Comment on “Uniform Derandomization from Pathetic Lower Bounds”

Eric Allender∗
Department of Computer Science
Rutgers University
New Brunswick, NJ 08855, USA
allender@cs.rutgers.edu

V Arvind
The Institute of Mathematical Sciences
C.I.T. Campus
Chennai 600 113, India
arvind@imsc.res.in

Rahul Santhanam
School of Informatics
University of Edinburgh
Edinburgh EH8 9AD, UK
rsanthan@inf.ed.ac.uk

Fengming Wang†
Google, Inc.
76 9th Ave.
New York, NY, 10011 USA
wfengm@gmail.com

December 18, 2012

1 Permutation Problems Complete for L

In Definition 18 of our ECCC paper [AASW10] (which corresponds to Definition 3.5 of the journal version of this work [AASW12]), we define the language PWP and state that it was shown to be complete by Cook and McKenzie [MC87]. We thank Eric Miles and Emanuele Viola [MV12] for calling our attention to the following facts:

∗Supported in part by NSF Grants CCF-0830133, CCF-0832787, and CCF-1064785. Some of this work was performed while this author was a visiting scholar at the University of Cape Town.

†Supported in part by NSF Grants CCF-0830133, CCF-0832787 and CCF-1064785.
• The correct citation for this paper is [CM87], instead of [MC87], and
• The problem that Cook and McKenzie actually show is complete, which they call Permutation Product (PP), is not obviously equivalent to PWP.

In this comment, we provide a simple reduction, to establish our claim that PWP is, indeed, complete for \( L \). It suffices to provide a reduction from the \( L \)-complete language PP to PWP.

## 2 Reduction from PP to PWP

First, we present the problem PP, as defined in Cook-Mckenzie, which is \( L \)-complete: Given a list of permutations \( \pi_1, \pi_2, \ldots, \pi_t \in S_n \) and indices \( i, j \in [n] \), check if the product \( \prod_{k=1}^{t} \pi_k \) maps \( i \) to \( j \).

Now, for completeness, we remind the reader of the definition of the problem PWP: For permutations \( \pi_1, \pi_2, \ldots, \pi_t \in S_n \) check if their product \( \prod_{k=1}^{t} \pi_k \) is the identity.

There is a direct reduction from PP to PWP as explained below:

Firstly, we reduce the PP instance to one in which \( i = j \) by considering the list of permutations \( \pi_1, \pi_2, \ldots, \pi_t, \pi_{t+1} \), where \( \pi_{t+1} \) is the transposition \((i, j)\). Clearly, their product maps \( i \) to \( i \) iff the first \( t \) of them map \( i \) to \( j \).

Next, enlarge the domain by one element, so that we will consider permutations in \( S_{n+1} \) instead of \( S_n \): Replace each \( \pi_k \) by \( \sigma_k \in S_{n+1} \), where \( \sigma_k \) coincides with \( \pi_k \) on \([n]\) and \( \sigma_k(n + 1) = n + 1 \). Let \( \tau \in S_{n+1} \) denote the transposition \((n + 1, i)\). Let \( g \) denote the permutation \( \prod_{k=1}^{t+1} \sigma_k \).

**Claim.** \( \prod_{k=1}^{t+1} \pi_k \) maps \( i \) to \( i \) if and only if \( g \tau g^{-1} \tau \) is the identity permutation.

**Proof.**

Suppose \( \prod_{k=1}^{t+1} \pi_k \) maps \( i \) to \( i \). Then \( g(i) = i \). Since \( g(n + 1) = n + 1 \) we can see that \( g \tau g^{-1} \tau \) maps \( i \) to \( i \) and \( n + 1 \) to \( n + 1 \). As for the other points \( j \in [n + 1] \), \( \tau \) doesn’t interfere and the combination of \( g \) and \( g^{-1} \) fixes them all.

Conversely, suppose \( g \tau g^{-1} \tau \) is the identity. Then, in particular, \( g \tau g^{-1} \tau(i) = i \) which means \( g \tau g^{-1}(n + 1) = i \) which implies \( g \tau(n + 1) = i \) which implies \( g(i) = i \) which implies \( \prod_{k=1}^{t+1} \pi_k \) maps \( i \) to \( i \).

In summary, the reduction from PP to PWP is:

\[
\pi_1, \ldots, \pi_{t+1} \mapsto \pi_1, \ldots, \pi_{t+1} \tau \pi_{t+1}^{-1} \ldots \pi_1^{-1} \tau.
\]
References


