Mathematical Problems of Computer Science 38, 42–43, 2012.

On some Systems of Minimal Propositional Logic with Loop Detection Mechanisms

Hovhannes Bolibekyan

Department of Informatics and Applied Mathematics,
Yerevan State University, Armenia
bolibekhov@ysu.am

Backwards proof search and theorem proving with a standard cut-free calculus for the propositional fragment of minimal logic is insufficient because of three problems. Firstly, the proof search is not in general terminating caused by the possibility of looping. Secondly, it might generate proofs which are permutations of each other and represent the same natural deduction. Finally, during the proof some choice should be made to decide which rules to apply and where to use them.

Several proof systems of I.Johansson's minimal logic of predicates were introduced in [1]. Looping is the main issue in the propositional fragment of the system GM^- developed in [1]. Looping may easily be removed by checking if a sequent has already occured in the branch. Though this insufficient as it requires much information to be stored. Some looping mechanisms have been considered earlier in [2,3].

One way to detect loops is adding history to each sequent. The history is a set of sequents that have occured so far in the branch of a proof tree. After each backwards inference the new sequent is verified if it is in this set. If it is we have looping and backtrack. Otherwise the new history is the extension of the old history by the old sequent. This method is unsufficient as it requires much information represented as list of sequents to be stored and on each step this lists should be checked. To improve the efficiency some mechanisms are required to cut down the amount of storage and checks needed to prevent looping. In fact to reduce the history we need only store goal formulae in order to check loops. Using the rules of GM^- the context cannot decrease: once a formula is in a context it will be in the context of all the sequents above it in the proof tree. So the sequents to be the same they need to have the same context. Therefore we may empty the history every time the context is extended. Only goal formulae are needed to be stored in the history. If there is a sequent whose goal is already in the history, then we have the same goal and the same context as another sequent, so looping occured.

We introduce two systems for propositional fragment of minimal logic which are slightly different. Both systems are based on the idea of adding context to the sequents. In one system, $SwMin$, the history is kept smaller, but $ScMin$ detects loops more quickly. The heart of the difference between the two systems is that in the $SwMin$ loop checking is done when a formula leaves the goal, whereas in the $ScMin$ it is done when it becomes the goal.

Theorem

1. The systems GM^- and $SwMin$ are equivalent.
2. The systems GM^- and $ScMin$ are equivalent.

Acknowledgment. This work is supported by Grant 11-1b023 of SSC of the Goverment of Armenia.

References

[1] Bolibekyan H.R., Chubaryan A.A., On some proof systems for I.Johansson's minimal logic of predicates, Proceedings of the Logic Colloquium 2003, p. 56.

[2] Gabbay D., Algorithmic proof with diminishing resources, Part 1, Springer Lecture Notes in Computer Science, 533, pp. 156-173, 1991

[3] Bolibekyan H., Muradyan T., On some loop detection strategies for minimal propositional logic, Proceedings of the Logic Colloquium 2011, p. 45-46.