Complexity Hierarchies

Intractable problems are solvable in principle, but in reality they require so much time or space that there no physical computers that can solve them in reasonable time. We would like to define a clear hierarchy of these set of problems.

Space Hierarchies

Def: Space constructible

We say that a function $f : \mathbb{N} \to \mathbb{N}$ such that $f(n) \geq O(\log n)$ is space constructible if there exists a function from $1^n \to \langle f(n) \rangle$ is $O(f(n))$ space complexity.

Intuitively, it is just a function that is computable in a specifically limited space. But at the moment I don't know why is this important.

Space hierarchy theorem

For any space constructible function $f : \mathbb{N} \to \mathbb{N}$ there exists a language A decidable in $O(f(n))$ but not in $o(f(n))$ space.

Proof of the theorem The proof is quite convoluted, and has some technical details that are quite ad-hoc to make things work. I personally think this is quite useless, and the hypothesis is quite intuitive with standard understanding. Nonetheless, we should need to understand the argument.

Define this algorithm: D = "On input w: 1. Let n be the length of w. 2. Compute $f(n)$ using space constructibility and mark off this much tape. If later stages ever attempt to use more, reject . 3. If w is not of the form $\langle M \rangle 10^*$ for some TM M , reject . 4. Simulate M on w while counting the number of steps used in the simulation. If the count ever exceeds $2^{f(n)}$, reject . 5. If M accepts, reject . If M rejects, accept ."

Then the language defined as $\{\omega : D \text{ accepts } \omega\}$ is the language we want in the statement.

Main consequence of Hierarchy theorem For all k, j such that $k < j$ we have that

$$SPACE(n^k) \subset SPACE(n^j)$$

NOTE: contained, and **not equal**!

More general way to say it: $f(x) = o(g(x))$ implies

$$SPACE(f(x)) \subset SPACE(g(x))$$

Other: PSPACE is in EXPSPACE

Why is this concept useful? This allows us to categorize problems is different and *not equal* complexity classes. In my own opinion this is probably caused by human need to make order in this meaningless caos. I don't know effective practical applications of this theory. But we need to be grounded on real problems, those are the ones driving innovation.

Time Hierarchies

Def: Time constructible

This is the same as #Def Space constructible.

We say that a function $f : \mathbb{N} \to \mathbb{N}$ such that $f(n) \geq O(n \log n)$ is time constructible if there exists a function from $1^n \to \langle f(n) \rangle$ is $O(f(n))$ Time complexity.

Time hierarchy theorem

This theorem is analogous to the space complexity counter-part, with a small formality of simulating the given Turing machine.

For any time constructible function $f : \mathbb{N} \to \mathbb{N}$ there exists a language A decidable in $O(f(n))$ but not in $o\left(\frac{f(n)}{\log f(n)}\right)$ time.