Ciencias de la Computación I

Lenguajes Formales

Condiciones Aprobación Cursada 2012

• 3 evaluaciones prácticas (por comisiones) y un examen parcial, con dos recuperatorios.

• Evaluaciones prácticas se califican con A (aprobado) o D (desaprobado) y no tienen recuperatorio.

• Parcial, 1er y 2do recuperatorios se califican con notas entre 1 y 10. Para aprobar en cualquiera de las tres instancias calificación ≥ 4

• Nota de cursada C (igual o superior a 4 para estar aprobada):

C = A + 0,70 * B siendo

A = cantidad de evaluaciones prácticas aprobadas
B = nota de parcial, 1er ó 2do recuperatorio, siendo B ≥ 4

Cantidad de evaluaciones prácticas aprobadas	Nota parcial o recuperatorios	Cursada
0	5 ó menos	Desaprobada
0	6 ó más	Aprobada
1	4 ó menos	Desaprobada
1	5 ó más	Aprobada
2 ó 3	Menos de 4	Desaprobada
2 ó 3	4 ó más	Aprobada

Final de Últimos Temas

- Fechas diciembre 2012 y febrero-marzo 2013

Cantidad de evaluaciones prácticas aprobadas	Nota del examen **parcial** para rendir el final de últimos temas
3	6 ó más
2	7 ó más
1	8 ó más
0	9 ó más

Nota del examen final de la materia:

F = 0.7 * U + 3 siendo

U = Nota del examen final de últimos temas, siempre que U ≥ 4

Fuera de estas fechas, se rinde final regular tradicional.

Ciencias de la Computación

- Estudio de los procedimientos computacionales
- Solución a los problemas

Procedimiento computacional → Programa
Fórmula
Modelo abstracto
…

Ciencias de la Computación

Ejemplos Simples: CASO 1

- Dados dos números naturales, encontrar el máximo común divisor

Entrada Salida

m, n → mcd → s

Ej: si m=9 y n=6 Algoritmo de Euclides

9 | 6
3   1

6 | 3
0   2

→ mcd(9,6)=3

- Dada una lista de n elementos y un elemento, determinar si el elemento pertenece a la lista

```
read(elem);
i:= 0;
repeat
    i := i + 1
until (elem = A[i] or  i = n);
if (elem = A[i]) then
   writeln('Elemento encontrado')
else
  writeln('Elemento no encontrado')
```


Ciencias de la Computación

En los ejemplos del CASO 1 construimos un procedimiento computacional que:

Para cualquier entrada SIEMPRE devuelve un resultado

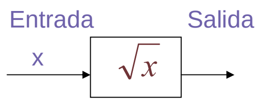
Garantiza finitud

PROCEDIMIENTO EFECTIVO= ALGORITMO

Ciencias de la Computación

Ejemplos Simples: CASO 2

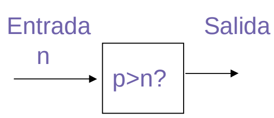
- Encontrar la raíz cuadrada exacta de un número natural x



Ej1: si x=4 $\sqrt{4} = 2$

Ej2: si x=2 $\sqrt{2} = 1.4142....?$

- Encontrar el número perfecto más grande que un número natural n



Ej1: si n=5 $p = 6 = \overbrace{3+2+1}^{\text{divisores}}$

Ej2: si n=10^{100} $p\,?$

Ciencias de la Computación

En los ejemplos del CASO 2 construimos un procedimiento computacional que:

NO SIEMPRE devuelve un resultado

↓

NO SIEMPRE garantiza finitud; depende de la entrada

↓

PROCEDIMIENTO

Ciencias de la Computación

Límites computacionales

¿Cuándo un problema tiene solución computacional?

¿Qué es un algoritmo?

¿Qué es un procedimiento?

