Building a Problem Library for First-Order Modal Logics

Thomas Raths Jens Otten

Institut für Informatik, University of Potsdam
August-Bebel-Str. 89, 14482 Potsdam-Babelsberg, Germany
{traths,jeotten}@cs.uni-potsdam.de

Abstract. First-order modal logics have many applications, e.g. in planning and program verification. Whereas comprehensive and standardized problem libraries exist for, e.g., classical (TPTP library) and intuitionistic (ILTP library) logic, nothing comparable is so far available for first-order modal logics. The aim of the Quantified Modal Logic Theorem Proving (QMLTP) library is to close this gap by providing a comprehensive set of problems for various first-order modal logics. We present a preliminary version of this library, which includes 200 problems represented in an extended TPTP syntax. The main purpose of the QMLTP library is to put the testing and evaluation of automated theorem proving (ATP) systems for first-order modal logic on a firm basis, make meaningful system evaluations and comparison possible, and to measure practical progress in ATP for modal logics. We would like to invite all interested users to submit problems and ATP systems for first-order modal logics.

1 Introduction

Modal logics extend classical logic with the modalities "it is necessarily true that" and "it is possibly true that" represented by the unary operators $\Box$ and $\Diamond$, respectively. The (Kripke) semantics of modal logics is defined by a set of worlds constituting classical logic interpretations, and a binary accessibility relation on this set. Then $\Box F$ or $\Diamond F$ is true in a world w, if F is true in *all* worlds accessible from w or *some* world accessible from w, respectively. First-order or *quantified* modal logics (QMLs) extend propositional modal logics by *domains* specifying sets of objects that are associated with each world, and the standard universal and existential quantifiers [4, 13].

Quantified modal logics allow a natural and compact knowledge representation. The subtle combination of the modal operators and first-order logic enables specifications on epistemic, dynamic and temporal aspects, and on infinite sets of objects. For this reason, first-order modal logics have many applications, e.g. in planning, natural language processing, program verification, querying knowledge bases, and modeling communication. In these applications, modalities are used to represent incomplete knowledge, programs, or to contrast different sources of information.

For example, the planning system PKS [20] constructs conditional plans. It uses modal operators to represent incomplete knowledge, constant and predicates to describe objects and their properties, and variables and functions to generate abstract plans, which are instantiated later, when sufficient knowledge is available. An inference procedure for a restricted quantified modal logic determines, whether the plan achieves the goal and the preconditions of the actions hold, and generates the effects of the actions. PKS can be applied to, e.g., dialogue planning [27]. The dialogue system Artimis [24] and the sentence-planner SPUD [29] plan, generate and interpret sentences in a natural language. They use modalities to distinguish beliefs, intentions, and actions of the system and the user. First-order logic components represent objects, properties and quantified statements. Variables enable to process abstract instructions that can be instantiated later, when more information is available [28]. An inference engine is adopted to plan and interpret the sentences. The systems KIV [22], VSE-II [1] and KeY [3] are advanced tools for program verification and synthesis. Their proof components use dynamic and temporal first-order logic which are closely related to first-order modal logic. The modalities represent programs, whereas functions, variables and quantifiers characterize attributes, types, and the creation of objects. Likewise the verification of database update programs [26] and the integration of UML specification [5] can be described in first-order modal logic. A first-order modal logic is also used as query language for description logic knowledge bases [6]. Automated reasoning is required to answer queries and to verify and optimize integrity conditions. Finally, first-order modal logic is used to describe communication and cooperation [8, 16]. All these applications would benefit from a higher degree of automation. Consequently there is a real need for efficient automated theorem proving (ATP) systems for first-order modal logic.

