EXAMEN DE VALIDACIÓN Y VERIFICACIÓN DEL SOFTWARE. 22/1/2015.

APELLIDOS Y NOMBRE: .....

1) **(25 puntos)** En la fase de test del nuevo teléfono móvil *Hayfón*, deseamos comprobar que siempre que se teclea tres veces consecutivas un código PIN incorrecto, el teléfono queda bloqueado hasta que alguien teclee correctamente el código PUK, lo que debería desbloquear el teléfono inmediatamente después, o bien queda bloqueado para siempre, si nadie teclea el PUK. Expresar en LTL la propiedad que se desea probar usando las proposiciones e="error: tecleado PIN incorrecto"; k="tecleado PUK correcto"; y b="teléfono bloqueado".

   El momento en que se teclean tres veces consecutivas un PIN incorrecto viene delimitado por la fórmula

$$e \wedge \bigcirc e \wedge \bigcirc \bigcirc e$$

   Esto es, si estamos en el instante i, la fórmula $\bigcirc e$ será cierta en el $i+1$ y la fórmula $\bigcirc \bigcirc e$ en el $i+2$. El enunciado no especifica a partir de qué momento se debe bloquear el teléfono. Supondremos que se bloquea a partir de la situación $i+3$. Por ejemplo, si se bloquease para siempre, la fórmula tendría este aspecto:

$$\Box(e \wedge \bigcirc e \wedge \bigcirc \bigcirc e \rightarrow \bigcirc \bigcirc \bigcirc \Box b)$$

   Sin embargo, nos piden que se bloquee hasta que alguien teclee el PUK (k). Una forma inmediata de representar esto sería

$$\Box(e \wedge \bigcirc e \wedge \bigcirc \bigcirc e \rightarrow \bigcirc \bigcirc \bigcirc(b \,\mathcal{U}\, k))$$

   es decir, en $i+3$ se cumple que b se mantiene cierto hasta que k. Como nos piden que se desbloquee el teléfono justo después de teclear el PUK, refinamos la especificación del siguiente modo

$$\Box(e \wedge \bigcirc e \wedge \bigcirc \bigcirc e \rightarrow \bigcirc \bigcirc \bigcirc(b \,\mathcal{U}\, (k \wedge \bigcirc \neg b)\ )\ )$$

   Sin embargo, ahora hay dos cosas que no están completamente especificadas. Primero, cuando se teclea el PUK se entiende que el teléfono aún debería estar bloqueado (se desbloquea justo después):

$$\Box(e \wedge \bigcirc e \wedge \bigcirc \bigcirc e \rightarrow \bigcirc \bigcirc \bigcirc(b \,\mathcal{U}\, (k \wedge b \wedge \bigcirc \neg b)\ )\ )$$

   Segundo, no queda claro si k es la primera vez que sucede, o quizá también pudo suceder en alguna otra situación de las que b fue cierto. Para asegurarnos de esto último, a la izquierda de $\mathcal{U}$ exigimos que $b\dagger \wedge \neg k$, es decir "bloqueado y no tecleando PUK"

$$\Box(e \wedge \bigcirc e \wedge \bigcirc \bigcirc e \rightarrow \bigcirc \bigcirc \bigcirc(\ (b \wedge \neg k) \,\mathcal{U}\, (k \wedge b \wedge \bigcirc \neg b)\ )\ )$$

   Para finalizar, la fórmula de arriba *exige* que tras los tres errores consecutivos de PIN *existe siempre un momento futuro* en que alguien teclea el PUK. Sin embargo, el enunciado dejaba libre la posibilidad de que esa situación nunca llegase a producirse. La solución a este último problema es sencilla: basta con reemplazar el operador "until" $\mathcal{U}$ por su versión débil $\mathcal{W}$:

$$\Box(e \wedge \bigcirc e \wedge \bigcirc \bigcirc e \rightarrow \bigcirc \bigcirc \bigcirc(\ (b \wedge \neg k) \,\mathcal{W}\, (k \wedge b \wedge \bigcirc \neg b)\ )\ )$$

NOTA: recuérdese que:

$$\alpha \,\mathcal{W}\, \beta \quad \leftrightarrow \quad (\alpha \,\mathcal{U}\, \beta) \vee \Box\alpha$$

es decir, en nuestro caso, con $\alpha = (b \wedge \neg k)$ el "weak until" deja abierta la posibilidad de que $\Box(b \wedge \neg k)$, o sea, siempre bloqueado y sin teclear PUK.

