Mateusz Biegański
mb385162

Computational Complexity - Homework 2b

Question: Consider words of the form $w_1 w_2 ... w_{2m}$, where all w_i are words of length m over the alphabet $\{0, 1\}$. Let $Perm$ be the set of those words of this form in which the words w_i are pairwise different (i.e., these are permutations of all m-bit numbers). Prove that $Perm$ belongs to log-space uniform $AC0$.

Solution

Let n be the input size ($n = m \cdot 2^m$). I construct circuit described below, that recognizes language $Perm$.

First, our TM ensures us that length of input word 1^n equals $k \cdot 2^k$ for any $k \in N$ and returns that k. It can be done in $log(n)$ space (in appendix I add algorithm for similar computation (case of $n = 2^k$). Having k (that equals to m from task description), we construct circuit as presented below:

Let S_n be the set of all words of length n over alphabet $\{0, 1\}$.

For word w let's define as C_w circuit, that for input v, such that $|v| = |w|$ returns 1 if and only if $w = v$. For each position i we generate NOT, if $w[i] = 0$ and do nothing if there was 1. Both NOT outputs and pure inputs we connect via AND gate. Depth of such circuit is $O(1)$, and construction TM uses logarithmic space (it needs to remember indices in binary representation).

For each input word w of length $k \cdot 2^k$ let's define I_w as the set of all well-formed infixes. Well-formed means here, that I_w contains only infixes of length k, starting at positions $\{0, k, 2k, ..., 2^k - k\}$ (simply, numbers that maybe are permutations).

Now, let's say we generate circuit for word w, $|w| = m \cdot 2^m$. For each well-formed infix from I_w we generate circuit C_v for every word v from $S_{|w|}$, and we connect both them. This way we obtain $2^m \cdot 2^m = O(n^2)$ results.

Word $s, |s| = m$ is part of* input w iff. any of circuits C_s returns 1 (there is 2^m such circuits). Our task is to check, whether all words from S_n are part of w. Thus, for each word $s, s \in S_n$ we add an OR such that it is connected to outputs of all C_s circuits. It returns 1 iff. s is part of w. Now, it is enough to put one AND gate at the top, for $\forall s \in S_n$ symbol, connected to each OR.

* being part of w I define as belonging to I_w.

Belonging to uniform AC0

Presented circuit belongs to AC0 in obvious way. Depth is constant and size is polynomial. Now, let's consider TM that given 1^n input generates that circuit.

The only thing we need to remember is two counters, one for number actual part of input word, and number of actual word from S_m. Both them require $log_2(2^m) = O(n)$ space, thus our TM is log-space.

It's worth remarking, that composition of log-space computations is still log-space, thus end TM is log-space.

Appendix:

TM that for input of length 2^k computes k:

Keeps counter k, equals to 0 at the beginning and counter $i = 1$. Each iteration do the following: Starting from cell 0, go $2 * i - 1$ cells right. Increase k by 1. If head is over 1, and on it's right there is *blank*, then return k. If head is over blank, throw error (input word's length is not in 2^k form for any k). If head is over 1, go to next iteration.