The mistake in "A Solution of the P versus NP Problem"

Norbert Blum
Institut für Informatik, Universität Bonn
Friedrich-Ebert-Allee 144, D-53113 Bonn, Germany
email: blum@cs.uni-bonn.de

October 11, 2017

Because of Tardos' function [2], the proof of Theorem 6 in [1] has to be wrong. The aim of this note is to elaborate precisely the mistake in this proof.

The first step of the approach in [1] is a transformation which eliminates the negated variables in the monomials of the DNF representation and in the clauses of the CNF representation of the functions computed at the nodes of the given standard network resulting into the so-called *reduced* DNF and CNF representations of these functions. Theorem 5 characterizes the reduced DNF and CNF representations of the function computed at the output node of the standard network. Theorem 5 is correct.

Given a standard network which computes a non-constant monotone Boolean Function at its output node and a CNF-DNF-approximator, the approximator was used for the approximation of the reduced formulas computed at the nodes of the network. I shall elaborate the mistake for the case that the reduced DNF formulas are approximated. With respect to the approximation of the reduced CNF formulas, an analogous reasoning can be given.

If the reduced DNF representations are used for the evaluation of a standard network β with respect to a given input c, Method 1 which computes for each gate the value with respect to the input c cannot be used. This can be seen as follows.

Consider an $\wedge$-gate g with direct predecessors h_1 and h_2. Assume that each monomial in the reduced DNF-representation with respect to h_1 which is fulfilled by c contains the variable x_i. Furthermore assume that each

monomial in the reduced DNF-representation with respect to h_2 which is fulfilled by c has absorbed the negated variable $\neg x_i$. Then all monomials which are fulfilled by c at the output of the gate g correspond to former trivial monomials which are replaced by 0 by the operator R. Hence, the reduced DNF-representation with respect to g contains only monomials which are not fulfilled by c. Therefore, for the evaluation of the standard network β, the reduced DNF-representation of the function computed at the output node has to be constructed.

Essential for the lower bound proof for the monotone complextiy of the given monotone Boolean function is the property that during a DNF/CNF-approximator switch, a negative test input c for which the replacement of a clause d of size k introduces an error, must have the value zero for each variable contained in d. Because of the transformation, this property is lost such that the upper bound proof for the number of negative test inputs for which a DNF/CNF-approximator switch introduces an error cannot be used here. To see this assume that x_i and x_j are variables in d. Let c be a negative test input which assigns x_i the value one and x_j the value zero. With respect to monotone Boolean networks, the reasoning is that the replacement of d by one cannot introduce an error for c since, because of $c_i = 1$, the value of d is anyway one. Because of the application of the operator R, this property cannot be applied with respect to the approximation of reduced DNF formulas. The reasoning is analogous to the reasoning why Method 1 cannot be used for the evaluation if the reduced DNF representations are used. For a successor of g, each monomial which contains x_i and which is fulfilled by c, could correspond to a former trivial monomial which is replaced by zero by the operator R. Therefore, the replacement of the clause d can introduce an error for the negative test input c although c_i assigns one to the variable x_i contained in d.

My motivation for the transformation was the avoidance of the explicit approximation of the negated variables. Now, I am convinced that such a transformation does not help for the proof of a lower bound for the standard complexity of a Boolean function. Therefore, the negated variables have been approximated as well. For doing this, some hard problems have to be solved. I am working on these problems.

References

[1] Blum, N., A Solution of the P versus NP Problem, arXiv:1708.03486v1 [cs.CC].

[2] Tardos, É., The gap between monotone and non-monotone circuit complexity is exponential, *Combinatorica* **8**, 141–142.