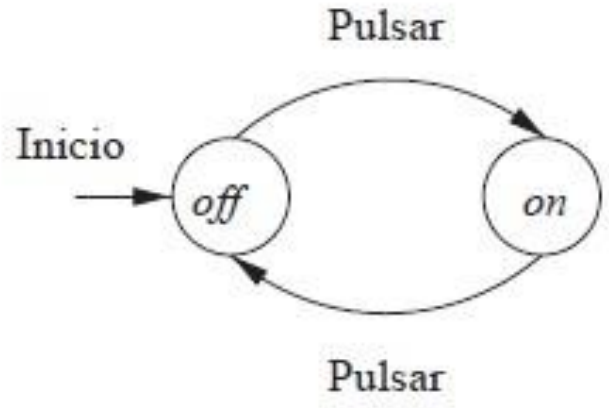
- **Década de 1930**. A. Turing estudió una **máquina abstracta** que tenía todas las capacidades de las computadoras de hoy día.
- **Décadas de los años 1940 y 1950**. Una serie de investigadores estudiaron las máquinas más simples, las cuales todavía hoy denominamos "**autómatas finitos**". Originalmente, estos autómatas se propusieron para modelar el funcionamiento del cerebro y, posteriormente, resultaron extremadamente útiles para otros propósitos.
- **1969**. S. Cook amplió el estudio realizado por Turing sobre lo que se podía y no se podía calcular.
- Hay problemas que, en principio, **pueden resolverse**, pero que **en la práctica consumen** tanto **tiempo** que las computadoras resultan inútiles para todo excepto para casos muy simples del problema. Este último tipo de **problemas** se denominan "**insolubles**" o "**NPdifíciles** ".

INTRODUCCIÓN

- Los **autómatas finitos** son un modelo útil para muchos tipos de hardware y software, los tipos más importantes son:
  - **Software** para diseñar y probar el comportamiento de **circuitos digitales**.
  - El "**analizador léxico**" de un compilador típico, es decir, el componente del compilador que separa el texto de entrada en unidades lógicas (tokens), tal como identificadores, palabras clave y signos de puntuación.
  - **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.
  - **Software** para verificar sistemas que tengan un **número finito de estados diferentes**, tales como protocolos de comunicaciones o protocolos para el intercambio seguro de información.

INTRODUCCIÓN

- Ejemplo:
  - El autómata finito más simple puede ser un **interruptor de apagado/encendido** (posiciones on/off).
  - El dispositivo recuerda si está en el estado encendido (“on”) o en el estado apagado (“off”).



CONCEPTOS FUNDAMENTALES

- Alfabetos.
- Cadenas de caracteres.
- La cadena vacía.
- Longitud de una cadena.
- Potencias de un alfabeto.
- Concatenación de cadenas.
- Potencia.

ALFABETOS

- Un alfabeto es un conjunto de símbolos finito y no vacío.
- Convencionalmente, utilizamos el símbolo ∑ para designar un alfabeto.
- Entre los alfabetos más comunes se incluyen los siguientes:
  1. ∑ = {0,1}, el alfabeto binario.
  2. ∑ = {a,b, . . . , z}, el conjunto de todas las letras minúsculas.
  3. ∑ = {0,1, . . . ,9}, el conjunto de los dígitos del 0 al 9.
  4. El conjunto de todos los caracteres ASCII o el conjunto de todos los caracteres ASCII imprimibles.

ASCII (American Standard Code for Information Interchange)

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 $\sum = \{0,1\}$.
  - 111 es otra cadena de dicho alfabeto.

LA 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.
- La notación estándar para indicar la longitud de una cadena w es |w|.
- Por ejemplo:
  - |011| = 3

POTENCIAS DE UN ALFABETO

- Si ∑ es un alfabeto, podemos expresar el conjunto de todas las cadenas de una determinada longitud de dicho alfabeto utilizando una notación exponencial.
- Definimos para que sea el conjunto de las cadenas de longitud k, tales que cada uno de los símbolos de las mismas pertenece a ∑ Observe que ∑ 0 = ε independientemente de cuál sea el alfabeto ∑ es decir, ε es la única cadena cuya longitud es 0.
- Si ∑ = {0,1}, entonces:
  - $\sum^0 = \varepsilon$
  - $\sum^1 = \{0,1\}$
  - $\sum^2 = \{00,01,10,11\}$
  - $\sum^3 = \{000,001,010,011,100,101,110,111\}$

POTENCIAS DE UN ALFABETO

- El conjunto de todas las cadenas de un alfabeto ∑ se designa mediante ∑* . Por ejemplo:
  - ∑* = { ε ,0,1,00,01,10,11,000, . . .}
  - Expresado de otra forma:
  - ∑* = ∑ 0 ∑ 1 ∑2 · · ·
- En ocasiones, desearemos excluir la cadena vacía del conjunto de cadenas. El conjunto de cadenas no vacías del alfabeto ∑ se designa como ∑+ . Por tanto, dos equivalencias apropiadas son:
  - ∑ += ∑ 1 ∑2 · · ·
  - ∑ *= ∑ + { 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.

POTENCIA

- Si w = 122 sobre el alfabeto Σ = { 1, 2 }
  - $w^0 = \varepsilon$
  - $w^1 = 122$
  - $w^2 = 122122$
  - $w^3 = 122122122$

$$w^n = \begin{cases} \varepsilon & n = 0 \\ w w^{n-1} & n > 0 \end{cases}$$