Departamento de Tecnologías de la Información

Área de Ciencias de la Computación e Inteligencia Artificial

Universidad de Huelva

Modelos Avanzados de Computación

Primera convocatoria

EJERCICIO 1 (1.5 puntos)

(a) Enuncie y demuestre el Lema de Bombeo para Autómatas Finitos.

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

EJERCICIO 2 (1 punto)

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

E → E A	F → M H	H → E N
E → T B	F → **id**	M → **lparen**
E → M C	A → O T	N → **rparen**
E → **id**	B → P F	O → **plus**
T → T D	C → E N	P → **prod**
T → M G	D → P F	
T → **id**	G → E N	

Verifique que la cadena "**lparen id plus lparen id prod id rparen rparen**" 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 cadena formada por los símbolos del alfabeto {A,B} y elimine la mitad posterior de la cadena. Si la longitud es impar el número de caracteres a eliminar se redondea a la baja. Por ejemplo, la entrada (#BAABBABAB~~bb~~) que tiene 9 caracteres debe eliminar los 4 últimos caracteres y generar como salida (#BAABB~~bb~~).

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 (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), Division(x,y), Resto(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)*.

(a) Construya como primitiva recursiva la función *Divisible(y,x)* que devuelve 1 si *x* es divisible entre algún número mayor que 1 y menor o igual que *y* (y devuelve 0 en caso contrario).

(b) Utilizando la función anterior, construya la función *Primo(x)*, que devuelve 1 si *x* es un número primo (es decir, no es divisible entre ningún número mayor que 1 y menor que *x*).

EJERCICIO 6 (1 punto)

(a) ¿Qué es un problema NP-completo?

(b) Demuestre que el problema CLIQUE es NP-Completo. Dado un grafo no dirigido G, se define un clique como un subgrafo en el que todos los nodos están conectados entre sí. El problema CLIQUE consiste en verificar si en el grafo G existe algún clique de tamaño *k*. 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 (1.5 puntos)

Enuncie y demuestre el teorema de no-clonación de qbits