Tema 1

Nociones Preliminares y Lenguajes

Dr. Luis Pineda, IIMAS, UNAM, 2010

Conceptos centrales

- Alfabetos
- Cadenas
- Lenguajes
- Representación
- Interpretación
- Problemas
- Funciones, algoritmos y fórmulas

Dr. Luis Pineda, IIMAS, UNAM, 2010

Alfabetos

- Conjunto finito (no vacío) de símbolos: Σ
  - $\Sigma = \{a, b, c, \ldots, z\}$
  - $\Sigma = \{\alpha, \beta, \gamma, \ldots, \omega\}$
  - $\Sigma = \{0, 1\}$
  - $\Sigma = \{0, 1, 2, 3, \ldots, 9\}$
  - $\Sigma = \{1\}$
  - Σ = El conjunto de caracteres ASCII

Dr. Luis Pineda, IIMAS, UNAM, 2010

Cadenas

- Una cadena (o palabra) sobre un alfabeto Σ
  - Una secuencia finita de símbolos de Σ
- Longitud de una cadena
  - El número (posiciones) de símbolos en la cadena:
    - w = "111" tiene un símbol pero tres posiciones
  - La longitud de una cadena w es $|w|$
    - $|w| = 3$
- La cadena *nula*: Λ (lambda)
  - Λ se puede seleccionar en cualquier alfabeto
    - $|\Lambda| = 0$

Dr. Luis Pineda, IIMAS, UNAM, 2010

Cadenas

- Notación:
  - Minúsculas al principio del alfabeto denotan símbolos: a, b, c…
  - Mínusculas al final del alfabeto denotan cadenas: w, x, y, z...

Dr. Luis Pineda, IIMAS, UNAM, 2010

Cadenas

- Potencias de un alfabeto
  - Si Σ es un alfabeto Σ^k es el conjunto de cadenas de longitud k, tales que todos los símbolos están en Σ
  - $\Sigma^0 = \{\Lambda\}$
  - Si $\Sigma = \{0, 1\}$ entonces
    - $\Sigma^0 = \{\Lambda\}$
    - $\Sigma^1 = \{0, 1\}$
    - $\Sigma^2 = \{00, 01, 10, 11\}$, etc.
  - $\Sigma \neq \Sigma^1$ (Σ es el alfabeto and Σ^1 es el conjunto de cadenas de longitud 1)

Dr. Luis Pineda, IIMAS, UNAM, 2010

Cadenas

- Para cualquier alfabeto Σ el conjunto de todas las cadenas sobre Σ se denota Σ^*
  - $\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \ldots$
    - $\Sigma^* = \{0, 1\}^* = \{\Lambda, 0, 1, 00, 01, 10, 11, \ldots\}$
- Y sin Σ^0:
  - $\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \ldots$
    - $\Sigma^+ = \{0, 1\}^+ = \{0, 1, 00, 01, 10, 11, \ldots\}$

Σ^* es denumerable

- Un conjunto es *denumerable*, *enumerable infinito* o *contable* si puede ser ordenado en una sóla lista (probablemente infinita):
  - 1, 2, 3, 4... es una lista infinita
  - 1, 3, 5, …, 2, 4, 6… no lo es! ($\infty + 1$?)
  - Es necesario poder decir qué número le corresponde a cada elemento de la lista: quién le precede y quién le sigue
  - Un conjunto *A* es denumerable si existe una función con dominio en los número naturales *N* tal que a cada miembro de *A* se asocia un número *n* en *N*

Σ^* es denumerable

N	Σ^*
1	Λ
2	0
3	1
4	00
5	01
6	10
7	11
8	000

N	Σ^*
9	001
10	010
11	011
12	100
13	101
14	110
15	111
16	...

