THE INTRACTABILITY OF QUANTUM COMPUTATION

DANIEL CORDERO GRAU

In this paper prove the Space Time Computational Intractability,
the Fundamental Theorem of Computational Complexity, proving the Quantum Computation Intractability.

Lemma *The computational complexity class* P *is a proper subclass of the computational complexity class* NP.

Proof. Let $\{p_i\}_{i\in\mathbb{N}}$ be a convergent sequence of deterministic polynomially time bounded computable functions $p_i \in \mathrm{P}$ with limit $p = \lim\limits_{i\to\infty} p_i$, and with every p_i time bounded by the polynomial function $f_i(n) = \sum\limits_{k=0}^{i} \frac{(n \ln 2)^k}{k!}$. Then p is time bounded by the exponential function $f(n) = \limsup\limits_{i\to\infty} f_i(n) = 2^n$, so $f \in O(2^n)$, hence $p \in \mathrm{NP}$ is NP-complete. Now f is the sum of polynomials in n of each degree so it is not time bounded by any polynomial function $g(n) \in \mathbb{N}[n]$, then $f \notin O(n^k)$ for every $k \in \mathbb{N}$, hence $p \notin \mathrm{P}$. Therefore p is an NP-complete computable function, not in P, therefore, since $\mathrm{P} \subset \mathrm{NP}$, $\mathrm{P} \subsetneq \mathrm{NP}$. ∎

As a corollary, the lemma proves that $\mathrm{P} \neq \mathrm{NP}$. In general, we have

Theorem *Every computational complexity class* $\mathcal{C}$ *is a proper subclass of its closure* $\overline{\mathcal{C}}$.

Proof. Let $\mathcal{C}$ be either such that $\mathcal{C} = \bigcup\limits_{k=1}^{\infty} \mathrm{DTIME}\left[f(n)^k\right]$ or $\mathcal{C} = \bigcup\limits_{k=1}^{\infty} \mathrm{DSPACE}\left[f(n)^k\right]$, that is, the computational complexity class of the $\mathcal{C}$-complete computable functions with computational time or space complexity $T(n) = O(f(n)^k)$ for some function $f\colon \mathbb{N} \to \mathbb{N}$ and for any nonzero $k \in \mathbb{N}$. Every computational complexity class $\mathcal{C}$ can be so defined for if $\mathcal{C}$ is the computational complexity class of the $\mathcal{C}-$complete computable functions computationally bounded by a function $h\colon \mathbb{N} \to \mathbb{N}$, then $\mathcal{C} = \bigcup\limits_{k=1}^{\infty} \mathrm{DTIME}\Big[(\ln h(n))^k\Big]$ or $\mathcal{C} = \bigcup\limits_{k=1}^{\infty} \mathrm{DSPACE}\Big[(\ln h(n))^k\Big]$. Let $\{p_i\}_{i\in\mathbb{N}}$ be a convergent sequence of computable functions $p_i \in \mathcal{C}$ with limit $p = \lim\limits_{i\to\infty} p_i$ and with every p_i computationally bounded by the function $f_i(n) = \sum\limits_{k=0}^{i} \frac{(f(n)\ln 2)^k}{k!}$. Then p is computationally bounded by the function $g(n) = \limsup\limits_{i\to\infty} f_i(n) = 2^{f(n)} \in O(2^{f(n)})$, so the computational complexity $T(n)$ of p is either $T(n) = O(2^{f(n)})$, hence $p \in \overline{\mathcal{C}}$ is $\overline{\mathcal{C}}$-complete for, by polynomial-time reduction and log-space reduction, $\overline{\mathcal{C}} = \bigcup\limits_{k=1}^{\infty} \mathrm{DTIME}\Big[2^{f(n^k)}\Big]$ is the computational time complexity class of the $\overline{\mathcal{C}}$-complete computable functions with computational time complexity $T(n) = O(2^{f(n^k)})$ for any nonzero $k \in \mathbb{N}$, or $\overline{\mathcal{C}} = \mathrm{DSPACE}\left[2^{f(n)}\right] = \bigcup\limits_{k=1}^{\infty} \mathrm{DSPACE}\Big[2^{f(2^{\log^k n})}\Big]$ is the computational space complexity class of the $\overline{\mathcal{C}}$-complete computable functions with computational space complexity $T(n) = O(2^{f(n)})$. Since g is the sum of polynomials in $f(n)$ of each degree, it is not bounded by any polynomial function $h \in \mathbb{N}[f(n)]$, so $g \notin O(f(n)^k)$ for every nonzero $k \in \mathbb{N}$, hence $p \notin \mathcal{C}$. Therefore p is a $\overline{\mathcal{C}}$-complete computable function, not in $\mathcal{C}$, therefore, since $\mathcal{C} \subset \overline{\mathcal{C}}$, $\mathcal{C} \subsetneq \overline{\mathcal{C}}$. ∎

Thus, by definition of computation, the Fundamental Theorem of Computational Complexity proves the space time computational intractability. In particular,

$$\text{DSPACE} \subsetneq \text{DTIME} \subsetneq \text{NSPACE} \subsetneq \text{NTIME} = \text{EXPTIME} \subsetneq \text{EXPSPACE}\,[\text{EXPSPACE}]$$

In Descriptive Computational Complexity

In Descriptive Computational Complexity, the Fundamental Theorem of Computational Complexity proves that

$$\text{FIRST-ORDER LOGIC} \subsetneq \text{SECOND-ORDER EXISTENTIAL LOGIC} \subsetneq \text{SECOND-ORDER LOGIC}$$

In Categorical Computational Complexity

In Categorical Computational Complexity, the Fundamental Theorem of Computational Complexity is logically equivalent to the Zariski topology axioms in the theory of Computational Complexity, the Topology of Computational Complexity, the Universal Algebra of Computational Complexity Topological Spaces with the polynomial-time and the log-space maps as sheaves.

In Quantum Computation

In Quantum Computation, and for any other future form of computation, the theorem proves that the quantum tractability computational complexity class is either the nondeterministic polynomially time bounded computational complexity class NP = EXPTIME, for which the computational time complexity of the NP-complete computable functions is $T(n) = O(2^{n^k})$ for any nonzero $k \in \mathbb{N}$, if the spectrum of physically realizable quantum computers is only discrete, or the continuous nondeterministic polynomially time bounded computational complexity class CNP = EXPTIME [EXPTIME], for which the computational time complexity of the CNP-complete computable functions is $T(n) = O(2^{2^{n^k}})$ for any nonzero $k \in \mathbb{N}$, if the spectrum of physically realizable quantum computers can also be continuous.