

The Axiomatic Translation of Modal Logic into First Order Logic

A dissertation submitted to The University of Manchester for the degree of
Master of Science
in the Faculty of Engineering and Physical Sciences

2008

Kevin John Smith

School of Computer Science

List of Contents.

List of Contents.	1
List of Figures and Tables.	4
Abstract.	10
Declaration.	11
Copyright Statement.	12
Acknowledgement.	13
1: Introduction.	14
1.1 Aims of the project.	14
1.2 Outline of Dissertation.	15
1.3 Basics of logic, modal logic and resolution.	16
1.3.1 Propositional and predicate logic.	16
1.3.2 Modal logic.	18
1.4 Translation of modal logic into first-order logic.	22
1.5 Resolution.	25
1.6 Essential knowledge of SPASS.	27
2: The Axiomatic Translation of Modal Logic.	30
2.1 Axiomatic translation of the input modal problem.	30
2.2 Inclusion of modal axioms in the axiomatic translation of modal logic.	35
2.2.1 The incorporation of the correspondence properties of common modal axioms.	35
2.2.2 The Incorporation of the schema encoding common modal axioms into axiomatic translation.	36
2.2.3 Composition in the incorporation of several modal axioms in the axiomatic translation.	40
2.2.4. Shortcuts.	48
2.2.5 Mixed translation modes.	49
2.2.6 Multi-modal target formulae.	49
2.2.7 Bi-modal axioms.	51
2.3 Resolution.	52
3: Materials and Methods.	53
4: Requirements.	56
4.1 Design strategy.	56
4.2 Informal requirements for extended-SPASS.	57
4.3 Informal requirements for the user interface.	61

4.4	Informal Requirements for a test suite.	62
5:	Specifications.	63
5.1	Read and analyze .dfg file.	64
5.1.1	Extensions to SPASS dfg syntax.	64
5.1.2	In-line axiom definition syntax.	64
5.1.3	Extensions to the standard flags mechanism to provide an axiom definition syntax and for control of axiomatic translation.	66
5.1.4	The set_axiom flag based axiom definition syntax.	69
5.1.5	Other ways to define correspondence properties.	71
5.2	Reorganize axioms and conjectures of input problem.	72
5.3	Standardization of input.	73
5.4	Perform axiomatic translation of input problem.	74
5.5	Incorporate modal axiom(s) into the translation of the input problem.	75
5.6	Consider each axiom in strict order.	78
5.7	Predicate names and setting the Precedence.	78
5.8	Pass collected terms from tasks 7 & 8 to SPASS.	80
5.9	The user interface.	80
6:	The Design and Implementation of SPASS with axiomatic translation of modal logic.	86
6.1	Introduction.	86
6.2	Data structures.	87
6.2.1	Data structures: Axiom.	87
6.2.2	Data structure: Propositional variables.	89
6.2.3	Data structure: Symbols.	89
6.2.4	Data structure: The axiom instantiation set.	91
6.2.5	Other data structures.	92
6.3	Task: Initialize EML module.	93
6.4	Task: Enter EML module from SPASS.	94
6.5	Task: Reorganize axioms and conjectures of input problem.	95
6.6	Task: Read and analyze .dfg file.	96
6.6.1	Bit-based options flags.	96
6.6.2	Modal axioms.	98
6.6.3	In-line syntax for modal axioms.	98
6.6.4	Extensions to the standard flag mechanism in SPASS.	98
6.6.5	The set_axiom() syntax.	100
6.7	Task: Standardization of input.	102

6.8	Task: Perform axiomatic translation of input problem.	103
6.9	Task: Perform translation of the input problem for each modal axiom in strict order.	105
6.9.1	Task: Encoding of modal axioms.	105
6.9.2	Task: Classical translation of modal-axioms.	107
6.9.3	Program logic for the translation of axioms: Strict order and composition.	108
6.9.4	Notes on the special handling of true/false in the translation.	110
6.10	Task: Pass collected Terms from tasks 7 & 8 to SPASS.	111
6.11	Clean up memory allocated to data, etc in EML module.	113
6.12	Utility functions.	113
6.13	The user interface.	114
6.14	Object-oriented design in Java.	118
7:	Testing and Results.	124
7.1	Software Testing.	124
7.2	Results.	128
8:	Discussion.	175
8.1.1	Is extended-SPASS maintainable?	175
8.1.2	Is extended-SPASS flexible and useful?	177
8.2.1	A candidate for a general complete formulation of the instantiation set for axiom S5 and other combinations of axioms.	179
8.2.2	A brief examination of the complexity of the axiomatic translation.	184
8.3	Further work.	190
9:	References.	192
10:	Appendix.	196
A.	Project web site at http://project.kjsmith.net/index.html .	196

Final word Count: 55940.

