HOL Provers for First-order Modal Logics — Experiments*

Christoph Benzmüller

Department of Mathematics and Computer Science, Freie Universität Berlin, Germany

Abstract

Higher-order automated theorem provers have been employed to automate first-order modal logics. Extending previous work, an experiment has been carried out to evaluate their collaborative and individual performances.

1 Introduction

Higher-order automated theorem provers are well suited as reasoners for a wide range of quantified non-classical logics [1]. The key idea is to exploit classical higher-order logic (HOL) as a universal meta-logic and, for example, to explicitly encode Kripke structures within this meta-logic. Experiments have shown that this approach, which is orthogonal to explicit world labeling techniques in direct theorem provers, is indeed competitive [2]. More recent, but non-exhaustive, experiments have improved and confirmed these results [3]. This short paper significantly further extends the experiments reported in [3]. The paper thus provides useful and relevant information for evaluations of competitor systems. Moreover, some light is shed on the collective and individual performances of the higher-order automated theorem provers LEO-II [4], Satallax [6], Isabelle [9], agsyHOL [8] and Nitpick [5].

2 Experiments

A meta-prover for HOL, called HOL-P has been introduced and evaluated in [3]. This meta-prover exploits the SystemOnTPTP infrastructure [11] and sequentially schedules the HOL reasoners LEO-II, Satallax, Isabelle, agsyHOL and Nitpick running remotely at the SystemOnTPTP cluster at Miami (which provides 2.80GHz computers with 1GB memory). The HOL-P system and the HOL-P constituent provers are evaluated here with respect to the 580 benchmark problems in the QMLTP library [10]. Extending previous experiments [3], these problems are studied for 5 different logics (K, D, T, S4, S5) and for 3 different domain conditions (constant, cumulative, varying). The total sum of considered problem variants is thus 8700. These QMLTP problem variants have been converted into THF syntax with the FMLtoHOL tool [3]. Moreover, the particular configuration of HOL-P has been varied, and different system timeouts and different numbers of constituent provers have been considered. These experiments, which have been conducted over the past four months, have required a substantial amount of time and computing resources on the SystemOnTPTP cluster (I am grateful to Geoff Sutcliffe for providing these resources).[1]

*This work has been supported by the German Research Foundation (DFG) under grants BE 2501/9-1 and BE 2501/11-1.

[1]Important technical remark: QMLTP axioms have been treated as global axioms in the experiments; cf. the definition of local versus global logical consequence in [7].

	K			D			T			S4			S5		
Type	co	cu	va	co	cu	va	co	cu	va	co	cu	va	co	cu	va
THM	192	168	149	206	180	159	260	234	211	298	271	242	345	333	282
CSA	259	284	309	253	270	299	177	190	229	132	146	186	77	77	129
SAT	3	3	3	2	2	2	2	2	2	2	2	2	2	2	2
Σ	454	455	461	461	452	460	439	426	442	432	419	430	424	412	413

Table 1: Performance of HOL-P (with 600s overall timeout, 120s timeout for each constituent prover) for first-order modal logics K, D, T, S4 and S5 with constant domains (co), cumulative domains (cu) and varying domains (va).

	THM	CSA	SAT	Σ	UNK
HOL-P	3530	3017	33	6580	2120
Satallax	3167	752	0	3919	4781
Nitrox	0	3017	33	3050	5650
Isabelle	2955	0	0	2955	5745
LEO-II	2647	284	0	2931	5769
agsyHOL	2784	0	0	2784	5916

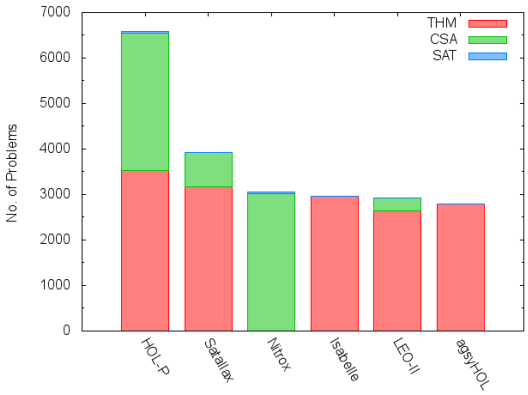

Table 2: Performance of the HOL-P constituent provers (120s timeout each; HOL-P results are wrt 600 seconds timeout).

