PRÁCTICA 0

Licenciatura en Ciencias Computacionales

ALUMNO:

Sánchez Lara José de Jesús

SEMESTRE/GRUPO

6 ° 3

Gramáticas y Lenguajes Formales

Alfabeto

Un alfabeto es un conjunto de símbolos finito no vacío que se representa convencionalmente con el símbolo $\sum$. [2]
Ejemplo 1:

- $\sum = \{0, 1\}$. Alfabeto binario.

Ejemplo 2:

- $\sum = \{a, b, ..., z\}$. Conjunto de todas las letras minúsculas.

1.Concatenación

Operación que partiendo de tener dos palabras, da como resultado la unión de esas palabras del alfabeto.
Formalmente:

$$x_1 = a_1 a_2 a_3 \dots a_m$$
$$x_2 = b_1 b_2 b_3 \dots b_n$$

dadas dos palabras, se puede ver como la concatenación de "m" símbolos de la palabra "x1" con "n" símbolos de la palabra "x2" es una cadena de resultante de la forma:

$$x_1 x_2 = a_1 a_2 \dots a_m b_1 \dots b_n$$

El vacío no aporta nada a esta operación. Concatenar con λ es igual a la longitud de la palabra.
Ejemplo 1:

- $Cadena1 = Tele, Cadena2 = visión.$
- $Cadena\,resultante = Televisión$

Ejemplo 2:

- $Cadena1 = Poder, Cadena2 = osos.$
- $Cadena\,resultante = Poderosos$

Potencia

La potencia "n-ésima" de una palabra es el resultado de concatenar esta palabra consigno mismo "n" veces y cualquier palabra elevada a 0 es cadena vacía.

Ejemplo 1:

- $Palabra : Tele^2 = TeleTele.$

Ejemplo 2:

- $Palabra : (co)^2 = coco.$

Prefijos, sufijos y segmentos:

Dadas x e y, palabras de un alfabeto.
y es un segmento de x si existe u y v tales que $x = u * y * v$.
Si $u = \lambda$entonces y es un prefijo.
Si $v = \lambda$ entonces y es un sufijo. ***Ejemplo 1:***

- Dada la palabra $hola$, los prefijos son: $\{\lambda, h, ho, hol, hola\}$

Ejemplo 2:

- Dada la palabra $hola$ los sufijos son: $\{\lambda, a, al, alo, aloh\}$

Reverso

Es el proceso de leer una palabra de derecha a izquierda. Formalmente se define como:

$$(xa)^r = ax^r$$

Notas:
El reverso de λ es λ
El reverso de un símbolo es ese símbolo.
Ejemplo 1:

- $Cadena = AMLO, reverso = OLMA.$

Ejemplo 2:

- $Cadena = Amor, reverso = roma.$

Palabras capicúa

Palabras cuyo reverso es igual a su estado normal.
Ejemplo 1:

- $Cadena = oso, reverso = oso.$

Ejemplo 2:

- $Cadena1 = oro, reverso = oro.$

Lenguaje

De manera formal se puede definir como un subconjunto de la clausula de Kleen sobre un alfabeto.
Ejemplo 1:

- Dado el alfabeto del español $\sum_{espanol}$, los posibles lenguajes son:
- $L_1 = \{bx|x \in \sum_{espanol}^*\}$
- L1 es el lenguaje de todas las palabras que empiezan por b.

Ejemplo 2:

- Dado el alfabeto binario $\sum_{binariol}$, los posibles lenguajes son:
- $L_1 = \{1x|x \in \sum_{binario}^*\}$
- L1 es el lenguaje de todas las palabras que empiezan por 1.

Operaciones booleanas

Unión de lenguajes

Dado un lenguaje $L1$ y un lenguaje $L2$, la unión da como resultado un lenguaje con todas las palabras tanto de $L1$ como de $L2$.
Ejemplo 1:

- Dado el alfabeto del español $\sum_{espanol}$, los posibles lenguajes son:
- $L_1 = \{bx \mid x \in \sum_{espanol}^*\}$
- $L_2 = \{ax \mid x \in \sum_{espanol}^*\}$
- L_1 es el lenguaje de todas las palabras que empiezan por b.
- L_2 es el lenguaje de todas las palabras que empiezan por a.
- La unión $L_1 \cup L_2$ es el lenguaje de todas las palabras que empiezan por a o por b.

