Modelos Avanzados de Computación

Segunda convocatoria

EJERCICIO 1 (1.5 puntos)

Considere el proceso de multiplicar por 3 un número binario. Esta operación se puede realizar desplazando el número 1 bit y sumándo el mismo número, es decir, multiplicando por (2+1). (NOTA: los estados deben expresar si los bits desplazados son 0 o 1 y si el resultado de la suma se lleva 0 o 1)

```
101101 x 3           101101
                    1011010
                  ----------
                   10000111
```

(a) Desarrolle la operación “multiplicar por 3” por medio de un Autómata de Mealy.

(b) Desarrolle la operación “multiplicar por 3” por medio de un Autómata de Moore.

EJERCICIO 2 (1 punto)

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

L → NM F	MT → SM T	T → **producto**	RP → **rpar**
L → LP LC	MT → SM L	F → ML T	NM → **num**
L → T MT	T → NM F	LC → L RP	SM → **plus**
L → **producto**	T → LP LC	LP → **lpar**	ML → **mul**

Verifique que la cadena “**num mul producto plus num mul lpar producto plus producto rpar**” pertenece al lenguaje definido por la gramática por medio del algoritmo de Cocke-Younger-Kasami.

EJERCICIO 3 (2 puntos)

Diseñar una Máquina de Turing que tome como entrada una palabra formada por los símbolos del alfabeto {a,b} y devuelve la longitud de la palabra expresada en código binario. Por ejemplo, para la entrada (#ababba~~bb~~) devuelve el número 6 (#011~~bb~~).

NOTA: El número binario está escrito de izquierda a derecha, es decir, la cifra menos significativa a la izquierda.

EJERCICIO 4 (1.5 puntos)

(Problema de la parada)

Sea $HALT_{TM}$ el lenguaje formado por las cadenas por las cadenas <*M*,*w*> tales que *M* es la codificación de una máquina de Turing y *w* es una cadena que hace que dicha máquina termine (ya sea aceptando o rechazando).

Demuestre que el lenguaje $HALT_{TM}$ es indecidible.

EJERCICIO 5 (1.5 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 *Division*(*x,y*), que calcula la división entera (*x / y*) es una función primitiva recursiva.

EJERCICIO 6 (1.5 puntos)

(a) ¿Qué es un problema NP-completo?

(b) Demuestre que el problema CLIQUE (encontrar un clique de tamaño k en un grafo) es NP-Completo. Considere demostrado que los problemas SAT (satisfactibilidad de fórmulas lógicas) y 3SAT (satisfactiilidad de formúlas lógicas en formato 3-cnf) son NP-competos.

EJERCICIO 7 (1 punto)

(a) ¿Qué es un qubit?

(b) ¿Qué es una puerta cuántica?