List of Figures and Tables.

Figure 1.1	Definitions in propositional logic and predicate logic.	17
Figure 1.2	A directed graph illustrating a Kripke frame.	20
Figure 1.3	The Model Checking Algorithm.	20
Figure 1.4	Valid propositional modal formulae.	21
Figure 1.5	The equivalence of some modal axioms and correspondence properties.	21
Figure 1.6	Modal axioms and correspondence properties.	22
Figure 1.7	Examples of Kripke frames in which modal axioms hold.	22
Figure 1.8	Inductive definition of the function, π .	23
Figure 1.9	Derivations of the correspondence property for various modal axioms.	23
Figure 1.10	A derivation of the correspondence property for axiom 4 from the literature.	25
Figure 1.11	Principles of resolution.	26
Figure 1.12	A .dfg input for SPASS and the results output by SPASS.	29
Figure 2.1	Definition of the instantiation set.	32
Figure 2.2	An example of an axiomatic translation.	32
Figure 2.3	Derivation of the simplification of the Definition for the axiomatic translation of $\text{Def}(p)$.	33
Figure 2.4	Derivation of the simplification of the Definition for the axiomatic translation of $\text{Def}(\neg\psi)$.	33
Figure 2.5	Example of an optimized axiomatic translation.	34
Table 2.6	Correspondence properties of common modal axioms.	35
Figure 2.7	An example of the incorporation of correspondence properties in a translation.	35
Table 2.8	Schema encoding of some modal axioms for axiomatic translation.	37
Figure 2.9	Derivation of schema encoding of common axioms.	37
Figure 2.10	The axiomatic translation of a target formula in axiom B.	39
Figure 2.11	The axiomatic translation of a target formula in axiom D.	40
Figure 2.12	The axiomatic translation of a target formula in axioms T and B.	45
Figure 2.13	The axiomatic translation of a target formula in axioms D and B.	45
Figure 2.14	The axiomatic translation of a target formula in axiom 5.	45

Figure 2.15	The axiomatic translation of a target formula in axiom combination $K5_oT$.	46
Figure 2.16	The axiomatic translation of a target formula in axiom $K5_o^2$.	46
Table 2.17	Axiomatic translation with combinations of axioms.	47
Table 2.18	Composition of the axiomatic translation of common modal axioms.	47
Figure 2.19	Derivation of composition of the axiomatic translation of some common modal axioms.	48
Figure 2.20	Mixed mode translations.	49
Figure 2.21	Translation of multi-modal target formulae.	50
Figure 2.22	Translation of multi-modal target formulae in axioms T_r , B_s and D_t .	50
Figure 2.23	Translation of multi-modal target formulae in with composition.	50
Figure 2.24	Schema encoding for bi-modal axioms.	51
Figure 2.25	The possible mappings of modality indices from bi-modal target formulae and bi-modal axioms.	52
Figure 5.1	Overview of sequence of tasks in execution for axiomatic translation.	63
Figure 5.2	Extensions to the dfg syntax for in-line axiom definitions.	64
Figure 5.3	Typical use of the in-line syntax.	65
Figure 5.4	Typical use of the in-line syntax for a multi-modal target formula.	65
Figure 5.5	Typical output for the in-line syntax.	65
Figure 5.6	Extensions to the standard flags mechanism.	67
Figure 5.7	The information printed by extended-SPASS for sequences of axioms requested.	68
Figure 5.8	Use of the flag-based input syntax in experimental mode.	68
Figure 5.9	Bit based flags available in extended-SPASS.	68
Figure 5.10	Typical output from extended-SPASS informing the user of the bit-based flag settings.	69
Figure 5.11	Format of the <code>set_axiom</code> syntax.	70
Figure 5.12	An example of the use of the <code>set_axiom</code> syntax.	71
Figure 5.13	Use of the <code>noAxiom</code> syntax.	72
Figure 5.14	Typical output produced by the <code>noAxiom</code> syntax.	72
Figure 5.15	Typical use of the <code>slf</code> syntax.	72
Figure 5.16	Typical output for the <code>slf</code> syntax.	72