Ciencias de la Computación

- Clase de problemas
  - Para cierta clase de problemas existe solución algorítmica
  - Para otros problemas sólo se puede diseñar un procedimiento
- Jerarquía de Máquinas Abstractas (modelos teóricos)

  Máquina de Turing tiene el poder computacional suficiente para resolver cualquier problema para el cual existe solución computacional
- Límites computacionales están dados por los límites de los procedimientos

Problemas y Lenguajes Formales

- Todo problema se puede describir mediante un lenguaje
- Cuanto más precisa y formal la definición del lenguaje, tanto más precisa será la definición del problema
- Definir un lenguaje plantea dos problemas:
  - determinar qué elementos pertenecen al lenguaje
  - generar los elementos del lenguaje
- Se debe distinguir lenguaje natural de lenguaje formal

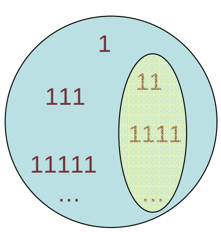
Lenguajes Formales

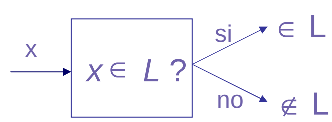
PROBLEMA: RECONOCIMIENTO DE LENGUAJES

Ejemplo: Números unarios

L={ x / x es unario y x es par}

L={11,1111,111111,….}





Lenguajes Formales

Alfabeto

Conjunto finito no vacío de símbolos indivisibles

Ejemplos

A = {0,1, 2, 3, 4, 5, 6, 7, 8, 9}

B = {0,1}

C = {a, b, c }

~~D = { a, b, ab}~~

No son símbolos indivisibles ab se forma con a y b

Lenguajes Formales

Cadena:

Sucesión finita de símbolos, sobre un alfabeto *A*

$w = s_1 s_2 \ldots s_n$ donde $s_i \in A$, para $1 \leq i \leq n$

s_i para $1 \leq i \leq n$ ocurre en posición i de la cadena

Por convención:

ε denota la cadena vacía

Ejemplos para el alfabeto A = {a, b, c}

$w_1 = abc$ $w_2 = \text{abb}$ $w_3 = aaaaaaa$ $w_4 = \varepsilon$

Lenguajes Formales

OPERACIONES SOBRE CADENAS

$w_1 = a_1a_2\ldots a_n$ $w_2 = b_1b_2\ldots b_m$ w_1, w_2 cadenas definidas sobre alfabeto A

- **Longitud de una cadena**
  $|w_1| = |a_1a_2..a_n| = n$
  Sea el alfabeto A = {a, b, c}
  Ej: |abc| = 3 |aaaaaa| = 6 |ε| = 0

- **Igualdad de cadenas**
  $w_1 = w_2$
  Ej: abc = abc   abc ≠ ac

- **Reversa de una cadena**
  $w_1^R = a_n \ldots a_2 a_1$
  Ej. $(abc)^R$ = cba $(aaaab)^R$ = baaaa

- **Concatenación de cadenas**
  $w_1 . w_2 = a_1a_2\ldots a_nb_1b_2\ldots b_m$
  Ej. abc.aaaaa = abcaaaaa

Propiedades concatenación
$w_1.\varepsilon = \varepsilon.w_1 = w_1$
$w_1.w_2 \neq w_2.w_1$

- **Potencia k-ésima de una cadena**
  $w_1^0 = \varepsilon$
  $w_1^k = \underbrace{w_1 . w_1 \ldots . w_1}_{(\text{k-veces})}$
  Ej. $ac^0b = ab$ $(ab)^2$ = abab abc^2 = abcc

Lenguajes Formales

Clausura de un alfabeto A

A*: Conjunto de todas las posibles cadenas sobre A

$$A^* = \bigcup_{i=0}^{i=\infty} A^i$$

A^i es el conjunto de todas las cadenas de longitud i sobre *A*

Ejemplo: A={a, b, c}

A^0 ={ε } A^1 ={a, b, c } A^2 ={a, b, c }{a, b, c }={aa, ab, ac, ba, bb, bc, ca, cb, cc}