Testing ATP systems using standardized problem sets is a well-established method for measuring their performance. E.g., for classical logic the TPTP library [30], and for intuitionistic logic the ILTP library [21] were developed. These libraries have stimulated the development of more efficient ATP systems for these logics. There already exist benchmark problems and methods for some *propositional* modal logics, e.g. there are some scalable problem classes [2] and procedures that generate formulas randomly in a normal form [19]. For *first-order* modal logics, there are only very small collections of formulas available, e.g. formulas used for testing the ATP system GQML-Prover [31].

The aim of the *Quantified Modal Logic Theorem Proving (QMLTP) library* is to provide a comprehensive set of problems for various first-order modal logics. This will put the testing and evaluation of ATP systems for quantified modal logic on a firm basis, make meaningful system evaluations and comparison possible, and will allow to measure practical progress in the field of automated theorem proving for first-order modal logic.

This paper introduces the first version of the QMLTP library. The requirements of the library are described, as well as its contents, and the suggested syntax for representing first-order modal formulas. A call for problems and ATP systems for first-order modal logic is included in the appendix.

2 A Problem Library for First-Order Modal Logic

In the following the objectives, requirements, and contents of the QMLTP library are described, before the syntax and presentation of formulas for first-order or quantified modal logic (QML) are specified.

2.1 Objectives and Requirements

The main objectives of the QMLTP problem library are as follows:

- to put the evaluation of ATP systems for QML onto a firm basis,
- to stimulate the development of novel, more efficient calculi for QML,
- to help finding out if an ATP system for QML is incorrect,
- to measure the progress in ATP for QML, and
- to provide information about available ATP systems for QML.

In order to achieve these objectives the QMLTP library needs to satisfy the following requirements, which are already used for the TPTP library [30] and the ILTP library [21].

1. It is easy to obtain and provides guidelines for its use in evaluating ATP systems.
2. It is well structured, documented, and provides statistics about the library as a whole.
3. It is easy to use; the problems are provided in an easy to understand format and conversion tools to other known syntax formats are included.
4. It is large enough for statistically significant testing; the long term goal is to include at least 1000 problems.
5. It contains problems of varying difficulty, i.e. it includes simple as well as unsolved/open problems.
6. It contains problems from diverse domains, e.g. problems from areas such as planning, natural language processing, and program verification.
7. It assigns each problem a unique name and provides status and difficulty rating for each problem.

All these requirements, except for number four, are already met by the first version of the QMLTP library described in the following. Requirement number four will be fulfilled in subsequent more comprehensive versions of the QMLTP library.

2.2 Obtaining and Using the Library

The QMLTP library is available at `http://www.iltp.de/qmltp`. When presenting results based on the QMLTP library the release number of the library and of the tested ATP system (including all settings) has to be stated. No part of the problems may be modified, e.g. by reordering of axioms, and the header information of the problems may not be exploited; each problem should be referred to by its unique name. It is a good practice to make the executable (or binary) of the tested ATP system available, in order to allow the verification of the published performance data.

2.3 Contents

The first release v0.2 of the QMLTP library includes 200 problems: 190 first-order (i.e. non-propositional) and 10 propositional problems. These 200 problems are divided into seven problem domains. These domains are APM (applications mixed), GAL/GLC/GSE/GSV/GSY (Gödel's embedding for algebra, logic calculi, set theory, software verification, and syntactic problems, respectively), and SYM (syntactic modal):

1. APM – *applications mixed*.
   10 problems from planning, natural language processing and communication, querying databases, and software verification [25, 9, 5, 6, 23, 28].

2. GAL/GLC/GSE/GSV/GSY – *Gödel's embedding*.
   130 problems are generated by using Gödel's embedding of intuitionistic logic into the modal logic S4 [15]. The original problems were taken from the TPTP library [30] and derived from problems in the domains ALG (general algebra), LCL (logic calculi), SET (set theory), SWV (software verification), and SYN (syntactic), respectively.

3. SYM – *syntactic modal*.
   60 problems from different sources on calculi, procedures and ATP systems for first-order modal logic [11, 12, 14, 10, 7, 31, 17]. These problems have no obvious semantic interpretation.