$\Sigma^* = \{0, 1\}^* = \{\Lambda, 0, 1, 00, 01, 10, 11, \ldots\}$

Σ^* es denumerable

- $\Sigma = \{a,..,z\}$ (26 letras)
  - Σ^0 tiene 1 cadena de longitud 0 (i.e. Λ)
  - Σ^1 tiene 26 cadenas de longitud 1 (i.e. a,…,z)
  - Σ^2 tiene 26^2 cadenas de longitud 2 (en orden alfabético)
  - Σ^3 tiene 26^3 cadenas de longitud 3 (en orden alfabético)
  - …
- Existe una función tal que cada argumento *n* tiene como valor la cadena correspondiente en Σ^*

Cadenas

- Concatenación de cadenas:
  - Si *x* y *y* son cadenas *xy* denota la concatenación de *x* con *y*
  - Más específicamente: Si $x = x_1 x_2 \ldots x_i$ & $y = y_1 y_2 \ldots y_j$ entonces $xy = x_1 x_2 \ldots x_i y_1 y_2 \ldots y_j$
  - e.g: $x = 01101$ y $y = 110$ entonces $xy = 01101110$
  - $|xy| = i + j$
  - Identidad para la concatenación: $\Lambda x = x\Lambda = x$
  - La concatenación es asociativa: $(xy)z = x(yz)$

Cadenas

- Concatenación de conjuntos de cadenas
  - Si $A, B \subseteq \Sigma^*$ la concatenación de *A* y *B* es

    $$AB = \{xy \mid x \in A \;\&\; y \in B\}$$
- La concatenación no es conmutativa
  - $A = \{a, b\}$ y $B = \{c, d\}$
  - $AB = \{ac, ad, bc, bd\}$
  - $BA = \{ca, cb, da, db\}$

Concatenación de cadenas

$AB = \{xy \mid x \in A \ \& \ y \in B\}$

d	ad	bd
c	ac	bc
B / *A*	a	b

Diferente de BA = {ca, cb, da, db}

Cadenas

- Sin embargo:
  - $\Sigma^{n+1} = \Sigma\,\Sigma^n = \Sigma^n\,\Sigma \quad (\Sigma = \Sigma^1)$
  - If $\Sigma = \{0, 1\}$ then
    - $\Sigma^0 = \{\Lambda\}$
    - $\Sigma^1 = \{\Lambda\}\{0, 1\} = \{0, 1\}\{\Lambda\} = \{0, 1\}$
    - $\Sigma^2 = \{0, 1\}\{0, 1\} = \{00, 01, 10, 11\}$
    - $\Sigma^3 = \{0, 1\}\{00, 01, 10, 11\} = \{00, 01, 10, 11\}\ \{0, 1\}$
      $= \{000, 001, 010, 011, 100, 101, 110, 111\}$
    - …

Cerradura de un conjunto de cadenas

- $\{x, y\}^*$ = Cadenas compuestas con x & y de todas las formas posibles!

y	*xy*	*yy*
x	*xx*	*yx*
	x	*y*

Lenguajes

- Un *lenguaje* es un conjunto de cadenas compuestas con símbolos de un alfabeto
- Lenguajes naturales (Español, Inglés, etc.)
  - Nivel sintáctico: oraciones compuestas de palabras
  - Nivel léxico: palabras compuestas de símbolos del alfabeto
- Lenguajes formales
  - Nivel sintáctico: expresiones bien formadas compuestas de cadenas (tokens)
  - Nivel léxico: tokens compuestos por símbolos del alfabeto (e.g. ASCII)

Definiciones de Lenguajes

- Un lenguaje sobre Σ es un subconjunto de Σ^*
- L es un lenguaje sobre Σ si $L \subseteq \Sigma^*$
- L no necesita incluir todos los símbolos de Σ, por lo mismo, si L es un lenguaje sobre Σ, también lo es sobre un super conjunto de Σ

Ejemplos de Lenguajes

- El lenguaje de todas las cadenas formadas por n 0's seguidas de n 1's, para algún $n \geq 0$:

  $\{\Lambda, 01, 0011, 000111, \ldots\}$
- El conjunto de cadenas de 0's y 1's con el mismo número de ambos símbolos:

  $\{\Lambda, 01, 10, 0011, 0101, 1001, \ldots\}$
- El conjunto de cadenas representando los números primos en notation binaria:

  $\{10, 11, 101, 111, 1011, \ldots\}$

  pero ¿Es ésta una definición correcta?