2.1 First experiment

In this experiment HOL-P was configured to sequentially schedule the provers LEO-II—1.6.2, Satallax—2.7, Isabelle—2013, Nitrox—2013, and agsyHOL—1.0. The timeout for each prover was set to 120 seconds of CPU time. For HOL-P this adds to a total timeout of 600 seconds. The results of this experiment are reported in Table 1 (in this paper THM, CSA, SAT, and UNK stand for theorem, countersatisfiable, satisfiable, and unknown, respectively). The particular setting of the experiment thus coincides with the setting chosen in [3]. However, here HOL-P is evaluated for logics K, T and S5 in addition to logics D and S4. The particular results for the latter two logics very slightly differs from those reported in [3]. We conjecture that these differences are related to SystemOnTPTP issues, which serves as black box in our experiments. In particular, there are no means to control the very detailled execution conditions for each prover run when using this infrastructure. Future work should therefore investigate how the replication precision of experiments conducted via the SystemOnTPTP infrastructure can be further improved.

The individual performances of the HOL-P constituent provers have also been evaluated. Table 2 depicts the cumulative performance of each prover for all 8700 QMLTP problem variants. Remember that each prover was given a timeout of 120 seconds. The cumulative performance of HOL-P is also depicted; however, the comparison is unfair since the underlying timeout of HOL-P is 600 seconds. An alternative comparison is possible with the results reported for HOL-P in Table 3. There HOL-P was run with the same constituent provers but with an overall timeout of just 100 seconds; in this setting HOL-P nevertheless performed better wrt the number of

	K			D			T			S4			S5		
Type	co	cu	va	co	cu	va	co	cu	va	co	cu	va	co	cu	va
THM	186	162	141	201	175	154	252	223	205	289	261	233	345	319	270
CSA	263	275	298	233	245	268	159	180	211	128	140	179	77	74	126
SAT	3	3	3	2	2	2	2	2	2	2	2	2	2	2	2
Σ	452	440	442	436	422	424	413	405	418	419	403	414	424	395	398

Table 3: Performance of HOL-P (100s overall timeout, 20s timeout for each constituent prover) for first-order modal logics K, D, T, S4 and S5 with constant domains (co), cumulative domains (cu) and varying domains (va).

	THM	CSA	SAT	Σ	UNK
HOL-P	3408	2856	33	6297	2403
Satallax	3024	749	0	3773	4927
Nitrox	0	2856	33	2889	5811
LEO-II	2472	231	0	2703	5997
agsyHOL	2644	0	0	2644	6056
Isabelle	2354	0	0	2354	6346

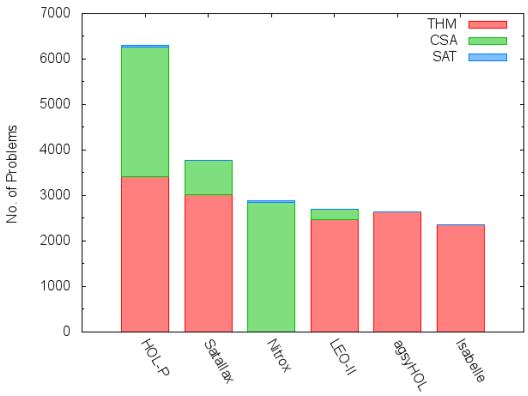

Table 4: Performance of the HOL-P constituent provers (20s timeout each; HOL-P results are wrt 100 seconds timeout).

proved theorems and the overall number of solutions found than any of the individual provers with a 120 seconds timeout. Only with respect to finding countermodels the situation differs, here Nitrox has a slight advantage over HOL-P.

2.2 Second experiment

In this experiment HOL-P was again configured to sequentially schedule the provers LEO-II—1.6.2, Satallax—2.7, Isabelle—2013, Nitrox—2013, and agsyHOL—1.0. However, the timeout for each prover was now set to 20 seconds of CPU time. For HOL-P this adds to a total timeout of 100 seconds. The results of this experiment are reported in Tables 3 and 4. Interestingly, the performance loss with regard to the first experiment is less significant as expected. Even with the short 20 second timeouts for the individual provers, HOL-P remains a competitive prover for first-order modal logics. The individual performances of the HOL-P constituent provers have slightly changed though. In particular Isabelle performs weaker with short timeouts.