As already done in the TPTP and ILTP library, each problem is assigned a status and a rating. The *rating* determines the difficulty of a problem with respect to current state-of-the-art ATP systems. It is the ratio of state-of-the-art ATP systems which are *not* able to solve a problem within a given time limit. For example a rating of 0.3 indicates that 30% of the state-of-the-art systems do *not* solve the problem; a problem with rating of 1.0 cannot be solved by any state-of-the-art system. A state-of-the-art system is an ATP system whose set of solved problems is not subsumed by any another ATP system.

Each problem is assigned a modal *status*. This status is either `Theorem`, `Non-Theorem`, `Unsolved` or `Open`. Problems with `Unsolved` status have not been solved by any state-of-the-art ATP system, but it is known whether they are theorems or not; for problems with `Open` status it is unknown if they are theorems or not, i.e. the abstract problem has not been solved so far. The status is specified with respect to a particular modal logic, e.g. S4, K, K4, D, D4, T, and a particular domain (condition), i.e. constant, cumulative, or varying domains.[1] For the first version of the QMLTP library all rating and status information is with respect to the modal logic S4 with cumulative domains. Future versions of the library will consider other modal logics and domains as well.

[1] Depending on whether the domains for the (Kripke) worlds stay constant, increase or vary arbitrarily when moving from a world to other accessible worlds we speak of constant, cumulative, and varying domains [13].

To determine rating and status of the problems in the first version of the QMLTP library, the ATP systems GQML-Prover v1.2 and MleanTAP v1.0 were used.[2] GQML-Prover [31] is based on a free-variable tableau calculus using annotated tableau nodes and function symbols; it is implemented in OCaml. MleanTAP implements an analytic free-variable tableau using prefixes and a prefix unification algorithm; it is written in Prolog and is an adaption of the ileanTAP [18] system for intuitionistic logic.[3]

2.4 Naming, Syntax and Presentation

Similar to the TPTP library, each problem is given an unambiguous name. The problem *name* has the form `DDD.NNN+V[.SSS].p` consisting of the mnemonic `DDD` of its domain, the number `NNN` of the problem, its version number `V`, and an optional parameter `SSS` indicating the size of the instance. For example `SYM002+1.p` is (the first version of) the second problem in the domain SYM.

For the *syntax* of the problems the Prolog syntax of the TPTP library [30] is extended by the modal operators. We use the two Prolog atoms "`box`" and "`dia`" for representing $\Box$ and $\Diamond$, respectively. The formulas $\Box F$ and $\Diamond F$ are then represented by "`box:F`" and "`dia:F`", respectively (see also Figure 1). For future extensions to multi-modal logic these atoms can, e.g., be used in Prolog terms of the form "`box(i)`"/"`dia(i)`" in which the index "`i`" is an arbitrary atom.

A header with useful information is added to the *presentation* of each problem. It is adapted from the TPTP library and includes information about the file name, the problem description, the (modal) status and the (modal) difficulty rating. An example file of a problem is given in Figure 1.

3 Conclusion

A first version of a problem library for first-order modal logic was presented. Like the TPTP library for classical logic and the ILTP library for intuitionistic logic, the objective of the QMLTP library is to put the testing and evaluation of ATP systems for first-order modal logic onto a firm basis. It will make meaningful systems evaluations and comparisons possible and help to ensure that published results reflect the actual performance of an ATP system. Experiences with existing libraries have shown that they stimulate the development of novel, more efficient calculi and implementations. Future works include adding more problems that are used within applications and the extension to, e.g., some first-order multi-modal logics.

Like other problem libraries the QMLTP library is an ongoing project. We invite all interested users to submit problems and ATP systems that use first-order modal logics (see call for problems and ATP systems in Appendix A).

[2] At the time of writing no other published ATP system for first-order modal logic could be found.

