

Computación Concurrente - Tarea 4

Damián Rivera González
Alexis Hernandez Castro

November 11, 2019

1. Para cada una de las siguientes ejecuciones contesta lo siguiente:

Inciso a)

- ¿Qué es $H|B$?
 $\langle B, cp.deq(b) \rangle$
 $\langle B, cp.dep(c) \rangle$
- ¿Qué es $H|r$? No existe el objeto r
- Transforma H en una subhistoria H'
 $\langle A, cp.enq(a, 3) \rangle$
 $\langle B, cp.deq(b) \rangle$
 $\langle A, cp : void \rangle$
 $\langle A, cp.enq(b, 2) \rangle$
 $\langle A, cp : void \rangle$
 $\langle B, cp : b \rangle$
 $\langle B, cp.deq(c) \rangle$
 $\langle B, cp : c \rangle$
- ¿ H' está bien formada? Sí, porque la proyección del hilo A y B son secuenciales
- ¿ H' secuencial? No, porque las invocaciones de métodos no tienen inmediatamente su regreso
- ¿ H' es linearizable? Marca los puntos de linearizabilidad. No es linearizable, porque se hace la invocación de $\langle B, cp.deq(c) \rangle$ sin embargo, c no ha sido ingresado en la cola.

Inciso b)

- ¿Qué es $H|B$?
 $\langle B, r.write(1) \rangle$
 $\langle B, r : void \rangle$
 $\langle B, r.read(1) \rangle$
 $\langle B, r : 1 \rangle$
- ¿Qué es $H|r$? $\langle B, r.write(1) \rangle$
 $\langle A, r.read(1) \rangle$
 $\langle C, r.write(2) \rangle$
 $\langle A, r : 1 \rangle$
 $\langle B, r : void \rangle$
 $\langle C, r : void \rangle$

$\langle B, r.read(1) \rangle$

$\langle B, r : 1 \rangle$

$\langle C, r.read(?) \rangle$

- Transforma H en una subhistoria $H' \langle B, r.write(1) \rangle$

$\langle A, r.read(1) \rangle$

$\langle C, r.write(2) \rangle$

$\langle A, r : 1 \rangle$

$\langle B, r : void \rangle$

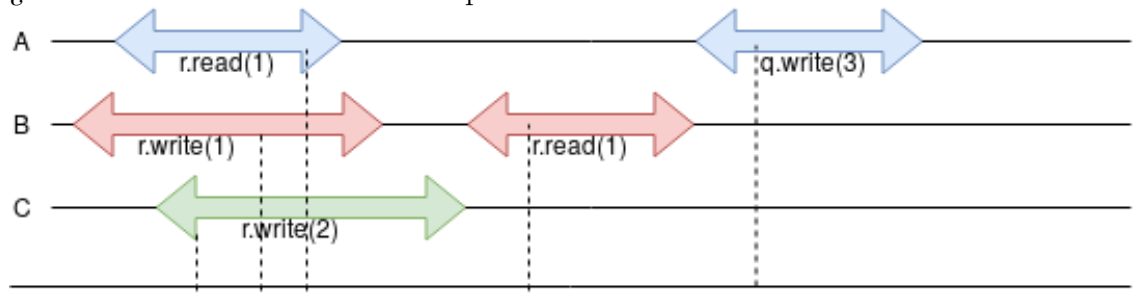
$\langle C, r : void \rangle$

$\langle B, r.read(1) \rangle$

$\langle B, r : 1 \rangle$

$\langle A, q.write(3) \rangle \langle A, q : void \rangle$

- ¿ H' está bien formada? Sí, porque la proyección de cada hilo es secuencial
- ¿ H' secuencial? No, porque las invocaciones no tienen inmediatamente su regreso
- ¿ H' es linearizable? Marca los puntos de linearizabilidad. Si es linearizable.



