

# Computación Concurrente - Tarea 4

Damián Rivera González  
Alexis Hernandez Castro

November 6, 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  $H' \langle B, r.write(1) \rangle$

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

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

$\langle A, r : 1 \rangle$

$\langle B, r : void \rangle$

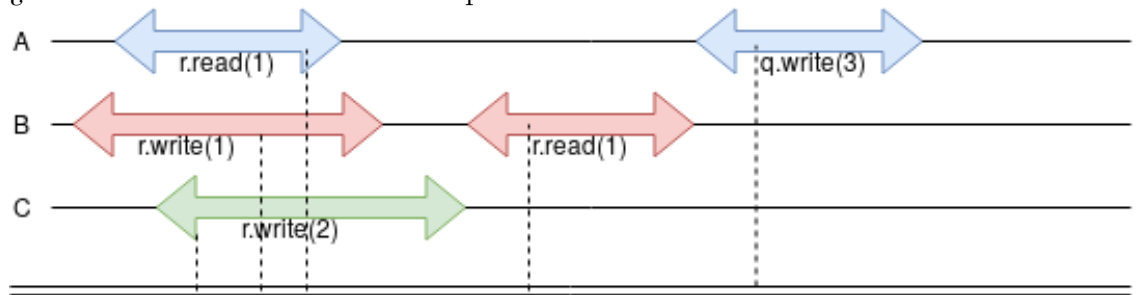
$\langle C, r : void \rangle$

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

$\langle B, r : 1 \rangle$

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

- ¿ $H'$  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.



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

$\langle B, r.read \rangle$

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

$\langle B, r : 1 \rangle$

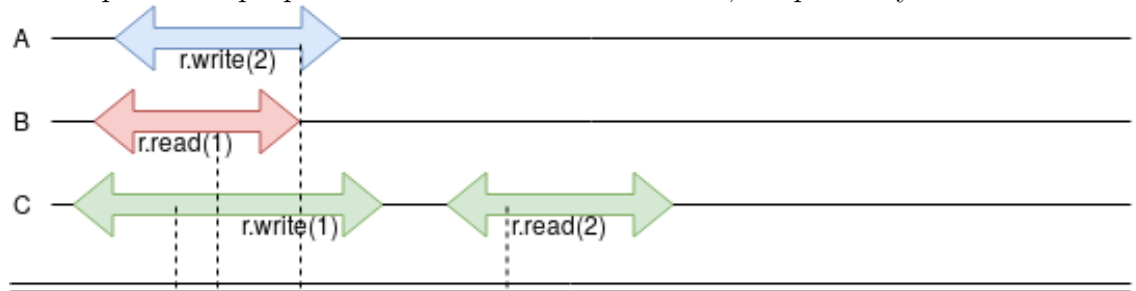
$\langle A, r : void \rangle$

$\langle C, r : void \rangle$

$\langle C, r.read \rangle$

$\langle C, r : 2 \rangle$

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

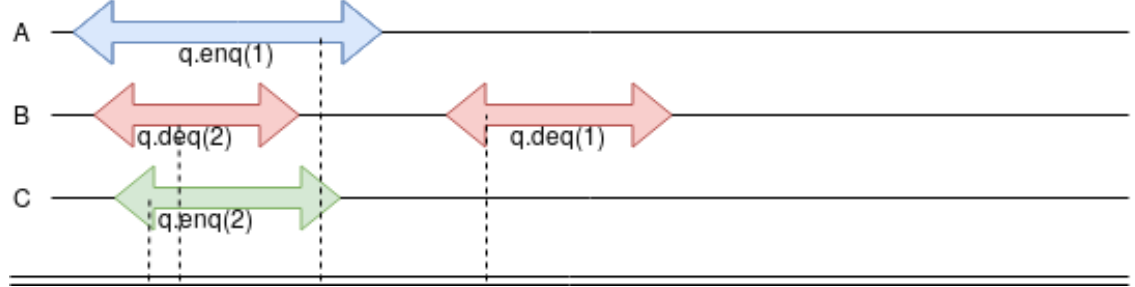


b)  $\langle A, q.enq(1) \rangle$

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

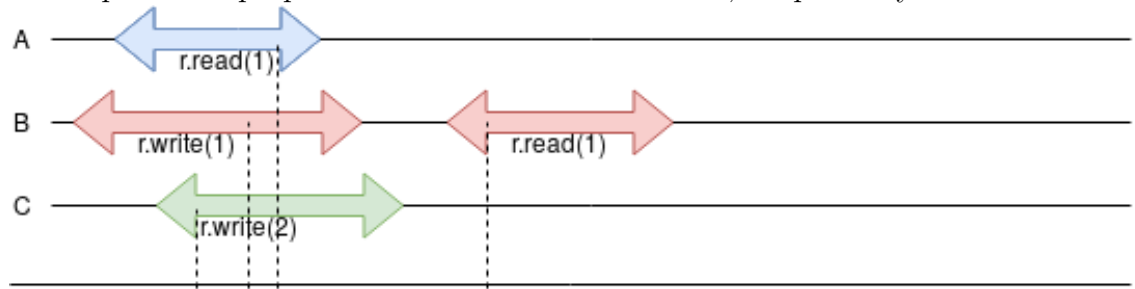
$\langle C, q.enq(2) \rangle$   
 $\langle B, q : 2 \rangle$   
 $\langle C, q : void \rangle$   
 $\langle A, q : void \rangle$   
 $\langle B, q.deq() \rangle$   
 $\langle B, q : 1 \rangle$

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.



5. Suponiendo que en el sistema existe un solo hilo, ¿Es la consistencia de quietud equivalente a la consistencia secuencial? Argumenta formalmente.  
Sí. Veamos la consistencia de quietud de un hilo, llamémosle A. Como la ejecución de un hilo es secuencial, entonces cada llamada a un método tiene inmediatamente su regreso

...  
 $\langle A, \text{objeto.método}() \rangle$   
 $\langle A, \text{objeto} : \text{regreso} \rangle$   
 ...

Con lo cual los tiempos de quietud de la ejecución de un hilo, están entre cada llamada de un método. Por esto si tenemos una llamada  $m_i$  que precede a la llamada  $m_{i+1}$  esto es  $m_i \rightarrow m_{i+1}$  entre cada tiempo de quietud solo podemos reordenar una única llamada a método  $m_x$  con lo cual en la consistencia de quietud del hilo A, cada relación de precedencia  $m_i \rightarrow m_{i+1}$  se mantiene igual y como la ejecución de un hilo es secuencial entonces la relación de precedencia  $m_i \rightarrow m_{i+1}$  en la consistencia secuencial se mantiene igual, por lo que al hacer  $H|A$  en la consistencia de quietud será la misma al hacer  $H|A$  en la consistencia secuencial.

6. 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.
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.