Departamento de Tecnologías de la Información

Área de Ciencias de la Computación e Inteligencia Artificial

Universidad de Huelva

Modelos Avanzados de Computación

Examen de febrero

EJERCICIO 1 (1.5 puntos)

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

```
101101 x 3          0101101
                    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 (1.5 puntos)

Considere la siguiente gramática libre de contexto, expresada en Forma Normal de Chomsky.

T → **id**	C → A T	N → **not**
T → T C	D → O T	A → **and**
T → T D	E → T R	O → **or**
T → N T	L → **lparen**	
T → L E	R → **rparen**	

Verifique que la cadena "**not lparen id and id or id rparen and not id**" 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 dos cadenas formada por los símbolos del alfabeto {A,B} y separadas por el símbolo $ y acepte si la primera cadena aparece como subcadena en la segunda y rechace en caso contrario.

Por ejemplo, la entrada (#ABB$BAABBABAB~~bb~~) debe aceptarse pero la entrada (#ABB$BAABABAB~~bb~~) debe rechazarse.

EJERCICIO 4 (1.5 puntos)

Sea E_{TM} el lenguaje formado por las cadenas <*M*> tales que *M* es la codificación de una máquina de Turing que no reconoce ninguna entrada, es decir, cuyo lenguaje es el lenguaje vacío. Demuestre que el lenguaje E_{TM} es indecidible.

NOTA: Considere demostrado que los lenguajes A_{TM} (problema de la aceptación) y $HALT_{TM}$ (problema de la parada) son indecidibles.

EJERCICIO 5 (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 *Log(x,y)*, que calcula el logaritmo en base *y* del número entero *(x+1)*, es una función primitiva recursiva.

NOTA: El logaritmo está definido para números mayores o iguales a 1. Al utilizar el argumento (x+1) el caso base de la recursión es x=0.

$$Log(x, y) = Log_y(x+1) = z \mid y^z \le x+1 < y^{z+1}$$

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.