

Examen 1

Cruz Perez Ramón
315008148

October 24, 2020

1. Sea $G = (\pi, E)$, Sea τ un arbol BFS con raiz en el proceso $p_s \in \pi$
P.D. Existe una ejecucion del algoritmo para crear arboles generadores que tienen como resultado τ con p_s .

Usando el Algoritmo4 en Algoritmo1.pdf, sabemos que un BFS, tiene todos los proceso a distancia d de la raiz.

Por induccion sobre d :

Existe un arbol $G = (\pi, E)$ con distancia d de la raiz, en la ejecucion del algoritmo.

P.B.

Con $\pi = \{p_s\} \implies p_s$ esta a distancia 0.

Con $\pi = \{p_s, 1, 2\}$

Por Algoritmo, se inicia el proceso en $p_s \in \pi$.

Dado p_s , le manda a todos sus hijos $GO()$ y el sera la raiz.

Hijos_s, estan a una arista, distancia = 1 ,

$\implies \text{parent}_{hijos} = p_s, \text{expected_msgi} = 0,$

$\implies \text{send BACK}(i), \implies p_s \text{ children} = \text{children} \cup \{j\},$ por algoritmo. \implies el arbol generado tiene distancia d de la raiz.

H.I

Existe un arbol $G = (\pi, E)$ con distancia d de la raiz, en la ejecucion del algoritmo.

P.I

Existe un arbol $G = (\pi+1, E+1)$ con distancia $d+1$ de la raiz, en la ejecucion del algoritmo.

Por Algoritmo, se inicia el proceso en $p_s \in \pi + 1$.

Dado p_s , SPG: envia a todos sus hijos $GO()$ al mismo tiempo. y el sera la raiz.

SPG. Cada hijo, $\text{parent}_{hijo} = j$, con $j =$ quien le envio $Go()$, y se repite

el proceso.

Sup. que cada $Go()$ fue enviado sin ninguna complicacion hasta el proceso d ;

SPG. y por H.I. Cada Proceso esta a distancia d de las raiz, y es un arbol $G = (\pi, E)$,

\implies cada Proceso a distancia d , envia $GO()$,

Como cada hijo esta a una arista, por P.B. se puede decir que esta a distancia 1.

\implies Los procesos hijos estan con distancia $= d+1$,

\implies Existe un arbol $G = (\pi+1, E+1)$ con distancia $d+1$ de la raiz, en la ejecucion del algoritmo.

Fin induccion.

Ahora ya sabemos que existe un arbol con on distancia d de la raiz, por lo cual cumple la definicion de BFS.

\implies Existe una ejecucion del algoritmo para crear arboles generadores que tiene como resultado τ con p_s el proceso que inicia la ejecucion.

2. Sea $G = (\pi, E)$ un sistema distribuido asincrono. Un arbol BFS que solo la raiz sabe que ya acabo la construccion.

Algorithm 1 Termino el arbol

```
1: Inicia
2: if raiz == p then
3:   fin = True
4:   for i ∈ hijos do
5:     send GO(fin) to i
6:   end for
7: end if
8: mensajes = ||hijos||
9:
10: When GO(fin)
11: if hijos == ∅ then eshoja
12:   send BACK(fin) to padre
13: else
14:   for i ∈ hijos do
15:     send GO(fin) to i
16:   end for
17: end if
18:
19: When BACK(finhijo)
20: mensajes -= 1
21: if mensajes == 0 then
22:   if fin == finhijo then esraiz
23:     Fin algoritmo.
24:   else
25:     fin = finhijo
26:     send BACK(fin) to padre
27:   end if
28: end if
```

Es Correcto:

Pues se envia un mensaje hasta llegar a las hojas, y una vez que llega a las hojas se enteran que ya esta construido el arbol, por lo que las hojas le dicen a sus padres que ya esta construido y el padre no envia back a menos que halla recibido todos los mensajes de sus hijos. Asi hasta que la raiz recibe todos los mensajes de sus hijos.

Complejidad del algoritmo:

Numero de mensajes:

- Cada BACK es enviado como respuesta de algun GO.
- Un mensaje GO por arista del arbol, ida.
- Total de mensajes 2, uno de ida y uno de regreso $\implies 2e$.
- En particular el arbol es el minimo por lo que son $(n-1) = e$, aristas.
 $\implies 2(n-1)$

Complejidad tiempo:

- Nos toma 2D rondas (los mensajes GO y BACK)

Tamaño de los mensajes:

- Como solo envia un booleano son $\log(\text{bool})$ bits
3. Sea $G = (\pi, E)$ un sistema distribuido sincrono. Sea $M = \{m_1, m_2, \dots, m_k\}$

Algorithm 2 Diseminar el mensaje

```
1: crearArbol(G)
2: if  $p == p_s$  then esraiz
3:    $M = \{m_1, m_2, \dots, m_k\}$ 
4:   for  $m \in M$  do
5:     for  $h \in \text{Hijos}_s$  do
6:       send  $G_0(m)$  to  $h$ 
7:     end for
8:   end for
9: else
10:   $\text{lst} = []$ 
11: end if
12:
13: When  $G_0(m)$ 
14:  $\text{lst.append}(m)$ 
15: for  $h \in \text{Hijos}_p$  do
16:   send  $G_0(m)$  to  $h$ 
17: end for
```

Se creo un arbol generador, esto implicara que no se envien mensajes repetidos entre hermanos, y cada proceso que recibe y envia m ademas lo guarda en una lista.

Como el sistema es sincrono los mensajes llegaran en orden por lo que cada proceso tendra $\text{lst} = M$

Como un arbol manda $O(n)$ mensajes este enviara $O(nk)$ con $k = \|M\|$

4. Sea $G = (\pi, E)$ un sistema distribuido sincrono y sea $p_s \in \pi$ un proceso desde que empezara a ejecutar broadcast. Demuestra que broad-cast no puede ejecutarse en menos de D rondas, con D la distancia mas grande entre p_s y cualquier otro proceso.

Por contradiccion: P.D que el broad-cast no puede ejecutarse en menos de $R = (D+1)$ rondas, con D la distancias mas grande entre p_s y cualquier

otro proceso.

Por induccion sobre R:

P.B

sea G un nodo.

Entonces: $D = 0, \rightarrow R = D+1 = (0)+1 = 1, R = 1$

\Rightarrow Broad-cast se ejecuta en la ronda 0 que es la inicializacion y termina en la misma. ▼

Por Hip. no se puede ejecutar en menos rondas que son $R=D+1 = 1$

Fin induccion.

\Rightarrow Broad-cast no puede ejecutarse en menos de $R = D$ rondas, con D la distancias mas grande entre p_s y cualquier otro proceso.

Pues se uso contradiccion