MODELOS ABSTRACTOS DE CÁLCULO
Resumen para problemas. Curso 2008/2009.

A. Problemas de calculabilidad:

1. Para demostrar que un lenguaje L es semidecidible hay que encontrar un programa $\widetilde{p}$ tal que
   i) Si $x \in L$ entonces $\widetilde{p}(x) \downarrow$ y $\widetilde{p}(x) = 1$.
   ii) Si $x \in L$ entonces $\widetilde{p}(x) \downarrow$ y $\widetilde{p}(x) = 0$ ó $\widetilde{p}(x) \uparrow$.
2. Para demostrar que un lenguaje L no es decidible o no es semidecidible se puede utilizar $H = \{p, x \,|\, p(x) \downarrow\}$ y demostrar lo siguiente:
   i) $H \leq_m L \Rightarrow L$ no decidible.
   ii) $\bar{H} \leq_m L \Rightarrow L$ no semidecidible.
3. Además se puede utilizar L decidible $\Leftrightarrow L$ y $\bar{L}$ semidecidibles.

B. Para demostrar que un problema A es NP-completo:

1. Ver que A está en NP.
   i) Definir el problema de comprobación comp$A$.
   ii) Ver que efectivamente:
      a) Si $x \in A$ entonces $\exists y$ tal que $(x, y) \in$comp$A$.
      b) Si $(x, y) \in$ comp$A$, entonces $x \in A$
   iii) Dados x dato de A y (x, y) dato de comp$A$, existen constantes c_1 y c_2 tales que $|(x, y)| \leq c_1|x|^{c_2}$.
   iv) Ver que comp$A$ está en P (es decir, se puede resolver A con un algoritmo que se ejecuta en tiempo polinómico en la longitud de la entrada).
2. Ver que A es NP-difícil. Para ello hay que elegir un problema NP-completo B y demostrar que B se reduce a A en tiempo polinómico, es decir $B \leq_m^p A$. Para ello, dado x dato de B definir $f(x)$ como dato de A que verifique:
   i) $x \in B \Leftrightarrow f(x) \in A$. Es decir:
      a) Si $x \in B \Rightarrow f(x) \in A$.
      b) Si $f(x) \in A \Rightarrow x \in B$ (o equivalentemente: Si $x \notin B \Rightarrow f(x) \notin A$).
   ii) f calculable en tiempo polinómico en la longitud de la entrada.

C. Problemas NP-completos: HAM:

Entrada: $G = (V, A)$ un grafo no dirigido de n vértices.

Salida: ¿Tiene G un camino hamiltoniano? (Es decir, ¿tiene G un camino sin repeticiones que pasa por todas los vértices de G?)

Codificación de la entrada: Con la matriz de adyacencia.

CLIQUE:

Entrada: $G = (V, A)$ un grafo no dirigido de n vértices, $k \in \mathbb{N}$ con $k \leq n$.

Salida: ¿Existe un clique de G con k vértices? Es decir, ¿existe un conjunto de k vértices U de forma que

$$\forall u, v \in U,\ u \neq v\ , \{u, v\} \in A?$$

Codificación de la entrada: Con la matriz de adyacencia y después una coma seguida de la codificación de k en binario.

PARTICION:

Entrada: $n \in \mathbb{N}$ el número de objetos,
$p_1, \ldots, p_n \in \mathbb{N}$ el peso de los objetos.

Salida: ¿Existe un conjunto de objetos $A \subseteq \{1 \ldots n\}$ que cumpla

$$\sum_{i \in A} p_i = \sum_{i \notin A} p_i = \frac{\sum_{i \in \{1, \ldots n\}} p_i}{2}?$$

Codificación de las entradas: Sobre el alfabeto $\Sigma = \{0, 1, ,\}$,los números naturales que componen una entrada se escriben en binario y se separan por comas.

SAT:

Entrada: Un conjunto de variables X y una fórmula CNF sobre X, L (que cumplen que todas las variables de X aparecen al menos una vez en los literales de L, no hay cláusulas repetidas en L y dentro de cada cláusula no hay literales repetidos).

Salida: ¿Existe una asignación de verdad que satisface L?

Codificación de la entrada: Sobre el alfabeto $\Sigma = \{0, 1, \neg, ,, (,), \vee, \wedge\}$ codificamos una entrada X, L como el número de variables n en binario, seguido de la fórmula escribiendo los números de variable en binario y los símbolos de las operaciones lógicas necesarios.