KPROP — An AND-parallel Theorem Prover for Propositional Logic implemented in KL1 System Abstract

Johann M. Ph. Schumann

September 6, 1994

KPROP is a theorem prover for first order propositional logic with *Model Elimination* [Lov78] as its underlying calculus.

The given input clauses in conjunctive normal form are fanned out into *contrapositives* (see e.g. citeSti88,LSBB89). After that, they are compiled into KL1 clauses. KL1 is the parallel logic language, based on Guarded Horn Clauses which has been developed at ICOT [Sus89].

An implementation of the Model Elimination proof procedure (i.e. the extension and reduction step) can be accomplished quite easily in that language, since each contrapositive can be compiled into one KL1 rule. With this approach, only very little code has to be added to yield the entire theorem prover.

The exploitation of AND-parallelism is done by the KL1 language itsself. As defined in the language, all subgoals of a clause (i.e. all literals which follow the guard) are tried in parallel.

No additional synchronisation has to be performed, since we are working with propositional logic where the problem of shared variables (see e.g. [KR88]) does not occur.

The performance of this theorem prover could be increased substantially by encorporating a rather powerful method for pruning the search space ("Equal Predecessor Fail")
([LSBB92], [AL91], and [Sti88]). In the Model Elimination tableau this means that no node may be dominated by a node in the same branch which is marked with the same literal. If such a situation occurs, this branch can considered to be failing, and a backtracking step can be performed. With this pruning method (see also [LSBB92, AL91, Sti88]) the calculus remains complete, but the search space becomes considerably smaller. Especially in the case of propositional calculus, this method has two additional advantages:

- To check whether two nodes of the tableau are equal can be accomplished in constant time.

- The number of different labels for the nodes in the tableau is finite, namely twice the number of variables. Therefore, the length of branches of the tableau is restricted with the given pruning method being applied. This allows an unrestricted depth first search to be performed. In contrast to this, theorem provers for predicate calculus have to perform iterative deepening.

First measurements with this theorem prover have been made on a simulation of KL1 running on a PSI-machine at ICOT. The following table shows the run-time and the number of inferences of the proof for several examples. The pigeon-hole examples are from [Pel86], the problem salt and mustard by L. Carroll is from [McC88], and $full_i$ is a formula constructed out of all permutations of i variables (with 2^i clauses) (given to me by R. Letz).

Example	Inferences	run-time [s]
pigeon3	67	0.083
pigeon4	393	0.254
pigeon5	2611	1.452
salt	1518	0.777
full2	7	0.065
full3	34	0.068
full4	197	0.141
full5	1306	0.993

References

[AL91] O. L. Astrachan and D. W. Loveland. METEORs: High Performance Theorem Provers Using Model Elimination. Technical Report CS-1991-08, Dept. of CS, Duke University, Durham, North Carolina, 1991.

[KR88] J. Chassin de Kergommeaux and P. Robert. An abstract machine to implement efficiently OR–AND parallel Prolog. Technical report, ECRC, Munich, 1988.

[Lov78] D. W. Loveland. *Automated Theorem Proving: a Logical Basis.* North–Holland, 1978.

[LSBB92] R. Letz, J. Schumann, S. Bayerl, and W. Bibel. SETHEO: A High-Performance Theorem Prover. *Journal of Automated Reasoning*, 8(2):183–212, 1992.

[McC88] W. McCune. OTTER users' guide. Technical report, Mathematics and Computer Sci. Division, Argonne National Laboratory, Argonne, Ill., USA, 1988.

[Pel86] F. J. Pelletier. Seventy–five problems for testing Automated Theorem Provers. *Journal of Automated Reasoning*, 2:191–216, 1986.

[Sti88] M. E. Stickel. A Prolog Technology Theorem Prover: Implementation by an Extended Prolog Compiler. *Journal of Automated Reasoning*, 4:353–380, 1988.

[Sus89] Kasumi Susaki. KL1 Programming. Technical Report TM-949, ICOT, Tokyo, Japan, 1989.