EXAMEN DE VALIDACIÓN Y VERIFICACIÓN DEL SOFTWARE. 18/01/2018.

APELLIDOS Y NOMBRE: .....

1) **(50 pts)** Un vehículo puede circular (c) si tiene el certificado de la ITV en regla (r). Si no lo tiene en regla, debe superar una inspección (i). Formula los siguientes enunciados en LTL

   - Nunca está permitido circular cuando el certificado no está en regla
     Varias formas de expresarlo

     $$\Box\neg(c \wedge \neg r)$$
     $$\neg\Diamond(c \wedge \neg r)$$
     $$\Box(c \rightarrow r)$$
     $$\Box(\neg r \rightarrow \neg c)$$

   - Si el certificado no está en regla, seguirá sin estarlo, a no ser que supere una inspección

     $$\Box(\neg r \rightarrow \neg r \ \mathcal{W} \ i)$$

   - Tras superar una inspección, el certificado volverá a estar en regla en algún momento

     $$\Box(i \rightarrow \bigcirc\Diamond r)$$

     si entendemos que "tras" significa que, al menos, pasa una situación. Podría aceptarse

     $$\Box(i \rightarrow \Diamond r)$$

     si entendemos que puede estar ya en regla en el mismo instante en que pasa la inspección.
   - Todo vehículo dejará de circular más tarde o más temprano

     $$\Diamond\Box\neg c$$

   - Un certificado puede dejar de estar en regla infinitas veces

     $$\Box\Diamond\neg r$$

     o, si entendemos que "dejar de estar" significa que antes lo estuvo:

     $$\Box\Diamond(r \wedge \bigcirc\neg r)$$

2) **(40 pts)** Dadas la fórmulas

$$\alpha \stackrel{def}{=} \Diamond(p \to q) \qquad\qquad \beta \stackrel{def}{=} q \vee \Diamond\neg p$$

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

Como contraejemplo, tomemos una secuencia M que cumpla p cierto en todos los estados y q cierto en alguno que no sea $i = 0$. Por ejemplo:

$$\underset{\bullet}{\{p\}} \longrightarrow \underset{\bullet}{\{p,q\}} \longrightarrow \underset{\bullet}{\{p\}} \longrightarrow \underset{\bullet}{\{p\}} \longrightarrow \underset{\ldots}{\ldots}$$

Esta interpretación satisface $\Diamond(p \to q)$ porque existe un punto, $i = 1$, en el que se cumple $M, 1 \models p \to q$, ya que s_1 hace cierto q y eso hace cierta la implicación. Sin embargo, para hacer cierta la fórmula $q \vee \Diamond\neg p$ necesitaríamos satisfacer una de las dos partes de la disyunción, y ninguna de ellas se cumple. Por un lado, $M, 0 \not\models q$ porque q es falso en s_0. Por otro lado, $M, 0 \not\models \Diamond\neg p$ porque no hay ningún punto i en el cual p sea falso. Por tanto, $M, 0 \models \alpha$ pero $M, 0 \not\models \beta$ y la implicación no es válida.

$\models \beta \to \alpha$ ¿se cumple? [X]-Sí [ ]-No
Explicación:

Como β es una disyunción, podemos separar la prueba en dos casos:

- CASO 1: Probar que $\models q \to \alpha$ esto es, $\models q \to \Diamond(p \to q)$.
  Este caso es sencillo ya que $M, 0 \models q$ implica $M, 0 \models p \to q$, pues esto último no es más que la disyunción $\neg p \vee q$. Pero entonces, ya hemos encontrado un caso $i = 0$ en que $M, i \models p \to q$ y, por tanto, $M, 0 \models \Diamond(p \to q)$.
- CASO 2: Probar que $\models \Diamond\neg p \to \alpha$ esto es, $\models \Diamond\neg p \to \Diamond(p \to q)$.
  Esta situación es similar. $M, 0 \models \Diamond\neg p$ equivale a decir que hay un punto $i \geq 0$ en el que $M, i \models \neg p$. Pero entonces, $M, i \models p \to q$ ya que la implicación es lo mismo que $\neg p \vee q$. Como hemos encontrado un punto $i \geq 0$ en el que se cumple la implicación, entonces $M, 0 \models \Diamond(p \to q)$.

3) **(10 pts)** Explica en qué consiste la técnica de reducción de orden parcial (*partial order reduction*) usada a veces en *model checking* para disminuir la explosión combinatoria.

Consiste en detectar cuándo el orden de entrelazado no determinista entre distintos procesos es irrelevante, y comprobar sólo una de las posibles ordenaciones sin comprobar todas las demás, ya que es innecesario. Un ejemplo típico es en el que tenemos n distintos procesos con instrucciones I_i, $i \in \{1, \ldots, n\}$ que realizan cambios locales pero no en memoria compartida. En ese caso, el efecto global final de las $n!$ posibles ordenaciones de entrelazado es siempre el mismo y podemos tomar una cualquiera (por ejemplo, ordenar por el número de proceso).