Ejemplo 2:

- Dado el alfabeto binario $\sum_{binario}$, los posibles lenguajes son:
- $L_1 = \{1x \mid x \in \sum_{binario}^*\}$
- $L_2 = \{0x \mid x \in \sum_{binario}^*\}$
- L_1 es el lenguaje de todas las palabras que empiezan por 1.
- L_2 es el lenguaje de todas las palabras que empiezan por 0.
- La unión $L_1 \cup L_2$ es el lenguaje de todas las palabras en el alfabeto binario.

Producto

El producto de dos lenguajes es la concatenación de cada una de las palabras del primer lenguaje con cada una de los del segundo lenguaje.
Ejemplo 1:

- Dado el alfabeto del inglés $\sum_{ingles}$, los posibles lenguajes son:
- $L_1 = \{hx \mid x \in \sum_{ingles}^*\}$
- $L_2 = \{wx \mid x \in \sum_{ingles}^*\}$
- El producto L_1L_2 es el lenguaje de todas las palabras que consisten en una palabra de L_1 concatenada con una palabra de L_2.
- Ejemplo de elementos en L_1L_2: {hawa, hewa, hewi y así sucesivamente}.

Ejemplo 2:

- Dado el alfabeto octal $\sum_{octal}$, los posibles lenguajes son:
- $L_1 = \{3x \mid x \in \sum_{octal}^*\}$
- $L_2 = \{7x \mid x \in \sum_{octal}^*\}$
- El producto L_1L_2 es el lenguaje de todas las palabras que consisten en una palabra de L_1 concatenada con una palabra de L_2.
- Ejemplo de elementos en L_1L_2: {37, 374, 3127, 3567, 3007}.

Propiedad importante del producto: λ solo puede presentarse en el producto si esta en ambos lenguajes.

Potencia sobre lenguaje

sea L un lenguaje elevado a n, si $n = 0$, entonces el resultado es vacío, en caso contrario, hacemos $L^{n-1} * L$ de forma recursiva hasta llegar al primer caso.
Ejemplo 1:

- Dado el lenguaje $L = \{a, ab\}$, su potencia L^3 es:
- $L^3 = \{aaa, aaab, aba, abab, abaa, ababb\}$
- Cada palabra en L^3 se obtiene concatenando tres palabras de L.

Ejemplo 2:

- Dado el lenguaje $L = \{01, 10\}$, su potencia L^2 es:
- $L^2 = \{0101, 0110, 1001, 1010\}$
- Cada palabra en L^2 se obtiene concatenando dos palabras de L.

Cierre

Unión de todas las potencias de un lenguaje (se incluye la cadena vacía).
Ejemplo 1:

- Dado el lenguaje $L = \{a, ab\}$, la unión de todas sus potencias es:
- $L^* = \{\lambda, a, ab, aa, aab, aba, abab, aaa, \ldots\}$

Ejemplo 2:

- Dado el lenguaje $L = \{01, 10\}$, la unión de todas sus potencias es:
- $L^* = \{\lambda, 01, 10, 0101, 0110, 1001, 1010, 010101, \ldots\}$

Cociente por la izquierda/derecha

Todas aquellas palabras obtenidas tras eliminar el símbolo al que se le aplica la operación cociente dentro del lenguaje.
Formalmente: $u^{-1}L = \{v \in \sum^* : uv \in L\}$
Ejemplo 1:

- Dado el lenguaje $L = \{aa, ab\}$, el cociente de L por a es: $= \{a, b\}$

Ejemplo 2:

- Dado el lenguaje $L = \{hola\}$, el cociente de L por h es: $= \{ola\}$

Homomorfismo

Aplicar un alfabeto hacia otro. Se puede ver como una asignación de los símbolos del alfabeto a otro alfabeto (matemáticamente visto como una función o implicación).
Ejemplo 1:

- Dado el alfabeto $\sum_1 = \{a, b\}$ y el alfabeto $\sum_2 = \{0, 1\}$ la función de homomorfismo $h : \sum_1 \rightarrow \sum_2$ decimos que $h(a) = 0$ y $h(b) = 1$

Ejemplo 2:

- Dado el alfabeto $\sum_1 = \{a, b\}$ y el alfabeto $\sum_2^* = \{160, 98\}$ la función de homomorfismo $h : \sum_1 \rightarrow \sum_2$ decimos que $h(a) = 160$ y $h(b) = 98$ (ambos números representan el código ascii.

