Computational Complexity
Mateusz Biegański

Homework 1

A single-tape Turing machine M is called predictable if its head moves in the same way for every input word (formally: if there exists a function pos : N N such that after k steps of every run of M, regardless of the input word, the head is over the tape cell number pos(k)). Is it the case that for every nondeterministic Turing machine M there exists a predictable nondeterministic Turing machine M' recognizing the same language as M and working at most polynomially slower?

Solution

TODO TODO

We know $n, m, k \in \mathbb{Z}$. The integers are closed under addition, multiplication, and when integers are squared. So, we know $n^2k + nmk + m^2k$ is an integer under closure. Thus, $\exists j \in \mathbb{Z} : n^2k + nmk + m^2k = j$.

Now, we can say $n^3 - m^3 = 2j$ by substitution. This is the definition of an even integer. Therefore, $n^3 - m^3$ is even.

QED

A function $f\colon \Sigma* \to \Gamma*$ is called a morphism if f (w · v) = f (w)· f (v) for all words $w, v \in \Sigma*$ (the symbol "·" denotes concatenation of words). A morphism f is nonabbreviating if $|f(w)| \geq |w|$ for all $w \in \Sigma*$ (i.e., f does not decrease the length of words). For a set of words $L \in \Sigma*$ we define $f(L) = \{f(w)|w \in L\}$. We say that a class C is closed under images of nonabbreviating morphisms if for every L in C , and for every nonabbreviating morphism f , also $f(L)$ belongs to C. Prove that the complexity class P is closed under images of nonabbreviating morphisms if and only if P = NP.

Solution

1. **P = NP ⇒ P is closed**

   Let's consider any language $L \in P$ and any nonabbreviating morphism f. If we show, that there exists non-deterministic, polynomial time Turing machine that recognizes language $f(L)$, then based on assumption that P=NP we will show implication.

   Let's observe, that by definition, if f is morphism, then $f(ab) = f(a) \cdot (b)$, where $a, b \in \Sigma^1$ (each letter is transformed independently of context). For all $g\colon \Sigma* \to \Gamma*$

let's define $g' \colon \Sigma \to \Gamma*$ as a finite-domain function upon single letters from finite alphabet ($g'(a) = g(a)$).

Let's define a Turing machine running over word $f(w)$, that is guessing proper letter-split-position (it is non-deterministic, because f-images of each letter may overlap). After guessing split position, it checks, whether a infix between last and actual splitting positions belongs to f' image ad if it's the case, it writes down on a separate tape a letter that is inverse image of that infix, if it isn't, it jut finishes run with fail. Reaching end of word we are having set of proper input word inverse images, we can now easily check for each whether it belongs to language L we started with (there exists proper P-time machine).

2. **P is closed $\Rightarrow$ P = NP**

   Let's observe, that we can prove it, if we find L $\in$ P and morphism f, such that f(L) is NP-complete. Let's consider problem of logical formula evaluation and it's word-representation $formula\#values\_to\_check$. This problem belongs to P class (simply substitute given values). We would like to find such f, that it transforms evaluation problem to SAT problem, which is NP-complete. It's rather simple, we just abandon values, while transforming formula representation 1:1. To make it nonabbreviating, we substitute values by #'s, and obtain word $formula\#\#\#\#....$ There exists NP-machine recognising transformed language, as SAT problem (returns true iff formula is satisfable). Thus we prooved implication.