

# Lenguajes de Programación

26-6-2014

Nombre:

DNI:

NOTA: Es necesario un mínimo de 3 ptos ptos<sup>1</sup> en la prueba para sumar las prácticas correspondientes. La duración del examen es de 2 horas.

1. (1.5 ptos) Implementar un predicado `cart(Conj1, Conj2, Comp)` que se verifique sii `Comp` es el producto cartesiano de `Conj1` y `Conj2`.

NOTA: Podemos suponer que los argumentos son listas sin elementos repetidos.

Ejemplo: Una respuesta a la pregunta `cart([3,1], [a,b], X)` es `X = [[3,a],[3,b],[1,a],[1,b]]`, pero el programa sería también correcto si facilitase cualquier otra permutación de `X`.

---

<sup>1</sup>50% de la puntuación total de teoría.

2. (1'5 ptos) Describir una Máquina de Turing que implemente la resta entera, definida como sigue:

$$m \dot{-} n := \begin{cases} m - n & \text{si } m \geq n \\ 0 & \text{en otro caso} \end{cases}$$

3. (1'5 ptos) Implementar una función LISP del tipo `interseccion(Conj1, Conj2)` que devuelva el valor de la intersección de sus argumentos.

NOTA: Consideraremos que un conjunto es una lista de elementos no repetidos, sin incurrir en efectos colaterales. Podemos suponer que cada uno de los argumentos no contienen, por tanto, elementos repetidos.

4. (1.5 pts) Explicar cuál es la relación entre la recursividad izquierda en las cláusulas y las ramas infinitas en un árbol de resolución PROLOG.