Palabras y autómatas

Clase 01

IIC 2223

Prof. Cristian Riveros

Outline

Outline

Alfabetos, letras y palabras

Definiciones

- Un **alfabeto** Σ es un conjunto finito.
- Un elemento de Σ lo llamaremos una **letra** o **símbolo**.
- Una **palabra** o **string** sobre Σ es una secuencia finita de letras en Σ.

Ejemplo

- $\Sigma = \{ a, b, c \}$
- Palabras sobre Σ:

$$aaaaabb \ , \ bcaabab \ , \ a \ , \ bbbbbb \ , \ \ldots$$

¿cuál es el alfabeto preferido en computación?

Alfabetos, letras y palabras

Más definiciones

- El **largo** $|w|$ de una palabra w es el número de letras.

$$|w| \overset{\text{def}}{\equiv} \#\text{ de letras en } w$$

- Denotaremos ϵ como la **palabra sin símbolos** de largo 0.

$$|\epsilon| \overset{\text{def}}{\equiv} 0$$

- Denotaremos por Σ^* como el **conjunto de todas las palabras** sobre Σ.

Ejemplo

Para $\Sigma = \{0, 1\}$:

- $|00011001| = ?$
- $\Sigma^* = ?$

Concatenación entre palabras

Definición

Dado dos palabras $u, v \in \Sigma^*$ tal que $u = a_1 \dots a_n$ y $v = b_1 \dots b_m$:

$$u \cdot v \overset{\text{def}}{\equiv} a_1 \dots a_n \, b_1 \dots b_m$$

Decimos que $u \cdot v$ es la palabra "u **concatenada** con v".

Ejemplo

Para $\Sigma = \{0, 1, 2, \dots, 9\}$:

- $0123 \cdot 9938 \;=\; ?$
- $3493 \cdot \epsilon \;=\; ?$

Concatenación sobre palabras

Definición

Dado dos palabras $u, v \in \Sigma^*$ tal que $u = a_1 \dots a_n$ y $v = b_1 \dots b_m$:

$$u \cdot v \stackrel{\text{def}}{\equiv} a_1 \dots a_n b_1 \dots b_m$$

Decimos que $u \cdot v$ es la palabra "u **concatenada** con v".

Algunas propiedades:

- ¿es la concatenación **asociativa**: $(u \cdot v) \cdot w = u \cdot (v \cdot w)$?
- ¿es la concatenación **conmutativa**: $u \cdot v = v \cdot u$?
- ¿es verdad que $|u \cdot v| = |u| + |v|$?

Lenguajes

Definición

Sea Σ un alfabeto y $L \subseteq \Sigma^*$.

Decimos que L es un **lenguaje** sobre el alfabeto Σ.

Ejemplos de lenguajes

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

- $L_0 \ = \ \{\epsilon, a, aa, b, ba\}$
- $L_1 \ = \ \{\epsilon, b, bb, bbb, bbbb, \ldots\}$
- $L_2 \ = \ \{w \mid \exists u \in L_1.\ w = a \cdot u\}$
- $L_3 \ = \ \{w \mid \exists u, v \in \Sigma^*.\ w = u \cdot abba \cdot v\}$
- $L_4 \ = \ \{w \mid \exists u \in \Sigma^*.\ w = u \cdot u\}$

Un **lenguaje** puede ser visto como una **propiedad** de palabras

Ocuparemos estas definiciones durante TODO el curso

Convenciones

Durante todo el curso:

- Para **letras** usaremos los símbolos: $a, b, c, d, e, \ldots$
- Para **palabras** usaremos los símbolos: $w, u, v, x, y, z, \ldots$
- Para **alfabetos** usaremos los símbolos: $\Sigma,\ \Gamma, \ldots$
- Para **lenguajes** usaremos los símbolos: $L, M, N, \ldots$
- Para **números** usaremos los símbolos: $i, k, j, l, m, n, \ldots$

No olvidar!

Outline

Autómatas finitos

- Modelo de computación más sencillo,
  basado en una cantidad **finita** de memoria.