2) **(25 puntos)** Demostrar que las fórmulas:

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

son equivalentes o, si no lo son, encontrar un contraejemplo.

Para probar la equivalencia, usaremos las condiciones de satisfacción de una fórmula. Para ello probaremos que M es modelo de α, $M, 0 \models \alpha$, si y sólo si es modelo de β, $M, 0 \models \beta$.

Lo demostraremos encadenando expresiones equivalentes:

$$\begin{aligned} M, 0 \models \alpha \quad &\Leftrightarrow \quad M, 0 \models \bigcirc(p \,\mathcal{U}\, q) \\ &\Leftrightarrow \quad M, 1 \models p \,\mathcal{U}\, q \\ &\Leftrightarrow \quad \exists i \geq 1 \text{ such that } M, i \models q \text{ and } \forall j \in [1, i-1] : M, j \models p \end{aligned}$$

Ahora bien, en lugar de tomar un número i arbitrario mayor o igual que 1, podemos tomar el valor anterior $k = i - 1$. Esto implica cambiar i por $k + 1$ en la expresión de arriba:

$$\begin{aligned} &\Leftrightarrow \quad \exists k + 1 \geq 1 \text{ such that } M, k + 1 \models q \text{ and } \forall j \in [1, k + 1 - 1] : M, j \models p \\ &\Leftrightarrow \quad \exists k \geq 0 \text{ such that } M, k + 1 \models q \text{ and } \forall j \in [1, k] : M, j \models p \end{aligned}$$

Del mismo modo, en lugar de tomar un número j que varía entre 1 y k, podemos tomar esos valores decrementados en 1 $\forall h \in [0, k-1]$ de manera que cada j es en realidad $h + 1$.

$$\Leftrightarrow \quad \exists k \geq 0 \text{ such that } M, k + 1 \models q \text{ and } \forall h \in [0, k-1] : M, h + 1 \models p$$

Por último, por definición de la semántica de los operadores $\bigcirc$ y $\mathcal{U}$ podemos observar que:

$$\begin{aligned} &\Leftrightarrow \quad \exists k \geq 0 \text{ such that } M, k \models \bigcirc q \text{ and } \forall h \in [0, k-1] : M, h \models \bigcirc p \\ &\Leftrightarrow \quad M, 0 \models \bigcirc p \,\mathcal{U}\, \bigcirc q \\ &\Leftrightarrow \quad M, 0 \models \beta \end{aligned}$$

3) **(25 puntos)** Demostrar que las fórmulas:

$$\alpha \stackrel{def}{=} \bigcirc\Diamond q \qquad\qquad \beta \stackrel{def}{=} \Diamond \bigcirc q$$

son equivalentes o si no lo son, encontrar un contraejemplo. (NOTA: si es posible, usar el ejercicio anterior y la equivalencia válida $\bigcirc\top \leftrightarrow \top$).

Sabiendo que el operador $\Diamond$ se puede definir en función del $\mathcal{U}$ del siguiente modo:

$$\Diamond q \quad \leftrightarrow \quad \top \,\mathcal{U}\, q \tag{1}$$

entonces podemos encadenar las siguientes equivalencias:

$$
\begin{array}{lcll}
\bigcirc\Diamond q & \leftrightarrow & \bigcirc(\top \,\mathcal{U}\, q) & \text{usando equivalencia (1)} \\
 & \leftrightarrow & \bigcirc\top \,\mathcal{U}\, \bigcirc q & \text{usando el resultado del ejercicio anterior} \\
 & \leftrightarrow & \top \,\mathcal{U}\, \bigcirc q & \text{usando la equivalencia } \bigcirc \top \leftrightarrow \top \\
 & \leftrightarrow & \Diamond \bigcirc q & \text{usando otra vez (1) pero en sentido contrario}
\end{array}
$$

4) **(25 puntos)** Explicar brevemente qué significa que el problema de satisfactibilidad para LTL es PSPACE-completo.

El problema de satisfactibilidad para LTL consiste en decidir si una fórmula temporal α tiene un modelo LTL o no, es decir, si existe al menos un modelo M tal que $M, 0 \models \alpha$. El hecho de ser PSPACE quiere decir que este problema de decisión puede ser resuelto por una máquina de Turing que emplea un espacio (o cantidad de memoria) cuyo tamaño guarda una relación *polinomial* con el tamaño n de la fórmula de entrada α. Al ser PSPACE-*completo*, cualquier problema de decisión en la clase PSPACE puede ser transformado en un problema de satisfactibilidad LTL en un número de pasos polinomial con respecto a n.