Introducción a la Teoría de Autómatas, Lenguajes y Computación

Gustavo Rodríguez Gómez y Aurelio López López

INAOE

Propedéutico 2010

Capítulo 1
Introducción
Autómatas

Libro de texto

- John E. Hopcroft et al., "Introduction to Automata Theory, Languajes, and Computation", segunda edición, Addison Wesley

1. Introducción
   - Motivación
2. ¿Por qué estudiar Autómatas?
   - Introducción a los Autómatas Finitos
3. Introducción a las Demostraciones Formales
   - Condiciones Necesarias y Suficientes
4. Conceptos Centrales de la TA
   - Alfabetos
   - Cadenas
   - Lenguajes
   - Problemas

Motivación

- La teoría de autómatas es el estudio de dispositivos (*máquinas*) de computación abstractos.

Motivación

- La teoría de autómatas es el estudio de dispositivos (*máquinas*) de computación abstractos.
- Objetivo (Turing):

Motivación

- La teoría de autómatas es el estudio de dispositivos (*máquinas*) de computación abstractos.
- Objetivo (Turing):
  - Describir en forma precisa la frontera entre lo que una *computadora* puede hacer y lo que no puede hacer.

Motivación

- La teoría de autómatas es el estudio de dispositivos (*máquinas*) de computación abstractos.
- Objetivo (Turing):
  - Describir en forma precisa la frontera entre lo que una *computadora* puede hacer y lo que no puede hacer.
- Entre 1940 y 1950 surgen las máquinas hoy llamadas "autómatas finitos".

Motivación

- La teoría de autómatas es el estudio de dispositivos (*máquinas*) de computación abstractos.
- Objetivo (Turing):
  - Describir en forma precisa la frontera entre lo que una *computadora* puede hacer y lo que no puede hacer.
- Entre 1940 y 1950 surgen las máquinas hoy llamadas “autómatas finitos”.
- A finales de los 1950´s el lingüista Chomsky inicia el estudio formal de las “gramáticas”.

Motivación

- En 1969 S. Cook pudo clasificar los problemas que pueden ser resueltos en una computadora en dos categorías:

Motivación

- En 1969 S. Cook pudo clasificar los problemas que pueden ser resueltos en una computadora en dos categorías:
  - problemas que se pueden resolver en forma eficiente,

Motivación

- En 1969 S. Cook pudo clasificar los problemas que pueden ser resueltos en una computadora en dos categorías:
  - problemas que se pueden resolver en forma eficiente,
  - problemas que en principio se pueden resolver pero que en la práctica consumen mucho tiempo (NP–duros).

Motivación

- En 1969 S. Cook pudo clasificar los problemas que pueden ser resueltos en una computadora en dos categorías:
  - problemas que se pueden resolver en forma eficiente,
  - problemas que en principio se pueden resolver pero que en la práctica consumen mucho tiempo (NP–duros).
- Todos los desarrollos teóricos se apoyan en lo que los científicos de la computación desarrollan actualmente.

Motivación

- Los autómatas finitos y las gramáticas formales se usan en el diseño y construcción de software.

Motivación

- Los autómatas finitos y las gramáticas formales se usan en el diseño y construcción de software.
- La máquinas de Turing nos ayuda a entender lo que podemos esperar de nuestro software.

Motivación

- Los autómatas finitos y las gramáticas formales se usan en el diseño y construcción de software.
- La máquinas de Turing nos ayuda a entender lo que podemos esperar de nuestro software.
- La teoría de problemas intratables nos ayuda a deducir si nos enfrentamos con problemas tratable o no.

Introducción a los Autómatas Finitos

- Algunas aplicaciones de autómatas finitos

Introducción a los Autómatas Finitos

- Algunas aplicaciones de autómatas finitos
    - Diseño de software y verificación del comportamiento de circuitos digitales.

Introducción a los Autómatas Finitos

- Algunas aplicaciones de autómatas finitos
  - Diseño de software y verificación del comportamiento de circuitos digitales.
  - Analizadores léxicos de compiladores.

Introducción a los Autómatas Finitos

- Algunas aplicaciones de autómatas finitos
  - Diseño de software y verificación del comportamiento de circuitos digitales.
  - Analizadores léxicos de compiladores.
  - Software para explorar grandes volúmenes de texto y encontrar patrones.

Introducción a los Autómatas Finitos

- Algunas aplicaciones de autómatas finitos
  - Diseño de software y verificación del comportamiento de circuitos digitales.
  - Analizadores léxicos de compiladores.
  - Software para explorar grandes volúmenes de texto y encontrar patrones.
  - Software para verificar sistemas que tengan un número finito de estados, por ejemplo, protocolos de comunicación.

Condiciones Necesarias

Definición

Una proposición P es una condición necesaria de una proposición Q si

$$Q \Rightarrow P$$

Ejemplo

Una condición necesaria para que un humano este vivo es que respire.

Ejemplo

Una condición necesaria para ser presidente de México es tener 35 años o más.