Figure 5.17	Typical output indicating the reorganization of axioms and conjectures.	73
Figure 5.18	Typical output printed after standardization of a target formula.	74
Figure 5.19	Tasks: Standardize target formula.	74
Figure 5.20	Typical output describing the details of axiomatic translation in logic K.	75
Figure 5.21	Typical output describing the incorporation of modal axioms in the axiomatic translation.	76
Figure 5.22	Typical output describing the incorporation of modal axioms in the axiomatic translation.	76
Figure 5.23	Typical output describing the incorporation of modal axioms in the axiomatic translation with composition.	77
Figure 5.24	Typical output describing the incorporation of modal axioms as correspondence properties.	77
Figure 5.25	Typical content of the symbol cache.	79
Figure 5.26	Typical output indicating the precedence of symbols from the symbol cache.	80
Figure 5.27	Key features of the main editor of the web interface.	82
Figure 5.28	Details of the user interface.	83
Figure 5.29	Details of the modal parameters tab of the user interface.	85
Table 6.1	The AXIOM struct.	88
Table 6.2	Functions manipulating Axioms.	88
Table 6.3	Functions manipulating propositional variables.	89
Table 6.4	The AxiomSymbol struct.	91
Table 6.5	Functions manipulating Symbols.	91
Table 6.6	The AXIOMDEFLIST struct.	92
Table 6.7	Functions manipulating the instantiation set.	92
Table 6.8	Correspondence between dfg syntax and internal symbols in eml.c/eml.h.	94
Table 6.9	Functions involved in initialization.	94
Figure 6.10	Access to the eml-module code.	94
Table 6.11	Functions involved in reorganization of input axioms and conjectures.	95
Figure 6.12	The overall program flow for axiomatic translation.	95
Table 6.13	Implementation of tasks: Control flags.	97

Table 6.14	The correspondence between bitwise flags and <code>HiddenOpt</code> parameters.	97
Table 6.15	Implementation of modal axiom syntaxes.	98
Table 6.16	Implementation of in-line syntax.	98
Table 6.17	New flags implemented to support axiomatic translation.	99
Table 6.18	Implementation SPASS-type flags.	100
Table 6.19	Allowed values for axioms available to the user in the <code>set_axiom</code> syntax.	101
Table 6.20	The <code>AXIOMFLAG</code> struct.	102
Table 6.21	Implementation of <code>set_axiom()</code> mechanism.	102
Table 6.22	Functions associated with standardization of the input problem.	102
Figure 6.23	Transformations of an instantiation set element during ‘Definition’.	104
Table 6.24	Implementation of axiomatic translation of input target formula.	105
Table 6.25	The implementation of the instantiation of modal axiom.	106
Table 6.26	Implementation of the classical translation of modal axioms.	108
Table 6.27	The program flow for processing of a sequence of axioms.	109
Figure 6.28	Special handling of true and false.	110
Table 6.29	Passing collected terms from axiomatic translation to SPASS.	112
Table 6.30	Miscellaneous utility functions.	113
Figure 6.31	The design of the Jasper test system.	121
Figure 7.1	Typical <code>.dfg</code> template input file for testing axiomatic translation.	141
Figure 7.2	Typical script used to execute a series of modal-axiom combinations.	141
Figure 7.3	Typical script and <code>.dfg</code> file.	142
Figure 7.4	Formulae used for software testing.	142
Figure 7.5	Local satisfiability results from the SPASS resolution prover for the test set of target formulae in each of the listed single axioms.	145
Figure 7.6	Local satisfiability results from the SPASS resolution prover for the test set of target formulae in each of the listed axiom combinations.	147
Figure 7.7	Local satisfiability results from the SPASS resolution prover for the test set of target formulae in each of the listed single axioms.	149
Figure 7.8	Local satisfiability results from the SPASS resolution prover	

