Febrero 1998 (MD 2)

1. Encontrar el número de soluciones enteras de la siguiente ecuación:

$$x + y + z + t + u = 14$$

   donde $0 \leq x,\ u;\ 2 \leq y,\ z,\ t$.

   Utilizando este resultado (o de otra forma) calcular cuántos subconjuntos de cuatro elementos tiene $C = \{1, 2, 3, \ldots, 15\}$ que no contengan enteros consecutivos. Subconjuntos como por ejemplo: $\{1, 3, 7, 10\}$ o $\{2, 5, 11, 15\}$.

2. Una permutación σ de n elementos se llama *desarreglo* si no tiene ningún punto fijo (i.e. $\sigma(i) \neq i\ \forall i = 1..n$.) Denotando $D(n)$ el número de desarreglos de n objetos, demostrar que:

$$n! = \sum_{k=0}^{n} \binom{n}{k} D(k).$$

3. Describir el uso de las funciones generatrices para resolver recurrencias. Dar una justificación del método.

4. Definimos un *abanico* de orden n como un grafo de vértices $\{0, 1, \ldots, n\}$ con $2n - 1$ ejes definidos por: El vértice 0 se conecta, mediante un eje, con cada uno de los otros vértices, y el vértice k se conecta mediante un eje con el vértice $k+1$ para todo k, $1 \leq k < n$.

   Calcular el número de árboles de recubrimiento (*spanning trees*) que posee un abanico de orden n. Justificar la respuesta.

5. Describe como se representa la lógica de orden superior en el systema HOL. Describe los términos y los tipos. Justifica la necesidad del tipado con algún ejemplo.

6. Describe las tres reglas de inferencia primitivas de HOL ASSUME (introducción de hipótesis), DISCH (eliminación de hipótesis) y MP (modus ponens).

   La función `dest_thm` descompone un teorema en un par formado por una lista de hipótesis y una conclusión. El tipo `goal` es una abreviatura de `(term)list # term`.

   Consideremos la siguiente sesión en Hol88:

```
#let Th3 = ASSUME "t1==>t2";;
Th3 = . |- t1 ==> t2
#dest_thm Th3;;
["t1 ==> t2"],"t1 ==> t2" : goal
#let Th4 = DISCH "t1==>t2" Th3;;
Th4 = |- (t1 ==> t2) ==> t1 ==> t2
#let Th5 = ASSUME "t1:bool";;
Th5 = . |- t1
#let Th6 = MP Th3 Th5;;
Th6 = .. |- t2
```

   Las hipótesis de `Th6` pueden inspeccionarse con la función `hyp`, quer devuelve la lista de hipótesis de un theorema.

```
#hyp Th6;;
["t1 ==> t2"; "t1"] : term list
```

   Se puede hacer que HOL imprima todas las hipótesis explicitamente usando la función `print_all_thm` en lugar de la por defecto `print_thm`.

```
#top_print print_all_thm;;
- : (thm -> void)
#Th5;;
t1 |- t1
#Th6;;
t1 ==> t2, t1 |- t2
#let Th7 = DISCH "t1==>t2" Th6;;
Th7 = t1 |- (t1 ==> t2) ==> t2
#let Th8 = DISCH "t1:bool" Th7;;
Th8 = |- t1 ==> (t1 ==> t2) ==> t2
```

   Describe cada uno de los pasos de la sesión. La sucesión de teoremas `Th3, Th5, Th6, Th7, Th8` constituye una prueba en HOL. ¿De qué teorema?.

   Escribir la sesión anteror en notación lógica estándar.