- Procesa el input de principio a fin en **una sola pasada**.

- Al terminar, el autómata decide si **acepta** o **rechaza** el input.

Usaremos los autómatas finitos para definir **lenguajes**

Autómata finito determinista

Definición

Un autómata finito determinista (DFA) es una estructura:

$$\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$$

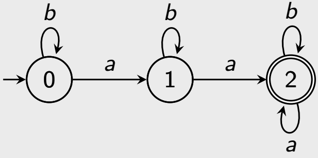
- Q es un conjunto finito de **estados**.
- Σ es el alfabeto de **input**.
- $\delta : Q \times \Sigma \rightarrow Q$ es la función de **transición**.
- $q_0 \in Q$ es el **estado inicial**.
- $F \subseteq Q$ es el conjunto de **estados finales** (o aceptación).

Autómata finito determinista

Ejemplo

- $Q = \{0, 1, 2\}$
- $\Sigma = \{a, b\}$
- $\delta : Q \times \Sigma \to Q$ se define como:

$$\begin{array}{lcll} \delta(0, a) & = & 1 & \\ \delta(1, a) & = & 2 & \\ \delta(2, a) & = & 2 & \\ \delta(q, b) & = & q & \quad \forall\ q \in \{0, 1, 2\} \end{array}$$



- $q_0 = 0$
- $F = \{2\}$

$(Q, \Sigma, \delta, q_0, F)$ es el **código** de la máquina

¿cómo se ejecuta un autómata sobre una palabra?

Sea:

- Un autómata $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$.
- Un input $w = a_1 a_2 \dots a_n \in \Sigma^*$.

Una **ejecución** (o run) ρ de $\mathcal{A}$ sobre w es una secuencia:

$$\rho : \quad p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} p_2 \xrightarrow{a_3} \dots \xrightarrow{a_n} p_n$$

- $p_0 = q_0$ y
- para todo $i \in \{0, 1, \dots, n-1\}$, $\delta(p_i, a_{i+1}) = p_{i+1}$.

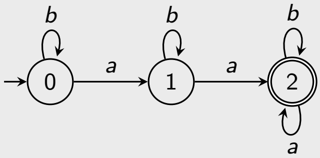
Una ejecución ρ de $\mathcal{A}$ sobre w es de **aceptación** si:

$$p_n \in F.$$

Desde ahora hablaremos de **LA** ejecución de $\mathcal{A}$ sobre w

¿cómo se ejecuta un autómata sobre una palabra?

Ejemplo



- ¿cuál es la ejecución de $\mathcal{A}$ sobre $bbab$?
- ¿cuál es la ejecución de $\mathcal{A}$ sobre $abab$?

¿cuál de las dos ejecuciones son de **aceptación**?

Lenguaje aceptado por un autómata

Sea un autómata $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ y $w \in \Sigma^*$.

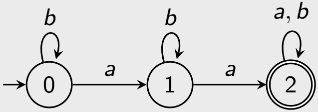
Definiciones

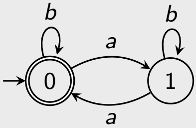
- $\mathcal{A}$ **acepta** w si la ejecución de $\mathcal{A}$ sobre w es de aceptación.
- $\mathcal{A}$ **rechaza** w si la ejecución de $\mathcal{A}$ sobre w NO es de aceptación.
- El **lenguaje aceptado** por $\mathcal{A}$ se define como:

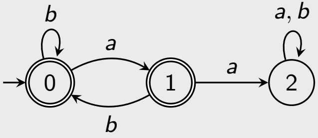
$$\mathcal{L}(\mathcal{A}) = \{w \in \Sigma^* \mid \mathcal{A} \text{ acepta } w\}$$

Lenguaje aceptado por un autómata

¿qué lenguaje acepta/define cada autómata?

1. 

2. 

3. 