- Para cada una de las siguientes historias, reconstruye su ejecución y menciona si cumple con la propiedad de consistencia secuencial, de quietud y/o es linearizable. Marca los puntos de linearizabilidad.

a) $\langle C, r.write(1) \rangle$

$\langle B, r.read \rangle$

$\langle A, r.write(2) \rangle$

$\langle B, r : 1 \rangle$

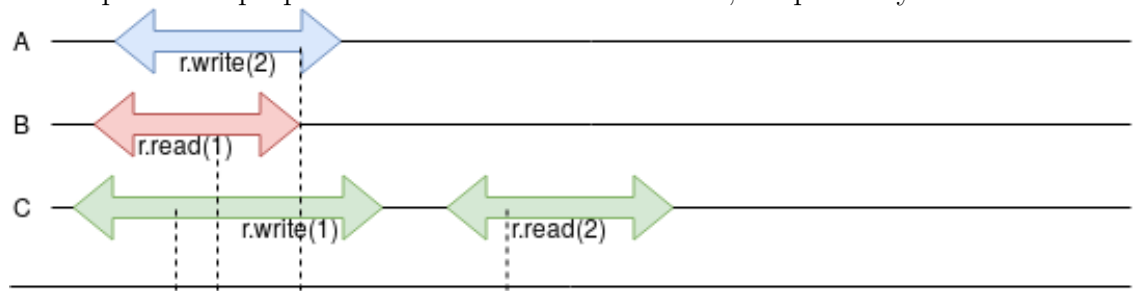
$\langle A, r : void \rangle$

$\langle C, r : void \rangle$

$\langle C, r.read \rangle$

$\langle C, r : 2 \rangle$

Sí cumple con la propiedad de consistencia secuencial, de quietud y es linearizable

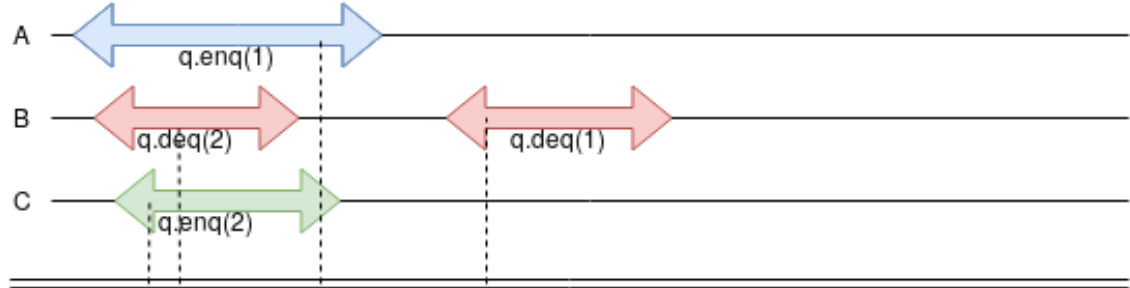


b) $\langle A, q.enq(1) \rangle$

$\langle B, q.deq() \rangle$

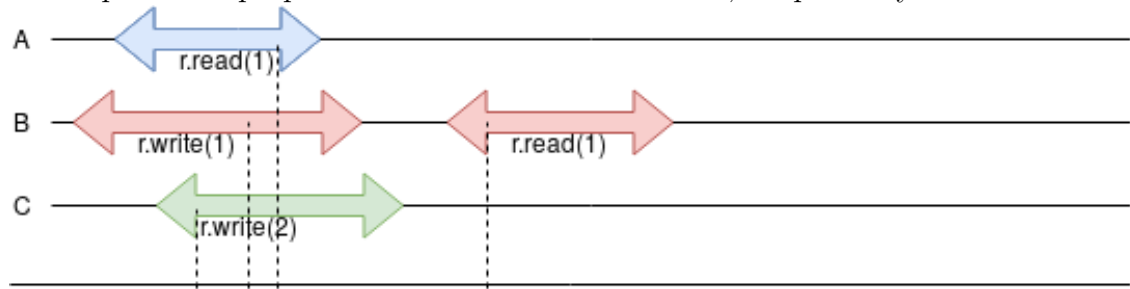
$\langle C, q.enq(2) \rangle$
 $\langle B, q : 2 \rangle$
 $\langle C, q : void \rangle$
 $\langle A, q : void \rangle$
 $\langle B, q.deq() \rangle$
 $\langle B, q : 1 \rangle$