[3] All systems were run on a 3.4 GHz Xeon system using Linux, OCaml 3.10.0, and ECLiPSe Prolog 5.8. The time limit for all proof attempts was set to 600 seconds.

```
%------------------------------------------------------------------------
% File     : SYM002+1 : QMLTP v0.2
% Domain   : Syntactic (modal)
% Problem  : Instance of converse Barcan scheme
% Version  : Especial.
% English  : Everything that exists in the actual world exists in every
%            world accessible to the actual world (monotonicity)
%
% Refs     : [Brc46] [1] R. C. Barcan. A functional calculus of first
%            order based on strict implication. Journal of Symbolic
%            Logic 11, 1-16, 1946.
% Source   : [Brc46]
% Names    : Converse Barcan scheme

% Status: S4 cumulative : Theorem
% Rating: S4 cumulative : 0.00 v0.2

%
% Comments :
%------------------------------------------------------------------------

fof(con,conjecture,
(( ! [X] : ( box : ( f(x) ) ) ) => ( box : ( ! [X] : ( f(x) ) ) )
)).

%------------------------------------------------------------------------
```

Fig. 1. The presentation of the modal problem `SYM002+1.p`

References

1. S. Autexier, D. Hutter et al. VSE: formal methods meet industrial needs. *STTT*, 3(1):66–77, 2000.
2. P. Balsiger, A. Heuerding, S. Schwendimann. A Benchmark Method for the Propositional Modal Logics K, KT, S4. *Journal of Automated Reasoning*, 24:297–317, 2000.
3. B. Beckert, M. Giese, R. Hähnle, V. Klebanov, P. Rümmer, S. Schlager, P. H. Schmitt. The KeY System (Deduction Component). In F. Pfennig, Ed., *CADE-21*, LNCS 4603, pp. 379–384. Springer, 2007.
4. P. Blackburn, J. van Bentham, F. Wolter. *Handbook of Modal Logic*. Elsevier, 2006.
5. V. Boeva, L. Ekenberg. A transition logic for schemata conflicts. *Data & Knowledge Engineering*, 51(3):277–294, 2004.
6. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati. EQL-Lite: Effective First-Order Query Processing in Description Logics. In M. M. Veloso, Ed., *IJCAI-2007*, pp. 274–279, 2007.
7. M. Cialdea Mayer, S. Cerrito. Ground and Free-Variable Tableaux for Variants of Quantified Modal Logics. *Studia Logica*, 69:97–131, 2001.
8. P. R. Cohen, H. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42(3):213–261, 1990.

