EXAMEN DE VALIDACIÓN Y VERIFICACIÓN DEL SOFTWARE. 14/1/2022.

APELLIDOS Y NOMBRE: .....

1) **(70 pts)** En cada ciclo, un proceso X puede estar en la cola de preparados (p) o en entrada/salida (s). Además, la CPU puede estar ocupada (c) por algún proceso o vacía ($\neg c$). Formula los siguientes enunciados en LTL

   – Si la CPU está vacía y el proceso X no está preparado, entonces X está en entrada/salida

   $$\Box(\neg c \wedge \neg p \rightarrow s)$$

   – Si el proceso X está preparado, la CPU debe estar ocupada en el siguiente ciclo.

   $$\Box(p \rightarrow \bigcirc c)$$

   – El proceso X no sufre inanición: si está en la cola, en algún momento sale de ella

   $$\Box(p \rightarrow \Diamond \neg p) \quad \equiv \quad \Box\Diamond\neg p$$

   – El proceso X nunca se bloquea permanentemente en entrada/salida

   $$\Box\Diamond\neg s$$

   – Entre dos períodos distintos de entrada/salida el proceso X pasa siempre por la cola de preparados

   $$\Box\neg(s \wedge \bigcirc\neg s \wedge \bigcirc(\neg p \,\mathcal{U}\, s))$$

   o, lo que es equivalente:

   $$\Box(s \wedge \bigcirc\neg s \rightarrow \bigcirc(p \,\mathcal{R}\, \neg s))$$

2) **(20 pts)** Explica cuál es la principal diferencia formal entre los contraejemplos para una propiedad de *safety* y los contraejemplos para una propiedad de *liveness*. Indica si la primera de las fórmulas que has obtenido para el ejercicio anterior es una condición de *safety* o de *liveness*.

   En los contraejemplos de una propiedad de *safety*, es suficiente con describir un *prefijo finito* de la traza, de modo que podemos parar en cuanto lleguemos a un estado en que la condición que se deseaba evitar aparezca. En los contraejemplos para *liveness* es necesario mostrar la

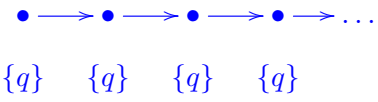
traza infinita, lo que en la práctica, sólo es posible si se usa un sufijo cíclico, es decir, que se repite indefinidamente. La condición $\Box(\neg c \wedge \neg p \rightarrow s)$ del ejercicio anterior es de *safety*. El contraejemplo a mostrar se detiene al llegar a un estado en que $\neg c \wedge \neg p \wedge \neg s$.

3) **(50 pts)**Dadas las fórmulas:

$$\alpha \stackrel{def}{=} \Box(p \,\mathcal{U}\, q) \qquad\qquad \beta \stackrel{def}{=} \Box(p \wedge \Diamond q)$$

demostrar cada dirección de la equivalencia o, si no se cumple, presentar un contraejemplo
$\models \alpha \rightarrow \beta$ ¿se cumple? [ ]-Sí [X]-No
Explicación:

Como contraejemplo podemos tomar la traza que tiene todos sus estados iguales a $\{q\}$ (es decir, q cierto y p falso)



Es fácil ver que $M, 0 \models \Box(p \,\mathcal{U}\, q)$ porque $M, i \models p \,\mathcal{U}\, q$ para cualquier $i \geq 0$ dado que todos los estados satisfacen q, que es la condición de parada del until. Por otro lado, está claro que $M, 0 \models \Box(p \wedge \Diamond q)$ no se cumple dado que esto último equivale a $M, 0 \models \Box p \wedge \Box\Diamond q$ es decir, tendría que darse p siempre cierto y q con frecuencia infinita. Se puede comprobar que $\Box p$ no se da: de hecho, p siempre es falso.

$\models \beta \rightarrow \alpha$ ¿se cumple? [X]-Sí [ ]-No
Explicación:
Como decíamos en el apartado anterior, $M, 0 \models \Box(p \wedge \Diamond q)$ equivale a $M, 0 \models \Box p \wedge \Box\Diamond q$, es decir, que cualquier modelo de β satisface que p siempre es cierto y que q aparece con frecuencia infinita. Para demostrar $M, 0 \models \Box(p \,\mathcal{U}\, q)$ debemos probar que $M, i \models p \,\mathcal{U}\, q$ para todo $i \geq 0$. Como q se da con frecuencia infinita aparecerá en algún punto j futuro de i, es decir, $M, i \models \Diamond q$ o, lo que es lo mismo, $\exists j \geq i : \; M, j \models q$. Pero como p se da en todos los estados también podemos garantizar $\forall k \in [i..j-1] : \; M, k \models p$ y ya obtenemos la definición de $M, i \models p \,\mathcal{U}\, q$.