Sí cumple con la propiedad de consistencia secuencial, de quietud y es linearizable



c) $\langle B, r.write(1) \rangle$
 $\langle A, r.read() \rangle$
 $\langle C, r.write(2) \rangle$
 $\langle A, r : 1 \rangle$
 $\langle B, r : void \rangle$
 $\langle C, r : void \rangle$
 $\langle B, r.read() \rangle$
 $\langle B, r : 1 \rangle$

Sí cumple con la propiedad de consistencia secuencial, de quietud y es linearizable



3. Para las historias H que hayan sido linearizables tanto en el primer como en el tercer ejercicio, da la historia linearizable. Justifica que es legal y muestra el conjunto de relaciones de precedencia que deben cumplirse.

Para todas las historias es legal porque en cada una se trabaja sobre un mismo objeto (ya sea r o q) y la proyección de cada objeto es la misma historia que damos (con excepción de la historia 1a, pero ambas proyecciones de los objetos están en la especificación secuencial de cada objeto) , la cual está en la especificación secuencial del objeto.

Historia 1b

$\langle C, r.write(2) \rangle$
 $\langle C, r : void \rangle$
 $\langle B, r.write(1) \rangle$
 $\langle B, r : void \rangle$
 $\langle A, r.read(1) \rangle$
 $\langle A, r : 1 \rangle$

$\langle B, r.read(1) \rangle$

$\langle B, r : 1 \rangle$

$\langle A, q.write(3) \rangle$

$\langle A, q : void \rangle$

Precedencia = $\{r.write(2) \rightarrow r.write(1), r.write(1) \rightarrow r.read(1), r.read(1) \rightarrow q.write(3)\}$
más todas las relaciones de precedencia que se dan por la transitividad

Historia 2a

$\langle C, r.write(1) \rangle$

$\langle C, r : void \rangle$

$\langle B, r.read(1) \rangle$

$\langle B, r : 1 \rangle$

$\langle A, r.write(2) \rangle$

$\langle A, r : void \rangle$

$\langle C, r.read(2) \rangle$

$\langle C, r : 2 \rangle$

Precedencia = $\{r.write(1) \rightarrow r.read(1), r.read(1) \rightarrow r.write(2), r.write(2) \rightarrow r.read(2)\}$
más todas las relaciones de precedencia que se dan por la transitividad

Historia 2b

$\langle C, q.enq(2) \rangle$

$\langle C, q : void \rangle$

$\langle B, q.deq(2) \rangle$

$\langle B, q : 2 \rangle$

$\langle A, q.enq(1) \rangle$

$\langle A, q : void \rangle$

$\langle B, q.deq(1) \rangle$

$\langle B, q : 1 \rangle$

Precedencia = $\{q.enq(2) \rightarrow q.deq(2), q.deq(2) \rightarrow q.enq(1) \rightarrow q.deq(1)\}$ más todas las relaciones de precedencia que se dan por la transitividad

Historia 3b

$\langle C, r.write(2) \rangle$

$\langle C, r : void \rangle$

$\langle B, r.write(1) \rangle$

$\langle B, r : void \rangle$

$\langle A, r.read(1) \rangle$

$\langle A, r : 1 \rangle$

$\langle B, r.read(1) \rangle$

$\langle B, r : 1 \rangle$

Precedencia = $\{r.write(2) \rightarrow r.write(1), r.write(1) \rightarrow r.read(1), r.read(1) \rightarrow r.read(1)\}$
más todas las relaciones de precedencia que se dan por la transitividad

4. Da un ejemplo de una ejecución que tenga consistencia de quietud pero no secuencial; y otra que tenga consistencia secuencial pero no de quietud. Argumenta porque las ejecuciones no cumplen con cada propiedad.

Consistencia de quietud, no secuencial: No se pueden ordenar las llamadas de los métodos de manera que se lea 1 y 2 sin a ver escrito 2 y 1. Si se leyó 1 debió escribirse 2 y 1, pero ya no se podría leer 2, si se leyó 2 se escribió 2, pero previamente no se

pudo haber leído 1.