9. L. Fariñas del Cerro, A. Herzig, D. Longin, O. Rifi. Belief Reconstruction in Cooperative Dialogues. *AIMSA 1998*, LNCS 1480, pp. 254–266. Springer, 1998.
10. M. Fitting. Destructive Modal Resolution. *Journal of Logic and Computation*, 1(1):83–97, 1990.
11. M. Fitting, R. L. Mendelsohn. *First-Order Modal Logic*. Kluwer, 1998.
12. G. Forbes. *Modern Logic. A Text in Elementary Symbolic Logic*. Oxford University Press, 1994.
13. J. Garson. Quantification in Modal Logic. *Handbook of Philosophical Logic*, Band II, pp. 249–307. D. Reidel Publ. Co, 1984.
14. R. Girle. *Modal Logics and Philosophy*. Acumen Publ., 2000.
15. K. Gödel. An interpretation of the intuitionistic sentential logic. In J. Hintikka, Ed., *The Philosophy of Mathematics*, pp. 128–129. Oxford University Press, 1969.
16. R. C. Moore. A formal theory of knowledge and action. In J.R. Hobbs, R.C. Moore, Eds., *Formal Theories of the Commonsense World*, pp. 319–358. Ablex Pubs., 1985.
17. H. J. Ohlbach. Semantics Based Translation Methods for Modal Logics. *Journal of Logic and Computation*, 1(5):691–746, 1991.
18. J. Otten. ileanTAP: An intuitionistic theorem prover. In D. Galmiche, Ed., *TABLEAUX-97*, LNAI 1227, pp. 307–312. Springer, 1997.
19. P. F. Patel-Schneider, R. Sebastiani. A New General Method to Generate Random Modal Formulae for Testing Decision Procedures. *Journal of Articial Intelligence Research* 18, pp. 351–389, 2003.
20. R. Petrick, F. Bacchus. A knowledge-based approach to planning with incomplete information and sensing. In *AIPS-2002*, pp. 212–221.AAAI Press, 2002.
21. T. Raths, J. Otten, C. Kreitz. The ILTP Problem Library for Intuitionistic Logic. *Journal of Automated Reasoning*, 38(1-3): 261 - 271, 2007. (`http://www.iltp.de`)
22. W. Reif, G. Schellhorn, K. Stenzel. Proving System Correctness with KIV 3.0. In W. McCune, Ed., *CADE-14*, LNAI 1249, pp. 69–72. Springer, 1997.
23. R. Reiter. What Should a Database Know? *Journal of Logic Programming* 14(1&2):127–153, 1992.
24. D. Sadek, P. Bretier, F. Panaget. ARTIMIS: Natural dialogue meets rational agency. In *IJCAI-1997*, pp. 1030–1035, 1997.
25. S. Saget, M. Guyomard. Goal-oriented dialog as a collaborative subordinated activity involving collaborative acceptance. In D. Schlangen, R. Fernández, Eds., *SemDial (brandial-2006)*, pp. 131–138, 2006.
26. P.A. Spruit, R.J. Wieringa, J. Meyer. Regular database update logics. *TCS*, 254(1-2):591–661, 2002.
27. M. Steedman, R. Petrick. Planning Dialog Actions. In S. Keizer, H. Bunt, T. Baek, Eds., *SIGdial 2007*, pp. 265–272, 2007.
28. M. Stone. Towards a computational account of knowledge, action and inference in instructions. *Journal of Language and Computation*, 1:231–246, 2000.
29. M. Stone, C. Doran, B. Webber, T. Bleam, M. Palmer. Microplanning with Communicative Intentions: The SPUD System. *Computational Intelligence*, 19(4):311–381, 2003.
30. G. Sutcliffe, C.B. Suttner. The TPTP Problem Library: CNF Release v1.2.1. *Journal of Automated Reasoning*, 21(2):177–203, 1998. (`http://www.tptp.org`)
31. V. Thion, S. Cerrito, M. Cialdea Mayer. A General Theorem Prover for Quantified Modal Logics. In U. Egly, C. G. Fermüller, Eds., *TABLEAUX-2002*, LNCS 2381, pp. 266–280. Springer, 2002.

Appendix

A Call for Problems and Automated Theorem Proving Systems using First-Order Modal Logic

The Quantified Modal Logic Theorem Proving (QMLTP) library aims to provide a comprehensive and useful selection of problems in first-order modal logics. All kind of problems in a first-order (multi-)modal logic are of interest.

- Do you have an application that uses a first-order modal logic?
- Does your application generate problems in a first-order modal logic?
- Do you need to know if a formula in a first-order modal logic is valid?

Please submit your *problems* to the email address given on the QMLTP website (see below for details). By submitting your problems you can ensure that developers of ATP systems for modal logic will consider your problems when optimizing their system.

Currently available ATP systems are used to evaluate the difficulty of the problems in the library. The QMLTP library will also contain a list with currently available ATP systems for (first-order) modal logic.

- Do you have developed an ATP system for some first-order modal logic?
- Do you want to let other people know about your ATP system?
- Do you want to know how your system performs?

We would like to list your *ATP system* on the QMLTP website and consider it for evaluating the difficulty of problems in the library. Please submit details of your system to the email address given on the QMLTP website (see below).

We also appreciate any general feedback and suggestions. The website of the QMLTP library can be found at `http://www.iltp.de/qmltp`.