Condiciones Suficientes

Definición

Una proposición P es una condición suficiente de una proposición Q si

$$Q \Leftarrow P$$

Ejemplo

Ser mamífero es una condición necesaria pero no suficiente para ser humano.

Ejemplo

Ser un número racional es una condición suficiente para ser un número real.

Alfabetos

- Un alfabeto

$$\Sigma = \text{ conjunto de símbolos finito no vacío}$$

Alfabetos

- Un alfabeto

$$\Sigma = \text{ conjunto de símbolos finito no vacío}$$

- Ejemplos

$$\Sigma = \{0, 1\}, \text{ el alfabeto binario,}$$
$$\Sigma = \{a, b, c, \ldots, z\}, \text{ conjunto de todas las letras minúsculas}$$

Cadenas

- Una cadena w es una sucesión finita de símbolos de algún alfabeto Σ

$$w = a_1 a_2 \cdots a_i, \quad a_k \in \Sigma \quad k = 1, 2, \ldots, i$$

Cadenas

- Una cadena w es una sucesión finita de símbolos de algún alfabeto Σ

$$w = a_1 a_2 \cdots a_i, \quad a_k \in \Sigma \quad k = 1, 2, \ldots, i$$

- La cadena vacía ϵ es la cadena con cero ocurrencias de símbolos de Σ.

Cadenas

- Una cadena w es una sucesión finita de símbolos de algún alfabeto Σ

$$w = a_1 a_2 \cdots a_i, \quad a_k \in \Sigma \quad k = 1, 2, \ldots, i$$

- La cadena vacía ϵ es la cadena con cero ocurrencias de símbolos de Σ.
- La longitud de una cadena $w = a_1 a_2 \cdots a_i$ es el número de posiciones de los símbolos de la cadena

$$|w| = i$$

Cadenas

- Una cadena w es una sucesión finita de símbolos de algún alfabeto Σ

$$w = a_1 a_2 \cdots a_i, \quad a_k \in \Sigma \quad k = 1, 2, \ldots, i$$

- La cadena vacía ϵ es la cadena con cero ocurrencias de símbolos de Σ.
- La longitud de una cadena $w = a_1 a_2 \cdots a_i$ es el número de posiciones de los símbolos de la cadena

$$|w| = i$$

- Sea Σ un alfabeto, definimos las potencias de Σ por

$$\Sigma^k = \{w | w \text{ es una cadena de } \Sigma \text{ y } |w| = k\},$$

donde $k \geq 0$ entero.

Ejemplos potencias de un alfabeto

Ejemplos

$\Sigma^0 = \{\epsilon\}$ para cualquier alfabeto
Si $\Sigma = \{0, 1\}$

$$\Sigma^1 = \{0, 1\},$$
$$\Sigma^2 = \{00, 01, 10, 11\},$$
$$\Sigma^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$$

Problema

¿Cuál es la diferencia entre Σ y Σ^1?

Conjunto de todas las cadenas de un alfabeto

- El conjunto de todas las cadenas de un alfabeto Σ es denotado por Σ^*

$$\begin{aligned}\Sigma^* &= \{w|w \text{ es cadena de } \Sigma\}\,, \\ &= \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \cdots\end{aligned}$$

Conjunto de todas las cadenas de un alfabeto

- El conjunto de todas las cadenas de un alfabeto Σ es denotado por Σ^*

$$\begin{aligned}\Sigma^* &= \{w | w \text{ es cadena de } \Sigma\}, \\ &= \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \cdots\end{aligned}$$

- Ejemplo

$$\{0,1\}^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, \cdots\}$$

Concatenación de cadenas

- El conjunto de cadenas no vacías de un alfabeto se define por

$$\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \cdots$$

Concatenación de cadenas

- El conjunto de cadenas no vacías de un alfabeto se define por

$$\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \cdots$$

- Sean $x = a_1 a_2 \cdots a_i$, $y = b_1 b_2 \cdots b_j$ cadenas de Σ, definimos la concatenación de x con y por

$$xy = a_1 a_2 \cdots a_i b_1 b_2 \cdots b_j,$$

Concatenación de cadenas

- El conjunto de cadenas no vacías de un alfabeto se define por

$$\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \cdots$$

- Sean $x = a_1 a_2 \cdots a_i$, $y = b_1 b_2 \cdots b_j$ cadenas de Σ, definimos la concatenación de x con y por

$$xy = a_1 a_2 \cdots a_i b_1 b_2 \cdots b_j,$$

- La longitud de la nueva cadena es

$$|xy| = i + j$$

Lenguajes

Definición

Sea Σ un alfabeto y $L \subseteq \Sigma^*$ diremos entonces que L es un lenguaje de Σ.

Problemas en teoría de autómatas

Definición

Sea Σ un alfabeto y L un lenguaje de Σ.El problema L es:

- Dado una cadena $w \in \Sigma^*$ decidir si $w \in L$ ó $w \notin L$.