Consistencia Secuencial, no de quietud: Antes del espacio de quietud la última escritura debe ser 0 ó 1, pero después del espacio de quietud se debe poder leer 0 y 1, pero la última escritura solo puede tener un valor. Por lo que si la última escritura (antes del tiempo de quietud) fue 0, no podríamos leer 0 y después 1, y viceversa, si la última escritura (antes del tiempo de quietud) fue 1, no podríamos leer 1 y luego 0.



- Suponiendo que en el sistema existe un solo hilo, ¿Es la consistencia de quietud equivalente a la consistencia secuencial? Argumenta formalmente.

Supongamos que existe un solo hilo A.

Supongamos que el A cumple con la consistencia de quietud.

Entonces sabemos que entre espacios de quietud se tiene un orden entre las llamadas a los métodos de manera que cumplen con la especificación del objeto. Notemos que por ser un único hilo, la ejecución de los métodos es secuencial, lo cual indica que para cada llamada a un método tenemos inmediatamente su regreso, $e_i \rightarrow e_{i+1}$, donde e es la llamada al método y e_{i+1} es el regreso de este. Denotemos a esto como m_i . Ahora, sabemos que los espacios de quietud existen entre las llamadas (y regresos) de métodos, m_i y m_{i+1} , así la consistencia secuencial mantiene en mismo orden de llamadas a métodos que en la consistencia de quietud, con lo cual se sigue cumpliendo la especificación del objeto y además $H|A$ será el mismo en ambas consistencias.

Supongamos que se cumple la consistencia secuencial.

Como es un solo hilo su ejecución será secuencial. Lo cual nos indica que los espacios de quietud solo existen entre las llamadas a métodos m_i y m_{i+1} , así como se cumple la especificación del objeto, la única manera de reordenar las llamadas a métodos entre espacios de quietud es la misma puesto que solo existe una única llamada m_k , y así en la consistencia de quietud se mantendrá el mismo orden de las llamadas que en la consistencia secuencial, por lo tanto cumplirá con la especificación del objeto y $H|A$ será la misma en ambas consistencias.

Por lo tanto se cumple la equivalencia.

- Al igual como le hicimos con la linearizabilidad, define formalmente el concepto de consistencia de quietud. HINT: Define la noción de un bloque sin quietud y agrega la precedencia de eventos (\rightarrow) entre ellos.

Definición. Un **bloque sin quietud**, es un conjunto de llamadas a métodos con sus respectivos regresos, acotado por la izquierda y derecha por un tiempo de quietud, de manera que si te colocas en un tiempo dentro de su ejecución la proyección de los métodos tendrá un regreso pendiente para alguna llamada.

Precedencia entre bloques sin quietud, sean dos bloques sin quietud, B_1 y B_2 , cada uno con su conjunto de llamadas a métodos

$$B_1 = \{m_{11}, m_{12}, \dots, m_{1n}\}, B_2 = \{m_{21}, m_{22}, \dots, m_{2n}\}$$

Diremos que B_1 precede a B_2 , denotado como $B_1 \rightarrow B_2$, si para toda llamada $m_{i1} \in B_1$ y toda llamada $m_{i2} \in B_2$ se tiene que $m_{i1} \rightarrow m_{i2}$.

Definición. Una ejecución cumple con consistencia de quietud, si para todo *bloque sin quietud*, B_i , existe una permutación entre las llamadas a métodos de cada B_i tal que para cada relación de precedencia $B_i \rightarrow B_{i+1}$ está cumple con la especificación del objeto.

7. Demuestra por qué la consistencia de **quietud** es composicional.
8. Considera un objeto p que utiliza dos registros. Por el ejercicio anterior, sabemos que si ambos registros cumplen con la consistencia de quietud, entonces p también la cumplirá ¿El caso contrario se cumple? Es decir, si p cumple con la consistencia de quietud entonces los registros también cumplen la consistencia de quietud? Esbozar una prueba o dar un contraejemplo.