Examen 2

Cruz Perez Ramón 315008148

21 de noviembre de 2020

- 1. Demuestra que el protocolo anterior es un protocolo que resuelve el k-set agreement.
 - Acuerdo:

Ya que cada proceso envia su valor y a lo más pueden fallar f, por lo menos se eviaron (n-f) valores.

 \implies k \le (n-f), con k = maximo de numero de valores.

 \Longrightarrow Sup. S.P.G que todo nodo envia un valor diferente y fallan f procesos, como en la linea 7 siempre se toma al minimo y solo tenemos k valores enviados.

 $\Longrightarrow ||W|| \le k$

Si no fallaran los f, al tomarse el minimo aseguramos que $\|W\| \le k$. (similar al caso con k =1)

■ Validez:

Sea $V = rec_{-}from_{i}$, valores de entrada.

En linea 7 simpre se toma al min. Entonces $\min(rec_from_i) \subseteq V$ \Longrightarrow en cada iteracion se va mandar el minimo, podemos asegurar que no tendremos de mas, $||W|| \leq ||V||$, por acuerdo. y como W es un sub-conjunto de $V \Longrightarrow W \subseteq V$

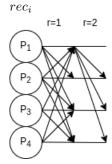
■ Terminacion:

Como tenemos f fallas y k tamaño, justo en la linea 3 termina cuando se llega al round $\lfloor \frac{f}{k} \rfloor + 1$, se tenga o no un valor acordado. Si estamos en el round $\lfloor \frac{f}{k} \rfloor + 1$ se envia el valor minimo (est_i) .

2. Sea $\Pi = \{p1, p2, p3, p4\}$ un sistema con n = 4 procesos en una grafica completa. Supongamos que los procesos p1 y p2 son bizantinos Muestra una ejecución del algoritmo del consenso con fallas bizantinas donde no pueden llegar a un acuerdo.

Sup. que P_1 empieza con 5 y P_2 empieza con 6.

Y si el valor mas frecuente es un empate, tomaremos el mas pequeño de



ronda = 1

$$\begin{array}{l} V_1=5,\,V_2=6,\,V_3=3,\,V_4=4\\ rec_1=\{3,6,4\},\,rec_2=\{5,3,4\},\,rec_3=\{4,6,5\},\,rec_4=\{6,5,3\}\\ most_freq_1=3,\,most_freq_2=3,\,most_freq_3=4,\,most_freq_4=3\\ occ_nb_1=1,\,occ_nb_2=1,\,occ_nb_3=1,\,occ_nb_4=1\\ \mathbf{ronda=2}\\ \mathbf{r}/2==1\colon\mathbf{send}\ 3\\ \mathbf{if}\ v_receivedfromP_{\underline{v}}\mathbf{:}\\ coord_val_1=5,\,coord_val_2=3,\,coord_val_3=3,\,coord_val_4=3\\ \mathbf{if}\ occ_nb_i>\frac{n}{2}+0\mathbf{:}\to 1>\frac{4}{2}+0\\ V_1=5,\,V_2=3,\,V_3=3,\,V_4=3\\ \end{array}$$

Fin de la ejecucion: (2(0+1) = 2)

 $V_1 = 5$

 $V_2 = 3$

 $V_3 = 3$

 $V_4 = 3$

Como $V_1 = 5$ es distinto a los demas, \Longrightarrow fallo el acuerdo.

Por que? Pues en el round 2, como P_1 no recibe un valor,

Ent. $coord_{-}val_{1} = v_{1}$, y $v_{1} = 5$.

Por lo que las rondas no son suficientes para llegar al acuerdo.

Entonces las condiciones para no llegar al acuerdo son:

No avisar que hay procesos bizantinos

No no respetar el minimo de fallo por n procesos. f $< \frac{n}{4}$

▶ Que el sistemas tenga menor o igual a 4 procesos.

Para llegar al acuerdo se necesita ejecutar 4 rondas minimo.

3. Sea α una ejecución y sean ϕ_1 y ϕ_2 dos eventos en α . Demuestra que si $\phi_1 \stackrel{\alpha}{\Rightarrow} \phi_2$ entonces $LC(\phi_1) < LC(\phi_2)$..

Como $\phi_1 \stackrel{\alpha}{\Rightarrow} \phi_2$ por def. de Happens Before. ϕ_1 paso antes que ϕ_2

Entonces, sea $LC(\phi_1)$ y $LC(\phi_2)$ marca de tiempo logica.

Usando el algorimo de Lamport. ϕ_1 incrementa su contador y se lo envia a ϕ_2 , por Happens Before.

Cuando ϕ_2 recibe, cambia su contador cambia su contador por el de ϕ_1 .

Pues el contador de ϕ_2 es menor, ya que pasa despues de ϕ_1 por Happens Before

$$\implies LC(\phi_1) < LC(\phi_2)$$

$$\implies \phi_1 \stackrel{\alpha}{\Rightarrow} \phi_2 \text{ entonces } LC(\phi_1) < LC(\phi_2) \blacksquare$$

4. Sea α una ejecución y sean ϕ_1 y ϕ_2 dos eventos en α . Demuestra que si $VC(\phi_1) < VC(\phi_2)$ entonces $\phi_1 \stackrel{\alpha}{\Rightarrow} \phi_2$.

Como tenemos dos relojes vectoriales de tamaño n. Para que sea $\phi_1 \stackrel{\alpha}{\Rightarrow} \phi_2$ se necesita cumplir 3 puntos:

■ Sup. que ϕ_1 y ϕ_2 estan en el mismo proceso. ⇒ por def. de reloj ϕ_1 esta antes de ϕ_2 , pues cuando es computo local incrementa su propio contador y es lo unico que varia.

$$\implies v_1 \neq v_2 \\ \implies \phi_1 \stackrel{\alpha}{\Rightarrow} \phi_2$$

 \blacksquare Sup. ϕ_1 y ϕ_2 son de diferentes procesos.

Como $VC(\phi_1) < VC(\phi_2)$, por def. de relojes vectoriales, si manda un mensaje, manda el vector, y ϕ_2 que es la que recibe, incrementa su contador y toma los maximo del vector recibido.

Por lo que la relacion entre vectores cambia solo en la posicion de ϕ_2

$$\Longrightarrow v_1 \neq v_2 \Longrightarrow \phi_1 \stackrel{\alpha}{\Rightarrow} \phi_2$$

 \blacksquare sup. existe un $VC(\phi)$ tal que: $VC(\phi_1) < VC(\phi) < VC(\phi_2)$

Como $VC(\phi_1) < VC(\phi)$, $\Longrightarrow \phi_1$ paso antes que ϕ por def. relojes vectoriales y relacion entre vectores.

Analogamente con $\phi < \phi_2$

como estan bien posicionados en el tiempo, cambio al menos un valor en $v_i[\mathbf{x}]$ entre los eventos, Ent. podemos decir:

$$\Longrightarrow v_1 \neq v \neq v_2$$

$$\phi_1 \stackrel{\alpha}{\Rightarrow} \phi \ y \ \phi \stackrel{\alpha}{\Rightarrow} \phi_2$$