Section 1: Proof Theory

A straightforward proof of Köbler - Messner's result

Zenon Sadowski

Institute of Mathematics, University of Białystok, 15-267 Białystok, ul. Akademicka 2, Poland
e-mail: sadowski@math.uwb.edu.pl

A proof system for a language L is a polynomial time computable function whose range is L. In order to compare the efficiency of different proof systems, S. Cook and R. Reckhow introduced the notion of p-simulation. Intuitively a proof system h p-simulates a second one g if there is a polynomial time computable function t translating proofs in g into proofs in h. A proof system is called p-optimal for L if it p-simulates any proof system for L. The question whether a p-optimal proof system for TAUT (the set of all tautologies in Propositional Logic) exists, posed by J. Krajíček and P. Pudlák is important in the theory of propositional proof systems.

J. Köbler and J. Messner formalized the intuitive connection between the existence of p-optimal proof systems and the existence of many-one complete sets. As a corollary from their master theorem they obtained the following result. Let SAT be the set of all satisfiable boolean formulas.

Theorem (Köbler, Messner)

If TAUT and SAT have p-optimal proof systems then $NP \cap$ co-NP has a complete language.

We present another proof of this theorem.

Let $x \in \Sigma^*$ and let M be a nondeterministic polynomial time clocked Turing machine with two distinguished states the accepting state and the rejecting state. The machine M behaves well on input x if there exists a computation of M accepting x or there exists a computation of M rejecting x. The machine M is is n-consistent if and only if for any input y of length n there does not exist simultaneously a computation path of M accepting y and a computation path of M rejecting y.

By $NP \cap$ co-NP-machine we mean a Turing machine which can be obtained by combining into one machine two polynomial time clocked nondeterministic Turing machines which accept complementary languages. A polynomial time clocked nondeterministic Turing machine M is a $NP \cap$ co-NP-machine if and only if for any $x \in \Sigma^*$, M behaves well on input x and for any n, M is n-consistent.

Let us define $L \subseteq \Sigma^*$ in the following way:

$L = \{< M, x, \alpha_x^M, \beta_n^M, Proof_1, Proof_2, 0^{n^k+k} > : x \in L(M)\}$ where:

M is a polynomial time clocked nondeterministic Turing machine, $n^k + k$ is its polynomial time bound.

$n = |x|$

α_x^M is a boolean formula such that α_x^M is satisfiable if and only if M behaves well on input x.

β_n^M is a boolean formula such that β_n^M is a propositional tautology if and only if M is n-consistent.

Proof_1 is a proof of $\alpha_x{}^M$ in the p-optimal proof system for SAT.
Proof_2 is a proof of $\beta_n{}^M$ in the p-optimal proof system for TAUT.
0^{n^k+k} is the sequence of zeros (padding).

The language L is the desired NP $\cap$ co-NP complete language.

References

1. Cook, S.A., Reckhow, R.A.: The relative efficiency of propositional proof systems. J. Symbolic Logic 44 (1979) 36-50
2. Köbler, J., Messner, J.: Complete Problems for Promise Classes by Optimal Proof Systems for Test Sets. In Proceedings of the 13th Annual IEEE Conference on Computational Complexity (1998) 132-140
3. Krajíček, J., Pudlák, P.: Propositional proof systems, the consistency of first order theories and the complexity of computations. J. Symbolic Logic 54 (1989) 1063-1079