2.3 Third experiment

In this experiment HOL-P was configured to sequentially schedule only Satallax—2.7 and Nitrox—2013. These two individual provers performed best in the above experiments. Moreover, they are quite complementary regarding their specialization in proving theorems and finding countermodels. The timeout for each prover was set to 50 seconds of CPU time. For HOL-P this adds

to a total timeout of 100 seconds. The results of this experiment are reported in Table 5. The theorem proving performance of HOL-P in this experiment is weaker than in the second experiment. This illustrates the complementary strength of the HOL proveres for proving theorems. However, the performance for finding countermodels has slightly improved now for HOL-P, since Nitrox, which is the only strong countermodel finder currently available for HOL, productively employs its increased reasoning time in the modified setting.

	K			D			T			S4			S5		
Type	co	cu	va	co	cu	va	co	cu	va	co	cu	va	co	cu	va
THM	162	150	132	175	161	141	225	212	190	262	246	219	305	305	258
CSA	266	280	308	251	267	298	176	190	223	132	146	186	77	77	129
SAT	3	3	3	2	2	2	2	2	2	2	2	2	2	2	2
Σ	431	433	443	428	430	441	403	404	415	396	394	407	384	384	389

Table 5: Performance of HOL-P (100s overall timeout, 50s timeout for each constituent prover) for first-order modal logics K, D, T, S4 and S5 with constant domains (co), cumulative domains (cu) and varying domains (va).

3 Summary

The collaborative and individual performances of higher-order automated theorem provers has been evaluated for first-order modal logic problems. The results demonstrate the dominance of the collaborative theorem prover HOL-P over its individual constituent provers. The strongest individual provers in our experiments were Satallax and Nitrox. As our experiments show, Satallax is not subsuming the other provers: assigning more reasoning time to Satallax is less effective than operating with different constituent provers in HOL-P and sharing the available resources. An optimal configuration of HOL-P has not yet been identified and further experiments could try find such a configuration. However, given that the individual HOL provers are subject to frequent revisions and improvements, this optimization problem appears to be a non-trivial, moving target.

References

[1] C. Benzmüller. A top-down approach to combining logics. In *Proc. of the 5th International Conference on Agents and Artificial Intelligence (ICAART)*. SciTePress Digital Library, 2013.

[2] C. Benzmüller, J. Otten, and T. Raths. Implementing and evaluating provers for first-order modal logics. In *Proc. of ECAI 2012*, Montpellier, France, 2012.

[3] C. Benzmüller and Th. Raths. Hol based first-order modal logic provers. In *Proc. of LPAR*, volume 8312 of *LNCS*, pages 127–136. Springer, 2013.

[4] C. Benzmüller, F. Theiss, L. Paulson, and A. Fietzke. LEO-II - a cooperative automatic theorem prover for higher-order logic. In *Proc. of IJCAR 2008*, volume 5195 of *LNCS*, pages 162–170. Springer, 2008.

[5] J.C. Blanchette and T. Nipkow. Nitpick: A counterexample generator for higher-order logic based on a relational model finder. In *Proc. of ITP 2010*, volume 6172 of *LNCS*, pages 131–146. Springer, 2010.

[6] C.E. Brown. Satallax: An automated higher-order prover. In *Proc. of IJCAR 2012*, volume 7364 of *LNCS*, pages 111 – 117. Springer, 2012.

[7] M. Fitting and R.L. Mendelsohn. *First-Order Modal Logic*. Kluwer, 1998.

[8] F. Lindblad. agsyHol:. `https://github.com/frelindb/agsyHOL`, 2012.

[9] T. Nipkow, L.C. Paulson, and M. Wenzel. *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*. Number 2283 in LNCS. Springer, 2002.

[10] T. Raths and J. Otten. The QMLTP problem library for first-order modal logics. In *Proc. of IJCAR 2012*, volume 7364 of *LNCS*, pages 454–461. Springer, 2012.

[11] G. Sutcliffe. The TPTP problem library and associated infrastructure. *Journal of Automated Reasoning*, 43(4):337–362, 2009.