

The Tweety Library Collection for Logical Aspects of Artificial Intelligence and Knowledge Representation

Matthias Thimm¹ 

Published online: 28 October 2016
© Springer-Verlag Berlin Heidelberg 2016

Abstract Tweety is a collection of Java libraries that provides a general interface layer for doing research in and working with different knowledge representation formalisms such as classical logics, conditional logics, probabilistic logics, and computational argumentation. It is designed in such a way that tasks like representing and reasoning with knowledge bases inside the programming environment are realizable in a common manner. Furthermore, Tweety contains libraries for dealing with agents, multi-agent systems, and dialog systems for agents, as well as belief revision, preference reasoning, preference aggregation, and action languages. A series of utility libraries that deal with e. g. mathematical optimization complement the collection.

Keywords Knowledge representation · Logic · Reasoning

1 Introduction

Many branches of research in knowledge representation and reasoning [5] are theoretical in nature and researchers usually do not put effort in implementation and empirical evaluation. To address this issue we developed the Tweety libraries for logical aspects of artificial intelligence and knowledge representation that provide a common infrastructure for experimental evaluation of logical frameworks within these fields.

Approaches to knowledge representation follow almost always a specific pattern. Starting from a formal syntax one can build formulas which are collected in knowledge bases. Using knowledge bases one can derive new information using either the underlying semantics of the language or a specific reasoner. For example, propositional logic is the most basic form for knowledge representation. Given some set of propositions (or atoms) one can build complex formulas using disjunction, conjunction, or negation. A set of propositional formulas, i.e., a knowledge base, can be used to derive new propositional formulas as conclusions. For instance, this can be done using the standard model-theoretic semantics of propositional logic or more sophisticated reasoning techniques such as paraconsistent reasoning.

The Tweety libraries support the implementation of such approaches in Java by providing a couple of abstract classes and interfaces for components such as `Formula`, `BeliefBase`, and `Reasoner`. Moreover, many strictly logic-based approaches to knowledge representation can also utilize further classes such as `Predicate`, `Atom`, and `Variable`, to name just a few. Currently, Tweety already contains implementations of over 18 different approaches to knowledge representation such as propositional logic, first-order logic, several approaches to probabilistic logics, and several approaches to computational models of argumentation.

The webpage of Tweety¹ provides details on all libraries, installation manuals, and further documentation. A previous paper with a description of the Tweety libraries can also be found in [34].

✉ Matthias Thimm
thimm@uni-koblenz.de

¹ Institute for Web Science and Technologies, Universität Koblenz-Landau, Universitätsstr. 1, 56070 Koblenz, Germany

¹ <http://tweetyproject.org>.

2 Project Overview

Tweety aims at providing a common framework for implementing different approaches to logical approaches of artificial intelligence in general and knowledge representation in particular. It can be used by undergraduate students to better understand logical approaches to knowledge representation by actually working with them in a familiar object-oriented manner. Moreover, the main purpose of Tweety is to allow the easy implementation of new approaches by following a given strict framework and with the benefit of easily integrating concepts and methods of other formalisms. This allows for early testing of ideas and experimental evaluation in terms of feasibility studies.

Most implementation works related to knowledge representation and reasoning are usually specific-purpose works and aim at providing effective reasoning procedures or appealing user interfaces for a single formalism. Examples of those are *protégé*² for ontology editing, *BContractor*³ for belief change in classical logic, or *Alchemy*⁴ for Markov Logic Networks. Tweety, however, provides a common framework to implement a wide variety of different approaches and does not focus on user interfaces or effective algorithms (although direct implementations and bridges to other implementations of effective algorithms are available as well).

In the following, we briefly discuss the architecture of Tweety and give an overview on its functionalities.

2.1 Software Architecture

Tweety is organized as a modular collection of Java libraries with a clear dependence structure. Each knowledge representation formalism has a dedicated Tweety library (ranging from a library on propositional logic to libraries on computational models of argumentation) which provides implementations for both syntactic and semantic constructs of the given formalism as well as reasoning capabilities. Several libraries provide basic functionalities that can be used in other libraries. Among those is the *Tweety Commons* library which contains abstract classes and interfaces for all kinds of knowledge representation formalisms. Moreover, the library *Math* contains classes for dealing with mathematical problems such as constraint satisfaction or optimization problems that often occur, in particular, in probabilistic approaches to reasoning. Most other Tweety projects deal with specific approaches to knowledge representation. Each Tweety library is organized as a Maven⁵ project. Most libraries can be used right

away as they only have dependencies to other Tweety libraries. Some libraries provide bridges to third-party libraries such as numerical optimization solvers which are not automatically found by Maven and have to be installed beforehand. However, all necessary third-party libraries can be installed by executing a single install file located within the Tweety distribution.

