Notas de la clase de computación cuántica

Jonathan Andrés Niño Cortés

13 de febrero de 2015

Una máquina de Turing M trabaja en timepo $T(n)$ si para cada entrada de longitud n, se necesitan a lo más $T(n)$ pasos para llevar a cabo el cálculo.

Una función $f(n)$ es crecimiento polinomial si $f(n) \leq cn^{\alpha}$ para alguna constante $c > 0$ y $\alpha \in \mathbb{N}$. $n >> 0$.

Definición: Una función $f : \mathbb{B}^* \to \mathbb{B}^*$ es calculable en tiempo polinomial

Si $\exists$ MT que calcula F en tiempo $T(n) = Poly(n)$.

$$P = \{F : \mathbb{B}^* \to \mathbb{B}^* | F \text{ es calculable en tiempo polinomial}\} \tag{1}$$

Definición: Se dice que MT trabaja en espacio $S(n)$ si visita a lo más $S(n)$ çélulas"de la cinta de la máquina

$$PSpace = \{F : \mathbb{B}^* \to \mathbb{B}^* | F \text{ es calculable en espacio polinomial}\} \tag{2}$$

Conjetura: $P \subset PSpace$.

Un predicado es una función $P : \mathbb{B}^* \to \mathbb{B}$.

Ej: Una función que me indique si los números son primos.

$D : \mathbb{B}^* \to \mathbb{B}$ tal que $D(n)$ es 0 si n es primo y 1 si n no es primo.

Un predicado en dos variables $R : \mathbb{B}^* \times \mathbb{B}^* \to \mathbb{B}$ es calculable en timepo polinomial si existe MT que calcule R en tiempo Poly($|x|, |y|$), es decir, la máquina de Turing usa a lo más Poly($|x| + |y$) tiempo.

Los problemas NP son tales que uno puede dar una prueba que justifique el resultado dado por las funciones NP.

Def: Un predicado $L : \mathbb{B}^* \to \mathbb{B}$ esta en la clase NP si se puede representar como

$$L(x) = \exists y(|y| < q(|x|) \wedge R(x, y)) \tag{3}$$

donde q es un polinomio y R es un predicado en dos variables calculable en tiempo polinomial.

Al menos una respuesta se puede dar por certificado.

Tenemos que $P \subseteq NP$. Pero áun no sabemos si el converso es cierto o falso.

Computación Cuántica. La idea de la computación es encontrar un algoritmo que calcule una función.

Podemos suponer que una función $F : \mathbb{B}* \to \mathbb{B}^*$ la podemos partir en partes más pequeñas.

$$n \in \mathbb{N}F_n : \mathbb{B}^n \to \mathbb{B}^{f(n)}. \tag{4}$$

Obsérvese que el número de funciones $B \to B$ es 2^{2^n} así que la complejidad de encontrar estas funciones es muy grande.

Vamos a probar que los conectores lógicos $\wedge, \vee, \neg$. pueden realizar cualquier función calculable.

$$L = \{a \in \mathbb{B}^n | f(a) = 1\} \tag{5}$$

$$f_{(x)}^{(a)} = 1 \text{ si } x = a \ 0 \text{ si } x \neq a \tag{6}$$

$$f(x) = \bigvee_{a \in L} \chi^a(x) \tag{7}$$

$$\chi^{(a)} : \mathbb{B}^s \to \mathbb{B}, a = 11111 \tag{8}$$

$$\chi^{(a)}(x_1, x_2, x_3, x_4, x_5) = x_1 \wedge x_1 \wedge x_2 \wedge x_3 \wedge x_4 \wedge x_5 \tag{9}$$

$$\chi^{(b)}(x_1, x_2, x_3, x_4, x_5) = \neg x_1 \wedge x_1 \wedge x_2 \wedge x_3 \wedge x_4 \wedge x_5 \tag{10}$$

Teorema: Toda función booleana se puede construir usando $\wedge, \vee, \neg$.

Def: Un circuito booleano con base $\{f_1, \cdots, f_k\}(f_i : \mathbb{B}^{n_i} \to \mathbb{B}^{m_i})$ es una sucesión finita de funciones de la forma

$$l_n = y_1^n \times y_2^n \times \cdots \times y_{s_n}^n \tag{11}$$

donde los y_i^n son elementos de B o son id_B o si $n = 1$ son funciones constantes $1 : \mathbb{B} \to \mathbb{B}$, $0 : \mathbb{B} \to \mathbb{B}$. que además se pueden componer entre si.