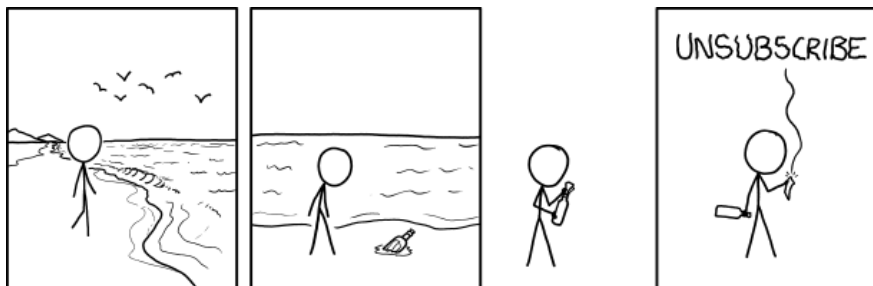


Certamen Recuperativo

Introducción a la Informática Teórica

Informática Teórica

12 de julio de 2016



1. En clase discutimos autómatas finitos, autómatas de stack y máquinas de Turing. Describa cómo podemos simular cada uno de ellos mediante un autómata que le sigue en la lista, haciendo referencia a teoremas de equivalencia según necesite.
(20 puntos)
2. Considere el lenguaje $L = \{\omega\omega\omega : \omega \in \{0,1\}^*\}$. Determine si es regular, de contexto libre, o ninguna de las anteriores.
(25 puntos)
3. Explique informalmente cómo puede construirse una máquina de Turing que acepte el lenguaje generado por una gramática de tipo 0 dada.
(20 puntos)
4. Nuestra definición es que un problema es NP-completo si puede reducirse polinomialmente al problema de determinar si una máquina de Turing no determinista acepta ω en un número de pasos acotado por un polinomio en $|\omega|$. Una definición alternativa es que hay un *certificado* c para ω , tal que una máquina de Turing determinista pueda verificar en un número de pasos acotado por un polinomio en $|\omega|$ que ω es aceptado conociendo ω y c . Dé argumentos (informales) que justifiquen la equivalencia entre estas dos definiciones.
(30 puntos)
5. Se define la clase de complejidad PSPACE como el conjunto de problemas que una máquina de Turing puede resolver en espacio polinomial en el largo de los datos de entrada. Demuestre que $NP \subseteq PSPACE$.
(25 puntos)