Ejemplos de Lenguajes

- Σ^* es un lenguaje sobre cualquier alfabeto
- Φ, el lenguaje vacío, es un lenguaje sobre cualquier alfabeto
- $\{\Lambda\}$, el lenguaje consistente de la cadena vacía
- En particular, notar que $\Phi \neq \{\Lambda\}$

Especifiación de lenguajes

- Expresado con notación informal de conjuntos:
  - $\{w \mid$ una propiedad de $w\}$
  - Ejemplo: $\{w \mid w$ consiste de una secuencia de n 0's seguida de una secuencia de n 1's$\}$
- Expresando w con parámetros
  - $\{0^n1^n \mid n \geq 0\}$ donde n es el parámetro
  - $\{0^i1^j \mid 0 \leq i \leq j\}$ donde i y j son los parámetros
- Combinando las operaciones de conjuntos con la concatencación
  - $\{ab, bab\}^* \cup \{b\}\{bb\}^*$
- E incluso:
  - $\{byb \mid y \in \{a, b\}^*\}$
- Es conveniente tener una manera simple y directa para definir lenguajes!

Representación

- Las cadenas de un lenguaje *representan* objetos del mundo:
  - Juan, Pedro y Luis se representan por $\{juan, pedro, luis\}$
  - 2, 3, 5, 7, 11 se representan por {10, 11, 101, 111, 1011, …}
- Una representación puede pensarse como una función del mundo al lenguaje
- Es la función que evalúa quien envía un mensaje!

Juan → se representa por → *juan*

Interpretación

- Las cadenas del lenguaje *se interpretan* como objetos en el mundo:
  - $\{juan, pedro, luis\}$ se interpretan como Juan, Pedro y Luis
  - {10, 11, 101, 111, 1011, …} se interpretan como 2, 3, 5, 7, 11
- Una interpretación se puede pensar como una función del lenguaje al mundo!
- Es la función que evalúa quien recibe el mensaje!

juan → se interpreta como → Juan

Problemas

- Un *problema* consiste en decidir si una cadena pertenece a un lenguaje
- Si Σ es un alfabeto, L es un lenguaje sobre Σ, el problema L es:
  - Dada una cadena w en Σ^*, decidir si o no w está en L
- Problemas y lenguajes son realmente lo mismo

Un Problema:

- El problema de decidir si un número es primo puede expresarse con el lenguaje L_p consistente de todas las cadenas monádicas cuya longitud sea un número primo:
  - $L_p = \{11, 111, 11111, 1111111, 11111111111, \ldots\}$
  - $111 \in L_p$
  - $1111 \notin L_p$
- Dada una cadena de 1's hay que decir "si" o "no" dependiendo de si dicha cadena representa o no a un número primo

Problemas

- Dos maneras de pensar acerca de los problemas:
  - Como un problema de decisión: Decidir si una cadena está incluida en el conjunto (i.e. el lenguaje)

    111 ∈ {11, 111, 11111, 1111111, 11111111111,…} ?
  - Como un proceso que transforma una entrada en cierta salida:

    111 → Algoritmo → si
  - ¿Qué es más fácil?

Algoritmos

- Función: Objeto matemático
- Algoritmo: Procedimiento de cálculo
- Fórmula: Expresión compacta (analítica) del procedimiento!

El problema de los racionales

¿Cómo podemos decidir si un número es racional?

El problema de los racionales

1/1	2/1	3/1	…	$m/1$
1/2	2/2	3/2	…	$m/2$
1/3	2/3	3/3	…	$m/3$
…	…	…	…	…
$1/m$	$2/m$	$3/m$	…	m/m

Los racionales son numerables!

La función que los ordena...