	for the test set of target formulae in each of the listed axiom combinations.	151
Figure 7.9	Local satisfiability results from the SPASS resolution prover for the test set of target formulae in each of the listed axiom combinations: Execution times.	153
Figure 7.10	Axiomatic translation of the test formulae: Summary of counter-examples.	154
Figure 7.11	Summary of the results for axiomatic translation of formulae in local satisfiability calculations in various formulations of logic S5.	155
Figure 7.12	Execution times for the eml-translation to first-order logic component of the axiomatic translation for all the examples in this study.	156
Figure 7.13	Comparing execution times of classical and axiomatic schema translations of modal axioms in local satisfiability calculations.	156
Figure 7.14	Scott-Lemmon translation (correspondence properties) for selected modal-axioms.	158
Figure 7.15	Summary of the correspondence properties and axiomatic schema of some modal axioms.	159
Figure 7.16	Derivation of the axiomatic schema of some modal axioms.	159
Figure 7.17	Local satisfiability results from the SPASS resolution prover for the test set of target formulae in each of the listed axiom combinations.	161
Figure 7.18	Local satisfiability results from the SPASS resolution prover for the test set of target formulae in each of the listed axiom combinations.	162
Figure 7.19	Local satisfiability results from the SPASS resolution prover for the test set of target formulae in each of the listed axiom combinations.	164
Figure 7.20	Comparing execution times of classical and axiomatic schema translations of modal axioms in local satisfiability calculations.	165
Figure 7.21	Global satisfiability results from the SPASS resolution prover for the test set of target formulae in each of the listed axioms.	166
Figure 7.22	Global satisfiability results from the SPASS resolution prover for the test set of target formulae in each of the listed axiom combinations.	167

Figure 7.23	Global satisfiability results from the SPASS resolution prover For the test set of target formulae in each of the listed axioms.	169
Figure 7.24	Global satisfiability results from the SPASS resolution prover for the test set of target formulae in each of the listed axiom combinations.	171
Figure 7.25	Global satisfiability results from the SPASS resolution prover for the test set of target formulae in each of the listed axiom combinations.	173
Figure 8.1	Summary of the counter-examples for axiomatic translation of formulae in local satisfiability calculations in various formulations of axiom S5 and other well-know modal axioms.	181
Figure 8.1.1	Results (for local satisfiability) for axiom formulations of the format X_oX .	181
Figure 8.1.2	Results (for local satisfiability) for axiom formulations of the format $((X_o)_1(X_o)_2..(X_o)_n)X_1X_2..X_n$.	182
Table 8.2	An example of the instantiation set for various formulations of the axiom combination 5T.	183
Table 8.3	An example of the instantiation set for various formulations of the axiom combination 4B.	183
Figure 8.4	Information required to estimate the number of subformulae introduced into any translation by incorporation of an axiom.	188

Abstract.

The axiomatic translation is a means by which propositional modal logic may be translated into first-order logic. The axiomatic translation has been implemented as part of the well-know first-order theorem prover SPASS, so that the satisfiability of an arbitrary multi-modal target formulae may be determined. Schema encodings have been implemented for a wide variety of modal axioms (including axioms T, 4, 5, B, D, alt_1 , and many other uni-modal and bi-modal axioms). Compositional subformulae may be included in the axiomatic instantiation set if required. The software is able to run in a mode in which only modal axioms and axiom combinations that have been previously proven to be complete are available. An additional experimental mode is available where arbitrary combinations of modal axioms may be used. The classical correspondence properties of modal axioms may be included in the translation, either alone or in a mixed mode together with the axiomatic schema encoding of axioms.

In the documentation provided, the axiomatic translation has been illustrated in detail with many examples, and the use of the software, and it's implementation and testing has been described. The usefulness of the software in solving the satisfiability of many problems in modal logic is illustrated. A large collection of results is presented that form a reference set of data for other investigators. In addition, features of the axiomatic translation have been investigated in a series of experiments. Data is presented on the executions times and completeness of a large number of axioms and axiom combinations, and this has been used to make suggestions for the further development of the axiomatic translation project, both practical and theoretical.

Declaration.

No portion of the work referred to in this dissertation has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright Statement.

(1) Copyright in text of this dissertation rests with the author. Copies (by any process) either in full, or of extracts, may be made **only** in accordance with instructions given by the author. Details may be obtained from the appropriate Graduate Office. This page must form part of any such copies made. Further copies (by any process) of copies made in accordance with such instructions may not be made without the permission (in writing) of the author.

(2) The ownership of any intellectual property rights which may be described in this dissertation is vested in the University of Manchester, subject to any prior agreement to the contrary, and may not be made available for use by third parties without the written permission of the University, which will prescribe the terms and conditions of any such agreement.

(3) Further information on the conditions under which disclosures and exploitation may take place is available from the Head of the School of Computer Science

Acknowledgement.

The advice and help of my supervisor Dr. Renate Schmidt is gratefully acknowledged.