Departamento de Tecnologías de la Información

Área de Ciencias de la Computación e Inteligencia Artificial

Modelos Avanzados de Computación

Examen de febrero

EJERCICIO 1 (1 punto)

En criptografía, el cifrado XOR es un algoritmo de cifrado basado en el operador binario XOR. Una cadena de texto puede ser cifrada aplicando el operador de bit XOR sobre cada uno de los caracteres utilizando una clave. Para descifrar la salida, solo hay que volver a aplicar el operador XOR con la misma clave.

Por ejemplo, la cadena "Wiki" (01010111 01101001 01101011 01101001 en 8-bit ASCII) puede ser cifrada con la clave 11110011 de la siguiente manera:

```
  01010111 01101001 01101011 01101001
⊕ 11110011 11110011 11110011 11110011
  ----------------------------------
= 10100100 10011010 10011000 10011010
```

Y viceversa para descifrarlo:

```
  10100100 10011010 10011000 10011010
⊕ 11110011 11110011 11110011 11110011
  ----------------------------------
= 01010111 01101001 01101011 01101001
```

(a) Desarrolle la operación de encriptado XOR con la clave 11110011 por medio de un Autómata de Mealy.

(b) Desarrolle la operación de encriptado XOR con la clave 11110011 por medio de un Autómata de Moore.

EJERCICIO 2 (1 punto)

Considere un lenguaje formado por cadenas construidas como una lista de elementos donde cada elemento puede ser un símbolo ‘a’ o una lista de elementos entre paréntesis. Por ejemplo, la cadena “((a) a (a) ((a)))” pertenece al lenguaje pero las cadenas “) a (” o “((a)” no pertencen al lenguaje.

(a) Desarrolle un Autómata de Pila que reconozca dicho lenguaje.

(b) Enuncie y demuestre el Lema de Bombeo para Autómatas de Pila.

EJERCICIO 3 (2 puntos)

Diseñar una Máquina de Turing que tome como entrada dos palabras formadas por los símbolos del alfabeto {0,1,2}, separadas por el símbolo ‘#’, y comprueba si son iguales. Por ejemplo, para la entrada (#2101#2101~~bb~~) devuelve que sí son iguales (#1~~bb~~) y mientras que para la entrada (#2101#212~~bb~~) devuelve que no son iguales (#0~~bb~~).

EJERCICIO 4 (1 punto)

Considere la siguiente gramática libre de contexto, expresada en Forma Normal de Chomsky, donde C es el símbolo inicial.

C → S P	P → TP S	L → TS B	LL → B L
C → B L	S → B L	B → TL CC	TP → **paralelo**
C → TL CC	S → TL CC	B → **id**	TS → **serie**
C → **id**	S → **id**	CC → C TR	TL → **parab**
P → TP PP	L → TS LL	PP → S P	TR → **parce**

Verifique que la cadena "**id serie parab id paralelo id parce**" pertenece al lenguaje definido por la gramática por medio del algoritmo de Cocke-Younger-Kasami.

EJERCICIO 5 (1 punto)

Sea A_{TM} el lenguaje formado por las cadenas <*M,w*> tales que *M* es la codificación de una Máquina de Turing y *w* es una cadena aceptada por dicha máquina.

Demuestre que el lenguaje A_{TM} es indecidible. (Problema de la aceptación)

EJERCICIO 6 (2 puntos)

Considere el modelo de computación de las funciones recursivas. Asuma que las siguientes funciones ya han demostrado ser recursivas primitivas: *Suma(x,y), Producto(x,y), Potencia(x,y), Decremento(x), RestaAcotada(x,y), Signo(x), SignoNegado(x), Min(x,y), Max(x,y), And(x,y), Or(x,y), Not(x), Igual(x,y), Mayor(x,y), Menor(x,y), MayorOIgual(x,y), MenorOIgual(x,y), If(x,y,z)*.

Demuestre que la función *Resto*(*x,y*), que calcula el resto de la división entera (*x % y*) es una función primitiva recursiva.

EJERCICIO 7 (1 punto)

(a) ¿Qué es un lenguaje NP?

(b) ¿Qué es un verificador de un lenguaje?

(c) Demuestre que un lenguaje es NP si y solo si es verificable polinomialmente.

EJERCICIO 8 (1 punto)

(a) ¿Qué es un problema PSPACE?

(b) ¿Qué es un problema NPSPACE?

(c) ¿Qué es un problema PSPACE-completo?