Autómatas

Máquina abstracta para modelar procesos de cálculo compuesta por estados y transiciones que indican como debe funcionar.
notas:

- Se usan grafos para representarlas.
- Los nodos son los estados.
- Los arcos son las transiciones.
- Ayudan a comprender y resolver problemas.

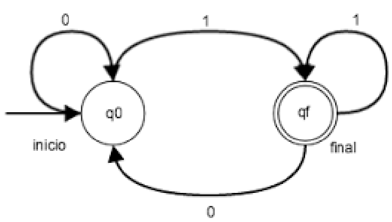

Figure 1: Autómata 1

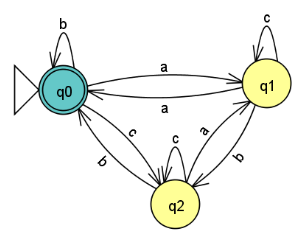

Figure 2: Autómata 2

Máquinas de turing

Máquinas que resuelven problemas que pueden ser resueltos por una computadora.

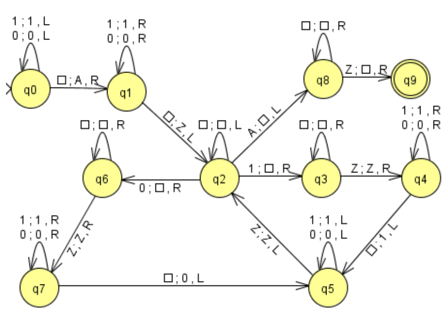

Figure 3: Máquina 1

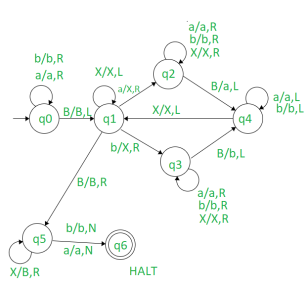

Figure 4: Máquina 2

Autómata Finito Determinista (AFD)

Máquina abstracta formada por:

- Q: estados del autómata.
- $\sum$: alfabeto del autómata.
- δ: función de transición que recoge todas las
- $q0$: estado inicial.
- F: subconjunto de Q. Recoge todos los estados finales.

Ejemplo 1:

- Dado el autómata finito determinista (AFD) su 5-tupla es:
  - Conjunto de estados: $Q = \{q0, q1, q2\}$
  - Alfabeto: $\Sigma = \{a, b\}$
  - Función de transición: $\delta = \{(q0, a, q1), (q0, b, q0), (q1, a, q1), (q1, b, q2), (q2, a, q1), (q2, b, q0)\}$
  - Estado inicial: $q_0 = q0$
  - Conjunto de estados de aceptación: $F = \{q2\}$

Ejemplo 2:

- Dado el autómata finito determinista (AFD) su 5-tupla es:
  - Conjunto de estados: $Q = \{q0, q1\}$
  - Alfabeto: $\Sigma = \{a, b\}$
  - Función de transición: $\delta = \{(q0, a, q1), (q0, b, q0), (q1, a, q1), (q1, b, q1)\}$
  - Estado inicial: $q_0 = q0$
  - Conjunto de estados de aceptación: $F = \{q1\}$

Es importante entender que se debe llevar a cabo un análisis no sólo de los caracteres que aparecen en cierta cadena, sino que también sus posiciones relativas. [1]

Autómata completo

En cada estado hay una transición para cada símbolo del alfabeto.
Ejemplo 1:

- Dado el autómata finito determinista (AFD) que acepta su 5-tupla es:
  - Conjunto de estados: $Q = \{q0, q1, q2\}$
  - Alfabeto: $\Sigma = \{a, b\}$
  - Función de transición: $\delta = \{(q0, a, q1), (q0, b, q2), (q1, a, q2), (q1, b, q2), (q2, a, q2), (q2, b, q2)\}$
  - Estado inicial: $q_0 = q0$
  - Conjunto de estados de aceptación: $F = \{q1\}$

Ejemplo 2:

- Dado el autómata finito determinista (AFD) que acepta su 5-tupla es:
  - Conjunto de estados: $Q = \{q0, q1, q2, q3\}$
  - Alfabeto: $\Sigma = \{a, b\}$
  - Función de transición: $\delta = \{(q0, a, q1), (q0, b, q2), (q1, a, q1), (q1, b, q3), (q2, a, q3), (q2, b, q2), (q3, a, q1), (q3, b, q2)\}$
  - Estado inicial: $q_0 = q0$
  - Conjunto de estados de aceptación: $F = \{q3\}$

Tabla de transiciones

Forma de representar un autómata finito determinista.
Ejemplos:

Table 1: Tabla de transición del AFD.

Estado	**a**	**b**
q0	q1	q2
q1	q2	q2
q2	q2	q2

Table 2: Tabla de transición del AFD".

Estado	**0**	**1**
q0	q1	q2
q1	q1	q3
q2	q3	q2
q3	q1	q2

Autómatas Finito no determinista (AFND)

Los autómatas finitos no deterministas son autómatas que aceptan varias transiciones desde un estado para un símbolo.
Todo AFD es un caso particular de un AFND. Formalmente: Tupla compuesta de Q, $\sum$, δ, $q0$ y F.
Ejemplos

Table 3: Tabla de transición del AFND

Estado	**a**	**b**
q0	q0, q1	q0, q1
q1	q0	q1

Table 4: Tabla de transición del AFND.

Estado	**a**	**b**
q0	q1, q0	q0
q1	q1	q1

Autómata finito determinista equivalente

Dado un AFND, el AFD equivalente se representa de la siguiente forma:

$$A' = (Q', \sum, \delta', q_{0'}, F') \tag{1}$$

$$Q' = 2^Q \tag{2}$$

$$q_0' = \{q_0\} \tag{3}$$

$$F' = \{q' \in Q' | q' \cap F \neq \emptyset\} \tag{4}$$

Autómatas con transiciones vacías

Todo AFND es un AFλ, por lo tanto, un AFλ es un autómata que permite como símbolo las cadenas vacías, por lo que su función de transición cambia y permite cambiar entre estados sin haber procesado símbolos de entrada. También se le conoce como Autómata finito no determinista con movimientos y es fácilmente diferenciable con el AFND ya que en la función de transición se incluye el símbolo λ/ϵ.[3]

Lambda / λ

- Palabra que representa la ausencia de símbolos.
- Tamaño 0.
- Se pueden referir a ella como epsilon/ϵ.
- El conjunto vacío no es un lenguaje.
- Se puede aplicar a todas las operaciones.

Ejemplo 1

- Dado el lenguaje $L = \{a, \lambda\}$, un autómata podría aceptar la cadena vacía λ.

Ejemplo 2

- Dado el lenguaje $L = \{ab, \lambda\}$, el autómata aceptará la cadena vacía λ.

$\lambda - clausura$

La $\lambda-clausura$ de un estado q es este mismo estado unión con todos aquellos estados accesibles desde q a través de transiciones con λ.
Ejemplo 1:

Table 5: Tabla de Transiciones

Estado	Entrada a	Entrada b	Entrada λ
q_0		q_2	q_1
q_1			
q_2	q_1	q_2	q0-

Ejemplo 2:

Table 6: Tabla de Transiciones

Estado	Entrada a	Entrada b	Entrada λ
q_0	q_0		q_1
q_1	q_2	q_2	q_0
q_2	q_2	-	-

Patter matching

Es una técnica que busca ocurrencias de un patrón específico en un conjunto de datos más grande.
Ejemplo 1

- Dado el patrón "ab" y el texto "abcabcab", el objetivo es encontrar las ocurrencias del patrón en el texto.
- En el texto "abcabcab", el patrón "ab" aparece en las siguientes posiciones: 0, 3, 6.

