Resultados de la Comparacion de los Modelos de Computo

Resumen

Mostraremos los conceptos que nos permitiran enunciar los teoremas centrales que desprenderan los problemas computacionales en Cuestion asi como su solución.

En esta parte seremos muy breves en que es los conceptos que son elementales para la solucion y presentacion de los problemas Computacionales en Cuestion.

Definiciones y conceptos

Caracteristicas del Modelo de Computo Distribuido LOCAL

- Cada proceso se inicializa en el mismo tiempo
- Los eventos Comunicacion y Computo se ejecutan en rondas y de manera sincronizada

Nocion de solucionar un problema con un algoritmo Distribuido

Definicion
Decimos que un problema $P_{I,O}$ es solucionado por un algoritmo Distribuido si existe un algoritmo π tal que es correcto, i.e $\pi(I) = O$

Extension de la nocion de Decibilidad en el sentido Distribuido
Definicion
Decimos que una cadena $x = x_1 \dots x_n$ es decidible si existe un proceso v_i tal que $v_i(x_i)$ la decide en el sentido usual de las maquinas de Turing.

Una vez que tenemos en mente las nociones anteriores es entonces cuando nos emprendemos a hacer una conexion entre ambos mundos en el sentido de que podemos introducir la nocion de simulacion en un sentido formal en teoria de la Computacion
Definicion
Sean MT y un algoritmo distribuido (En el sentido en el que presentamos la extension del modelo general) i.e π y los lenguajes que deciden estos dos objetos, i.e $L(TM)$ y $L(\pi)$ entonces decimos que la maquina de Turing simula el algoritmo distribuido si $L(TM) = L(\pi)$ y reciprocamente.

Una vez que tenemos definido que es la simulacion en un sentido formal, es entonces cuando podemos anunciar los teoremas de Conexion con todos los ingredientes de desarrollo de estos Objetos.

Por un lado queremos decir que si tomamos una maquina de Turing entonces existe un π que simula a dicha maquina de Turing diganos TM

Y por otro lado queremos decir que si tomamos un algoritmo π este algoritmo puede ser simulado por la maquina de Turing TM

Entonces declaremos lo anterior en Teoremas, de la siguinte manera:

Theorem 1 *$\forall TM$ Maquina de Turing con $L(TM)$ entonces $\exists \pi$ tal que $L(\pi) = L(TM)$*

Theorem 2 *$\forall \pi$ Algoritmo Distribuido con $L(\pi)$ el Lenguaje que decide dicho algoritmo, entonces $\exists TM$ tal que $L(TM) = L(\pi)$*

Iniciaremos con el primer Teorema, entonces para ello, tomamos una maquina de Turing, y lo que queremos es exhibir que existe un algoritmo π distribuido tal que simula a TM

Algorithm 1 $\pi(x_i \dots x_n)$

```
π(x_1 ... x_n)
for all r = 0 to n do
    init v_i(x_i)
    if x_i ∈ L(TM) then
        return q_accept
    else
        send x_i
    end if
end for
finally return q_accept
```

En este Diseño del Algoritmo $\pi(x)$ donde $x = x_1, \dots, x_n$ es una cadena que pertenece al $L(TM)$ que es el lenguaje aceptado por la maquina de Turing, ya que por un lado del Teorema Tenemos todos los elementos de la maquina de Turing.
En el siguiente Teorema Encapsulara una de las cuestiones que tiene el algoritmo con respecto a como se esta haciendo el control de la cadena sobre las entradas de las subrutinas que tienen comunicación una a la otra para solucionar el problema.

Theorem 3 *Sea una cadena x, entonces dicha cadena es aceptada por la maquina de Turing si y solo si es aceptada por π. Donde π es un algoritmo Distribuido.*

Con base al Diseño del Algoritmo anterior y al teorema Anterior podemos, encontrar una cadena que es decidible por TM pero que no es decidible con nuestro algoritmo descrito anteriormente.
El siguiente programa hace que se solucione esta cuestion:

Algorithm 2 $\pi(x, G)$

```
for all r ← 1 to n do
    π(x_1, ..., x_n)
    init p_i(x)
    return q_accept
end for
```

Podemos observar que esta subrutina hace que el problema se trivialice; ya que estamos alimentando a cada proceso con toda la entrada que come π, lo cual rompe con el paradigma Distribuido, ya que todos los procesos tiene toda la informacion.

Sea una gráfica de diametro D, podemos diseñar el siguiente algoritmo solucionando de manera correcta el problema en cuestión P reduciendo un grado de libertad con la restricción anterior sobre el diametro de la gráfica, y con los dos programas anteriores como subrutinas podemos hacer los siguientes algoritmos:

Algorithm 3 $\pi(x, G)$

```
π_finally(x, G(D))
Donde x es una cadena en L(TM)
x ← (x_1 ... x_n)
for all r ← 1 to m do
    π(x, G(r))
    broadcast(v_0, G(r), M)
    Donde G(r) es una grafica de diametro r
end for
```