Lenguaje aceptado por un autómata

Defina un autómata para los siguientes lenguajes

1. Todas las palabras sobre $\{a, b\}$ tal que
   cada a-letra esta seguida de una b-letra.

2. Todas las palabras sobre $\{a, b\}$ que terminan con ab.

3. Todas las palabras sobre $\{a, b\}$
   con una cantidad par de a-letras tal que no hay dos a-letras seguidas.

Lenguaje aceptado por un autómata

Sea un autómata $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ y $w \in \Sigma^*$.

Definiciones

- $\mathcal{A}$ **acepta** w si la ejecución de $\mathcal{A}$ sobre w es de aceptación.
- $\mathcal{A}$ **rechaza** w si la ejecución de $\mathcal{A}$ sobre w NO es de aceptación.
- El **lenguaje aceptado** por $\mathcal{A}$ se define como:

$$\mathcal{L}(\mathcal{A}) = \{w \in \Sigma^* \mid \mathcal{A} \text{ acepta } w\}$$

- Un lenguaje $L \subseteq \Sigma^*$ se dice **regular** si, y solo si, **existe** un autómata finito determinista $\mathcal{A}$ tal que:

$$L = \mathcal{L}(\mathcal{A})$$

Outline

Autómatas con función parcial de transición

Definición

Un autómata finito determinista con **función parcial de transición** (DFAp):

$$\mathcal{A} \;=\; (Q, \Sigma, \gamma, q_0, F)$$

- Q es un conjunto finito de estados.
- Σ es el alfabeto de input.
- $\gamma : Q \times \Sigma \rightharpoonup Q$ es una **función parcial** de transición.
- $q_0 \in Q$ es el estado inicial.
- $F \subseteq Q$ es el conjunto de estados finales (o aceptación).

Autómatas con función parcial de transición

Sea:

- Un autómata $\mathcal{A} = (Q, \Sigma, \gamma, q_0, F)$ con $\gamma : Q \times \Sigma \rightharpoonup Q$.
- El input $w = a_1 a_2 \dots a_n \in \Sigma^*$.

Una **ejecución** (o run) ρ de $\mathcal{A}$ sobre w es una secuencia:

$$\rho : p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} p_2 \dots \xrightarrow{a_n} p_n$$

- $p_0 = q_0$ y
- para todo $i \in \{0, \dots, n-1\}$ **esta definido** $\gamma(p_i, a_{i+1}) = p_{i+1}$.

Una ejecución ρ de $\mathcal{A}$ sobre w es de **aceptación** si:

$$p_n \in F.$$

Notar que ahora una palabra puede **NO** tener ejecución!

Autómatas con función parcial de transición

Sea un autómata $\mathcal{A} = (Q, \Sigma, \gamma, q_0, F)$ con $\gamma : Q \times \Sigma \rightharpoonup Q$ y $w \in \Sigma^*$.

Definiciones

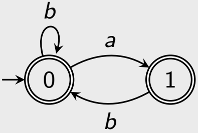
- $\mathcal{A}$ acepta w si
  existe una ejecución de $\mathcal{A}$ sobre w que es de aceptación.

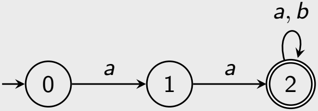
- El **lenguaje aceptado** por $\mathcal{A}$ se define como:

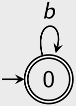
$$\mathcal{L}(\mathcal{A}) \quad = \quad \{w \in \Sigma^* \mid \mathcal{A} \text{ acepta } w\}$$

Autómatas con función parcial de transición

Ejemplos

1. 

2. 

3. 

¿DFA $\not\equiv$ DFAp?

Proposición

Para todo autómata $\mathcal{A}$ con función parcial de transición, existe un autómata $\mathcal{A}'$ (con función total de transición) tal que:

$$\mathcal{L}(\mathcal{A}) \;=\; \mathcal{L}(\mathcal{A}')$$

En otras palabras, DFA $\equiv$ DFAp.

¿cómo demostramos esta afirmación?

¿DFA $\not\equiv$ DFAp?

Demostración

Sea $\mathcal{A} = (Q, \Sigma, \gamma, q_0, F)$ un autómata con función parcial de transición.

