



## Introducción a las ciencias de la computación Ejercicios para el examen de recuperación

1. Construya una máquina de Turing para los siguientes problemas:
  - a) Construya una máquina de Turing que reciba una cadena sobre el alfabeto  $\Sigma = \{a, b\}$  y haga una copia enseguida de la cadena original dejando un símbolo blanco que las separe. Por ejemplo, al recibir la cadena *ababb*, la máquina presentará *ababbBababb* como salida.
  - b) Una máquina de Turing que reciba una cadena sobre el alfabeto  $\Sigma = \{a, b\}$  y muestre como salida la cadena de entrada invertida. Por ejemplo, al recibir la cadena *ababb*, la máquina presentará *bbaba* como salida.
2. Obtenga expresiones regulares que representen los siguientes lenguajes:
  - a) El lenguaje formado por **todas** las cadenas de 0 y 1 que no contienen la subcadena 011.
  - b) El lenguaje formado por **todas** las cadenas de 0 y 1 que contengan a lo más un dígito 1.
  - c) El conjunto de cadenas sobre  $\Sigma = \{0, 1, 2\}$  donde después del último 0 (al menos habrá un 0) no aparece ningún 1.
3. Describa el lenguaje que generan las siguientes gramáticas:
  - a)  $S \rightarrow aSbS \mid bSaS \mid \epsilon$ .
  - b)  $S \rightarrow aSa \mid bSb \mid c$ .
  - c)  $S \rightarrow aBcD$   
 $aBc \rightarrow ac$   
 $cD \rightarrow \epsilon$   
 $cD \rightarrow ccD$
4. Construya gramáticas que generen los siguientes lenguajes:
  - a) Todas las cadenas de 0 y 1 con un número distinto de dígitos 0 y dígitos 1.
  - b) Todas las cadenas de longitud par (longitud 0,2,4,6,...) sobre el alfabeto  $\Sigma = \{a, b\}$ .
  - c) El lenguaje representado por la siguiente expresión regular  $\mathbf{a^* + b^* + (ab)^*}$ .
5. Mencione cuál es el tipo de las gramáticas que construyó en el ejercicio anterior.
6. Convierta los siguientes números a la base indicada:
  - a) El número  $(10110110)_2$  a base decimal.
  - b) El número  $(10110110)_2$  a base hexadecimal.
  - c) El número  $(1256)_8$  a base binaria.
  - d) El número  $(1256)_{16}$  a base octal.
  - e) El número  $(F1A2)_{16}$  a base decimal.

- f) El número  $(1345)_{10}$  a base binaria.
- g) El número  $(1345)_{10}$  a base hexadecimal.
- h) El número  $(1345)_{10}$  a base octal.
7. Escriba en **lenguaje ensamblador** (utilizando las directivas vistas en clase) los siguiente programas:
- a) Reescriba el programa visto en clase que calculaba  $x^n$  a partir de los números enteros  $x$  y  $n$  dados por el usuario.
- b) Un programa que determine si un número  $n$ , introducido por el usuario, es impar o no <sup>1</sup>. Para realizar este programa suponga que la UCP cuenta con una nueva instrucción, MOD. Esta instrucción tiene la siguiente especificación:
- MOD-i <dato> Divide el valor contenido en el acumulador entre el número que viene a continuación, pero en el acumulador NO se almacena el resultado de la división sino el **residuo**. Por ejemplo, si en el acumulador tenemos un 5 y se ejecuta MOD-i 2, en el acumulador se almacenará un 1 (residuo de la división  $5 \div 2$ ).
- c) Un programa que calcule y muestre el resultado de la suma  $0 + 1 + 2 + \dots + n$ , donde  $n$  es un número dado como entrada.

**No olvide estudiar y comprender temas importantes como, por ejemplo, los siguientes:**

- Concepto de algoritmo.
- Complejidad de los algoritmos.
- Problemas tratables e intratables; problemas decidibles e indecidibles (o no computables).
- Hipótesis de Church-Turing.
- Modelo de von Neumann (¿Cuál es su característica principal?).
- El ciclo de ejecución de la UCP, el lenguaje máquina, y otros temas del funcionamiento de la computadora.

---

<sup>1</sup>Recuerde que un número  $n$  es impar si el residuo es 1 al dividirlo por 2.