A*={ε, a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, aab, aac,….}

Lenguajes Formales

<u>Lenguaje</u>:
Un lenguaje L sobre un alfabeto *A* es un subconjunto de *A**

$$L \subseteq A^*$$

- Un lenguaje puede ser finito o infinito
- Un lenguaje se puede definir por comprensión o por extensión

<u>Ejemplos</u>: Lenguajes sobre el alfabeto $A = \{a, b, c\}$

$A^*=\{\varepsilon,a,b,c,aa,ab,ac,ba,bb,bc,ca,cb,cc,aaa,aab,aac,\ldots\}$

$L_1 = \varnothing$
$L_2 = \{\varepsilon\}$
$L_3 = \{a, b, aa, bb, ab, ba\}$
$L_4 = \{a, aa, aaa, aaaa, \ldots\} = \{a^n \;/\; n \geq 1\}$
$L_5 = \{a^n b^n \;/\; n \geq 1\} = \{ab, aabb, aaabbb, aaaabbbb, \ldots\}$

Lenguajes Formales

<u>Operaciones con Lenguajes</u> Dados $L_1 \subseteq A^*$ y $L_2 \subseteq A^*$

- $L_1 \cup L_2 = \{ w \in A^* / w \in L_1$ o $w \in L_2\}$ (unión)
- $L_1 \cap L_2 = \{ w \in A^* / w \in L_1$ y $w \in L_2\}$ (intersección)
- $L_1 - L_2 = \{ w \in A^* / w \in L_1$ y $w \notin L_2\}$ (diferencia)
- $\overline{L_1} = \{ w \in A^* / w \notin L_1\} = A^* - L_1$ (complemento)
- $L_1 . L_2 = \{ w_1 . w_2 \in A^* / w_1 \in L_1$ y $w_2 \in L_2 \}$ (concatenación)

Propiedades de la concatenación
1) $L_1 . \varnothing = \varnothing = \varnothing . L_1$
2) $(L_1 . L_2) . L_3 = L_1 . (L_2 . L_3)$
3) $L_1 . L_2 \neq L_2 . L_1$
4) $L_1 . (L_2 \cup L_3) = L_1 . L_2 \cup L_1 . L_3$
5) $L_1 . (L_2 \cap L_3) \neq L_1 . L_2 \cap L_1 . L_3$

Lenguajes Formales

Dado L $\subseteq A^*$

- Potencia del lenguaje L

  $L^0 = \{\varepsilon\}$

  $L^k = L.L.L....L$ (k veces)

- Clausura del lenguaje L

$$L^* = \bigcup_{i=0}^{i=\infty} L^i = L^0 \cup L^1 \cup L^2 ....$$

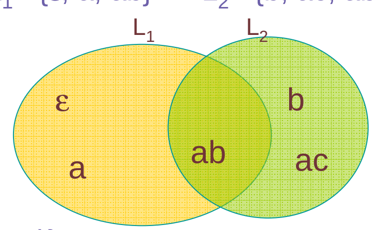
- Reversa del lenguaje L

  $L^R = \{ w^R \in A^* / w \in L \}$

Lenguajes Formales

<u>Ejemplo</u>

Dados L_1 y L_2 sobre A = {*a, b, c*} $L_1 = \{\varepsilon, a, ab\}$ $L_2 = \{b, ac, ab\}$



$L_1 \cup L_2 = \{\varepsilon, a, ab, b, ac\}$

$L_1 - L_2 = \{\boldsymbol{\varepsilon}, a\}$

$L_1^{\,R} = \{\varepsilon, a, ba\}$

$\overline{L_1} = \{w / w \in A^* \text{ y } (w \neq \varepsilon \text{ y } w \neq ab \text{ y } w \neq a)\}$

$L_1^* = L_1^0 \cup L_1^1 \cup L_1^2 \cup L_1^{3..} = \{\varepsilon\} \cup \{\varepsilon, a, ab\} \cup \{\varepsilon, a, ab, aa, aab, aba, abab\} \cup ..$

$L_1 \cap L_2 = \{ab\}$

$L_1.L_2 = \{b, ac, ab, aac, aab, abb, abac, abab\}$