Ejemplo 2

- Dado el patrón "123" y el texto "123456123", el objetivo es encontrar las ocurrencias del patrón en el texto.
- En el texto "123456123", el patrón "123" aparece en las siguientes posiciones: 0, 6.

Naive algorithm

"Algoritmo Ingenuo" es un algoritmo que puede buscar patrones en todas las posiciones en que aparece dentro de un texto, cosa que no puede hacer un AFD pero no es eficiente, de ahí su nombre.
Ejemplo 1

- Dado el patrón "abc" y el texto "abaabc", el algoritmo ingenuo comienza a comparar el patrón con cada subcadena del texto.
- Primer intento de comparación: "abc" con "aba", no coincide.
- Segundo intento de comparación: "abc" con "abc", coincide.
- El patrón "abc" se encuentra en la posición 3 del texto "ababcabc".

Ejemplo 2

- Dado el patrón "ab" y el texto "abab", el algoritmo ingenuo comienza a comparar el patrón con cada subcadena del texto.
- Primer intento de comparación: "ab" con "ab", coincide.
- El patrón "ab" se encuentra en la posición 0 del texto "abab".

Lenguaje por la derecha

Son todas aquellas palabras que nos permiten a partir de un estado, llegar a un estado final y sirve para definir el comportamiento de los estados del autómata. ***Ejemplo 1***

- Lenguaje dado: $L = \{b, a\}$. **Lenguaje por la derecha:** $L_R = \{a, b, aa, bb, aaa\}$ (para un autómata que acepta cualquier palabra que tiene al menos una a o una b.

Ejemplo 2

- Lenguaje dado: $L = \{b, a\}$. **Lenguaje por la derecha:** $L_R = \{bab, ba\}$ (Para un autómata que termina cuando detecta que la cadena tiene ba.

Lenguaje regular

Aquel que contiene un número finito de clases de equivalencia.
Ejemplo 1

- Consideramos el lenguaje $L = (a|b)^*$, que es el lenguaje de todas las cadenas formadas por una secuencia de a y b, incluida la cadena vacía.
- Este lenguaje es regular, ya que puede ser expresado por una expresión regular y aceptado por un autómata finito determinista.

Ejemplo 2

- Consideramos el lenguaje $L = \{a^n b^n \mid n \geq 0\}$, que consiste en las cadenas formadas por n a's seguidos de n b's.
- Este lenguaje también es regular, ya que puede ser descrito por una expresión regular adecuada y aceptado por un autómata finito determinista.

Evidencia fotográfica

Figure 5: Evidencia 1

Figure 6: Evidencia 2

Figure 7: Evidencia 3

Referencias Bibliográficas

References

[1] Kelley, D. (**1995**). *Teoría de autómatas y lenguajes formales*. Prentice-Hall, Madrid. ISBN 0134977777. Clasificación LC QA76 .9 .M35 K4. Traducción de Ma. Luisa Diez Platas.

[2] Hopcroft, J. E. (**2002**). *Introducción a la teoría de autómatas, lenguajes y computación*. 2a ed. Pearson Education, México.

[3] Cases Muñoz, R. (**2002**). *Lenguajes, gramáticas y autómatas: curso básico*. Pearson Education, México.

[4] Codemath. (**2024**). *Autómatas y lenguajes formales desde cero* [Lista de reproducción]. YouTube. Recuperado de https://www.youtube.com/

[5] Codemath. (**2024**). *Convertir un Automata NO Determinista (AFND) a Determinista (AFD)* [Video]. YouTube. Recuperado de https://www.youtube.com/watch?v=_UdVL-84rXc

[6] Codemath. (**2024**). *Qué es un Autómata Finito No Determinista (AFND)* [Video]. YouTube. Recuperado de https://www.youtube.com/watch?v=

[7] Codemath. (**2023**). *Qué es un Autómata Finito Determinista (AFD)* [Video]. YouTube. Recuperado de https://www.youtube.com/watch?v=

[8] Codemath. (**2023**). *Operaciones con Lenguajes y Aplicaciones — Lenguajes Formales III* [Video]. YouTube. Recuperado de https://www.youtube.com/watch?v=