Sea q_s un **nuevo estado** tal que $q_s \notin Q$.

Construimos el DFA $\mathcal{A}' = (Q \cup \{q_s\}, \Sigma, \delta', q_0, F)$ tal que:

$$\delta'(p, a) = \begin{cases} \gamma(p, a) & \text{si } p \neq q_s \text{ y } (p, a) \in \mathrm{dom}(\gamma) \\ q_s & \text{si no} \end{cases}$$

para todo $p \in Q \cup \{q_s\}$ y $a \in \Sigma$.

¿cómo demostramos que $\mathcal{A}$ y $\mathcal{A}'$ definen el **mismo lenguaje**?

¿DFA $\not\equiv$ DFAp?

Demostración: $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{A}')$

Sea $w = a_1 \dots a_n \in \mathcal{L}(\mathcal{A})$.

Entonces existe una **ejecución de aceptación** ρ de $\mathcal{A}$ sobre w:

$$\rho : p_0 \stackrel{a_1}{\to} p_1 \stackrel{a_2}{\to} p_2 \dots \stackrel{a_n}{\to} p_n$$

- $p_0 = q_0$,
- para todo $i \in \{0, \dots, n-1\}$, esta definido $\gamma(p_i, a_{i+1}) = p_{i+1}$ y
- $p_n \in F$.

Como $\delta(p_i, a_{i+1}) = \gamma(p_i, a_{i+1})$ para todo $i \in \{0, \dots, n-1\}$ (¿por qué?) entonces ρ es también una ejecución de aceptación de $\mathcal{A}'$ sobre w.

Por lo tanto, $w \in \mathcal{L}(\mathcal{A}')$.

¿DFA $\not\equiv$ DFAp?

Demostración: $\mathcal{L}(\mathcal{A}') \subseteq \mathcal{L}(\mathcal{A})$

Sea $w = a_1 \dots a_n \in \mathcal{L}(\mathcal{A}')$.

Existe una **ejecución de aceptación** ρ de $\mathcal{A}'$ sobre w:

$$\rho : p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} p_2 \dots \xrightarrow{a_n} p_n$$

- $p_0 = q_0$,
- para todo $i \in \{0, 1, \dots, n-1\}$, $\delta(p_i, a_{i+1}) = p_{i+1}$ y
- $p_n \in F$.

¿cómo demostramos que ρ **también** es una ejecución de $\mathcal{A}$ sobre w?

¿DFA $\not\equiv$ DFAp?

Demostración: $\mathcal{L}(\mathcal{A}') \subseteq \mathcal{L}(\mathcal{A})$

Demostraremos que $p_i \neq q_s$ para todo $i \in \{0, \ldots, n\}$.

Por **contradición**, suponga que existe i tal que $p_i = q_s$.

Entonces, tenemos que $p_{i+1} = q_s$. (¿por qué?)

Por **inducción**, podemos demostrar que $p_j = q_s$ para todo $j \geq i$. (¿cómo?)

Por lo tanto, $p_n = q_s$. (**contradicción**!) (¿por qué?)

Como $p_i \neq q_s$ para todo $i \in \{0, \ldots, n\}$ tenemos que:

$$\delta(p_i, a_{i+1}) = \gamma(p_i, a_{i+1}) \quad \forall\ i \in \{0, 1, \ldots, n-1\}$$

y ρ es una **ejecución de aceptación** de $\mathcal{A}$ sobre w.

Por lo tanto, concluimos que $w \in \mathcal{L}(A)$. $\square$

¿DFA $\not\equiv$ DFAp?

Proposición

Para todo autómata $\mathcal{A}$ con función parcial de transición, existe un autómata $\mathcal{A}'$ (con función total de transición) tal que:

$$\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$$

En otras palabras, DFA $\equiv$ DFAp.

Advertencia:

Desde ahora, utilizaremos autómatas con **funciones totales de transición**, pero **sin perdida de generalidad** en algunos ejemplos utilizaremos **funciones parciales de transición** por simplicidad.