**1. El problema de la parada (una vez más):** Definimos la función HALT(p; x) como una función booleana que nos indica si la máquina de Turing Mp representada por p para en un número finito de pasos para la entrada x. Por otro lado, sabemos (por diagonalización) que existe una función no computable UC definida como se describe a continuación. La función UC(s) es 0 si la máquina de Turing codificada por s devuelve 1 cuando recibe s como entrada; esto es, UC(s) = 0 cuando Ms(s) = 1. En cualquier otro caso, cuando la máquina de Turing devuelve cualquier otro valor o, simplemente, no para, UC(s) = 1. Dada la existencia de la función no computable UC, demostrar por reducción que la función HALT tampoco es computable.

NOTA: Para demostrar que UC no es computable, basta con que supongamos que UC es computable. Entonces, existiría una máquina de Turing M tal que M(s) = UC(s) para cualquier cadena de bits s. En particular, debería verificarse que M(M) = UC(M), lo cual, por la propia definición de UC, es imposible.

Sabemos que UC ≤ HALT, es decir, que la función no computable UC se reduce a HALT.

La función UC trabaja del siguiente modo:

0 si Ms(s)=1

UC(s) =

1. en otro caso

Construimos un algoritmo que nos permita calcular UC usando el problema de la parada del siguiente modo:

Muc(s){

if HALT(s,s) == 0{ //caso en el que la máquina no para

return 1;

}else{

if(Ms(s)==1)

return 0;

else

return 1;

}

}

Como UC no es computable, dicho algoritmo Muc no puede dar el resultado correcto, luego ha de fallar en la función HALT, es decir, el problema de la parada tampoco es computable.

**3. Spin glass**: Muestre cómo calcular el estado de mínima energía del sistema (`ground state' para los físicos) para materiales ferromagnéticos (Jij ≥0) reduciendo el problema al de calcular un corte mínimo en un grafo [Min Cut], problema que se puede resolver en tiempo polinómico.

Partimos de un grafo inicial GI que dispone en cada nodo de un spin si , en cada arista de una fuerza Jij ≥0 . Además en cada nodo actúa un campo externo hi.

Vamos a construir un grafo ponderado GP a partir de GI, haciendo uso de los valores si, Jij y hi, de manera que sean análogos. De este modo podremos resolver el problema de minimizar la energía del grafo inicial, obteniendo el corte mínimo del grafo ponderado.

Veamos cómo construimos GP:

Hacemos una copia del grafo inicial y asignamos a cada arista las fuerzas Jij. Añadiremos un par de nodos (nodo final s y nodo inicial t) para poder realizar el MINCUT.

Conectamos s y t con cada uno de los nodos que teníamos, creando dos nuevos conjuntos de aristas: Aristass y Aristast .A cada una de las aristas del conjunto de Aristass le asociamos su correspondiente hi (si hi>0) y a cada una de las aristas del conjunto de Aristast le asociamos su correspondiente hi (si hi<0). De este modo ya tenemos el grafo ponderado “análogo” al inicial.

Cuando calculamos un corte en este GP, estamos los dejando los spit de signo positivo a un lado, y los de signo negativo a otro. Calcular el corte mínimo sobre GP nos da el calcular el estado de mínima energía del sistema representado por el grafo inicial GI.