2.2 Libraries

An overview of the Tweety libraries is given in Table 1 which provides both the name of a library and its Java root package name. Furthermore, the final column lists references to original literature and the implemented reasoning mechanisms and solvers. There, a dagger (†) indicates that the particular reasoning mechanism has been directly implemented from the original literature, a double dagger (‡) means that a wrapper for the existing original implementation is provided, and an asterisk (*) refers to related literature.

All libraries dealing with specific knowledge representation formalisms also provide reasoning capabilities, either via direct implementations or bridges to other software solutions. As the libraries are mainly designed to be used by developers and researchers, all functionalities are exposed through APIs⁶ but several user interfaces—such as a general command line interface (located in the package `net.sf.tweety.cli`) and specific web interfaces such as for inconsistency measurement⁷—are also available. A detailed description of the functionalities of each library is out of the scope of this paper, see <http://tweetyproject.org/lib> for more information.

3 Example

Specific parts of libraries have been empirically evaluated in other works such as in [26] where opponent models for strategic argumentation are compared or in [33] where measures of coherence are applied to Markov Logic Networks.

Due to the small scope of this paper we only give a short example on propositional logic in Tweety. Consider the code snippet provided in Fig. 1.

There, a propositional knowledge base (`PlBeliefSet`) consisting of the four propositional formulas a , b , $\neg a \vee \neg b$, and $\neg a$ is defined (the propositional letters do not have any specific meaning here). Afterwards, all minimal unsatisfiable sets (**MUS**)—i. e., subset-minimal

² <http://protege.stanford.edu>.

³ <https://bitbucket.org/renatolundberg/bcontractor>.

⁴ <http://alchemy.cs.washington.edu>.

⁵ <http://maven.apache.org>.

⁶ <http://tweetyproject.org/api/1.4/>.

⁷ <http://tweetyproject.org/w/incmes/>.

Table 1 Overview on the Tweety libraries (the general package prefix is `net.sf.tweety`)

Library	Project root package	Notes
General Libraries		
Tweety Commons	<code>commons</code>	
Command Line Interface	<code>cli</code>	
Web	<code>web</code>	
Plugin	<code>plugin</code>	*JSPF ^a
Math	<code>math</code>	‡Ipsolve ^b , ‡OpenOpt ^c , ‡Apache simplex ^d , ‡Choco ^e , †Gradient descent
Graphs	<code>graphs</code>	
Logic Libraries		
Logic Commons	<code>logics.commons</code>	†[11], †[13]
Propositional Logic	<code>logics.pl</code>	‡Sat4j ^f
First-Order Logic	<code>logics.fol</code>	*[5]
Reiter's Default Logic	<code>logics.rdl</code>	†[24]
Conditional Logic	<code>logics.cl</code>	†[22], †[14]
Relational Conditional Logic	<code>logics.rcl</code>	†[16]
Probabilistic Conditional Logic	<code>logics.pcl</code>	†[27], †[32], †[23]
Relational Prob. Conditional Logic	<code>logics.rpcl</code>	†[15]
Markov Logic	<code>logics.ml</code>	†[25], ‡Alchemy ^g
Epistemic Logic	<code>logics.el</code>	*[7]
Description Logic	<code>logics.dl</code>	*[1]
Logic Translators	<code>logics.translators</code>	
Logic Programming Libraries		
Answer Set Programming	<code>lp.asp</code>	‡Clingo ^h , ‡DLV ⁱ
Dynamics in Answer Set Programming	<code>lp.asp.beliefdynamics</code>	†[17]
Nested Logic Programming	<code>lp.nlp</code>	*[19]
Argumentation Libraries		
Abstract Argumentation	<code>arg.dung</code>	†[6], †[2]
Deductive Argumentation	<code>arg.deductive</code>	†[3], †[4]
ASPIC ⁺	<code>n.s.f.arg.aspic</code>	†[21]
Structured Argumentation Frameworks	<code>arg.saf</code>	†[35]
Defeasible Logic Programming	<code>arg.delp</code>	†[9], †[29]
Logic Programming Argumentation	<code>arg.lp</code>	†[28]
Probabilistic Argumentation	<code>arg.prob</code>	†[30]
Agent Libraries		
Agents	<code>agents</code>	*[37]
Dialogues	<code>agents.dialogues</code>	†[35], †[26]
Other Libraries		
Action and Change	<code>action</code>	†[10]
Belief Dynamics	<code>beliefdynamics</code>	†[12], †[8], †[18]
Machine Learning	<code>machinelearning</code>	‡LIBSVM ^j
Preferences	<code>preferences</code>	*[36], †[31]