1	2	5	10	…
4	3	6	11	…
9	8	7	12	…
16	15	14	13	…
…	…	…	…	…

$\xrightarrow{r}$

1/1	2/1	3/1	…	$m/1$
1/2	2/2	3/2	…	$m/2$
1/3	2/3	3/3	…	$m/3$
…	…	…	…	…
$1/m$	$2/m$	$3/m$	…	m/m

Los racionales son numerables!

Forma de *r*

1	2	5	10	…
4	3	6	11	…
9	8	7	12	…
16	15	14	13	…
…	…	…	…	…

$\xrightarrow{r}$

1/1	2/1	3/1	…	$m/1$
1/2	2/2	3/2	…	$m/2$
1/3	2/3	3/3	…	$m/3$
…	…	…	…	…
$1/m$	$2/m$	$3/m$	…	m/m

Los enteros: $r(1) = 1$, $r(2) = 2$, $r(5) = 3$, ...

Forma de r

1	2	5	10	…
4	3	6	11	…
9	8	7	12	…
16	15	14	13	…
…	…	…	…	…

$\xrightarrow{r}$

1/1	2/1	3/1	…	m/1
1/2	2/2	3/2	…	m/2
1/3	2/3	3/3	…	m/3
…	…	…	…	…
1/m	2/m	3/m	…	m/m

Redundante: $r(1) = r(3) = r(7) = r(13) = ... = 1$

Algoritmo para $r(n)$

1	2	5	10	…
4	3	6	11	…
9	8	7	12	…
16	15	14	13	…
…	…	…	…	…

$\xrightarrow{r}$

1/1	2/1	3/1	…	m/1
1/2	2/2	3/2	…	m/2
1/3	2/3	3/3	…	m/3
…	…	…	…	…
1/m	2/m	3/m	…	m/m

Cualquier n está sobre un cuadrado de lado m:

$sq(6) = 3$

$sq(8) = 3$

Algoritmo para $r(n)$

1	2	5	10	…
4	3	6	11	…
9	8	7	12	…
16	15	14	13	…
…	…	…	…	…

$\xrightarrow{r}$

1/1	2/1	3/1	…	m/1
1/2	2/2	3/2	…	m/2
1/3	2/3	3/3	…	m/3
…	…	…	…	…
1/m	2/m	3/m	…	m/m

$sq(n)$ es el numerador o el denominador de un racional en la imagen de r: $sq(n) = m$

Algoritmo para $r(n)$

1	2	5	10	…
4	3	6	11	…
9	8	7	12	…
16	15	14	13	…
…	…	…	…	…

$\xrightarrow{r}$

1/1	2/1	3/1	…	m/1
1/2	2/2	3/2	…	m/2
1/3	2/3	3/3	…	m/3
…	…	…	…	…
1/m	2/m	3/m	…	m/m

Si n está en columna derecha:

$r(n) = sq(n)/? = m/?$

Ejemplo: $r(6) = sq(6)/? = 3/?$

Algoritmo para $r(n)$

1	2	5	10	…
4	3	6	11	…
9	8	7	12	…
16	15	14	13	…
…	…	…	…	…

$\xrightarrow{r}$

1/1	2/1	3/1	…	m/1
1/2	2/2	3/2	…	m/2
1/3	2/3	3/3	…	m/3
…	…	…	…	…
1/m	2/m	3/m	…	m/m

Si n está en renglón inferior:

$r(n) = ?/sq(n) = ?/m$

Ejemplo: $r(8) = ?/sq(8) = ?/3$

Algoritmo para $r(n)$

1	2	5	10	…
4	3	6	11	…
9	8	7	12	…
16	15	14	13	…
…	…	…	…	…

$\xrightarrow{r}$

1/1	2/1	3/1	…	m/1
1/2	2/2	3/2	…	m/2
1/3	2/3	3/3	…	m/3
…	…	…	…	…
1/m	2/m	3/m	…	m/m

Cálculo esquina inferior derecha:

Si n está en la columna derecha n es cuando más: $n \leq (sq(n) - 1)^2 + sq(n) = (m - 1)^2 + m$

Ejemplo: $7 \leq (3 - 1)^2 + 3 = 7$

Algoritmo para *r*(*n*)

1	2	5	10	…
4	3	6	11	…
9	8	7	12	…
16	15	14	13	…
…	…	…	…	…

$\xrightarrow{r}$

1/1	2/1	3/1	…	m/1
1/2	2/2	3/2	…	m/2
1/3	2/3	3/3	…	m/3
…	…	…	…	…
1/m	2/m	3/m	…	m/m

Cálculo de ? si n está en la columna derecha:
Quitar a n el cuadrado anterior: $n - (sq(n) - 1)^2$
Ejemplo: si $n = 6$: ? = $6 - (3-1)^2 = 2$

Algoritmo para *r*(*n*)

1	2	5	10	…
4	3	6	11	…
9	8	7	12	…
16	15	14	13	…
…	…	…	…	…

$\xrightarrow{r}$

1/1	2/1	3/1	…	m/1
1/2	2/2	3/2	…	m/2
1/3	2/3	3/3	…	m/3
…	…	…	…	…
1/m	2/m	3/m	…	m/m

Cálculo de ? si n está en la renglón inferior:
$sq(n)^2 - n + 1$
Ejemplo: si $n = 8$: ? = $9 - 8 + 1 = 2$

Algoritmo y formula para *r*(*n*)

1	2	5	10	…
4	3	6	11	…
9	8	7	12	…
16	15	14	13	…
…	…	…	…	…

$\xrightarrow{r}$

1/1	2/1	3/1	…	m/1
1/2	2/2	3/2	…	m/2
1/3	2/3	3/3	…	m/3
…	…	…	…	…
1/m	2/m	3/m	…	m/m

$$r(n) = \begin{cases} \dfrac{sq(n)}{n-(sq(n)-1)^2} & si\ n \leq (sq(n)-1)^2 + sq(n) \\ \dfrac{sq(n)^2 - n + 1}{sq(n)} & en\ otro\ caso \end{cases}$$

Pero hay otras funciones...

1	2	4	7	…
3	5	8	…	…
6	9	…	…	…
10	…	…	…	…
…	…	…	…	…

$\xrightarrow{r}$

1/1	2/1	3/1	…	m/1
1/2	2/2	3/2	…	m/2
1/3	2/3	3/3	…	m/3
…	…	…	…	…
1/m	2/m	3/m	…	m/m

Y otros algoritmos!

Algoritmos

- Función: independiente de la representación
- Algoritmo: dependiente de la representación
- Fórmula: expresión de un algoritmo!

Problemas

- Dos maneras de pensar acerca de los problemas:
  - Como un problema de decisión (declarativo):
    - 1/2 ∈ {1/1, 2/1, 2/2, 1/2, 3/1, 3/2, 3/3, 2/3, 1/3,…} ?
  - Como un proceso que transforma una entrada en cierta salida (Procedural):

1/2 → Algoritmo → si

  - Pero en este caso, podemos decidir en base a la estructura de las cadenas!

Problemas

- Pero que tal los primos:
  - Como un problema de decisión:
    111 ∈ {11, 111, 11111, 1111111, 11111111111,…} ?
  - Como un proceso que transforma una entrada en cierta salida:

$$\textit{111} \longrightarrow \boxed{\text{Algorithm}} \longrightarrow \text{yes}$$

  - ¿Podemos ordenar los primos?
  - Conocemos el algoritmo, pero no la función!
  - Para conocer la función habría que enumerar los primos
  - ¿Qué reflexiones podemos sacar?

Teoría de la complejidad

- Todo algoritmo requiere recursos:
  - Pasos de procesamiento (tiempo)
  - Memoria
- El problema es complejo si lo expresamos tanto:
  - Como problema de decisión
  - Como proceso algorítmico