^a <https://code.google.com/p/jspf/>^b <http://lpsolve.sourceforge.net>^c <http://openopt.org>^d <http://commons.apache.org/math>^e <http://www.emn.fr/z-info/choco-solver/>,^f <http://www.sat4j.org>^g <http://alchemy.cs.washington.edu>,^h <http://potassco.sourceforge.net>ⁱ <http://www.dlvsystem.com>^j <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

```

PlBeliefSet k = new PlBeliefSet();
PlParser parser = new PlParser();

k.add((PropositionalFormula) parser.parseFormula("a"));
k.add((PropositionalFormula) parser.parseFormula("b"));
k.add((PropositionalFormula) parser.parseFormula("!a || !b"));
k.add((PropositionalFormula) parser.parseFormula("!a"));

PlMusEnumerator enum = new NaiveMusEnumerator<PropositionalFormula>(new Sat4jSolver());

System.out.println("MUSes: " + enum.minimalInconsistentSubsets(k));
System.out.println("MCSes: " + enum.maximalConsistentSubsets(k));

```

Fig. 1 Code snippet for determining MUSes and MCSes from propositional knowledge bases

subsets of the knowledge base that are inconsistent—and all maximal consistent sets (MCS)—i. e., subset-maximal subsets of the knowledge base that are still consistent—are determined using a naive approach based on SAT4j⁸. The output of running this code snippet is

```

MUSes: [[a, b, !a||!b], [!a, a]]
MCSes: [[a, !a||!b], [!a, b, !a||!b], [a, b]]

```

Determining MUSes and MCSes is a subtask in many approaches to inconsistent-tolerant reasoning and the approaches for enumerating those in Tweety can be easily used and customized (there are also further approaches integrated than the one illustrated).

4 Conclusions

To the best of our knowledge, Tweety is the first attempt to provide a general-purpose framework for a broad variety of knowledge representation formalisms. It is an open source project^{9,10} and can therefore be used and extended by everyone. Although Tweety is implemented in an object-oriented programming language it follows a strict declarative formal way to define concepts from theoretical knowledge representation research. Tweety is available under the GNU Lesser General Public License version 3.0. In order to contribute to Tweety contact contribute@tweetyproject.org.

Acknowledgements Tweety is being collaboratively developed by several contributors. Thanks go, among others, to Linda Briese-meister, Nils Geilen, Sebastian Homann, Tim Janus, Patrick Krümpelmann, Nico Potyka, Tjitze Rienstra, Stefan Tittel, Thomas Vengels, and Bastian Wolf.

⁸ <http://www.sat4j.org>.

⁹ <http://tweetyproject.org/>.

¹⁰ The source code of Tweety is hosted at <http://sourceforge.net/p/tweety/code/>.

References

1. Baader F, Calvanese D, McGuinness DL, Nardi D, Patel-Schneider PF (eds) (2003) The description logic handbook: theory, implementation, and applications. Cambridge University Press, New York
2. Baroni P, Caminada M, Giacomin M (2011) An introduction to argumentation semantics. *Knowl Eng Rev* 26(4):365–410
3. Besnard P, Hunter A (2001) A logic-based theory of deductive arguments. *Artif Intell* 128(1–2):203–235
4. Besnard P, Hunter A (2006) Knowledgebase compilation for efficient logical argumentation. In: Proceedings of the 10th International Conference on Knowledge Representation (KR'06), AAAI Press, pp 123–133
5. Brachman RJ, Levesque HJ (2004) Knowledge representation and reasoning. The Morgan Kaufmann series in artificial intelligence. Morgan Kaufmann Publishers, San Francisco
6. Dung PM (1995) On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif Intell* 77(2):321–358
7. Fagin R, Halpern J, Moses Y, Vardi MY (1995) Reasoning about knowledge. MIT Press, Cambridge
8. Fermé E, Hansson SO (1999) Selective revision. *Stud Log* 63(3):331–342
9. Garcia A, Simari GR (2004) Defeasible logic programming: an argumentative approach. *Theory Pract Logic Program* 4(1–2):95–138
10. Gelfond M, Lifschitz V (1998) Action languages. *Electron Trans AI* 2:193–210
11. Grant J, Hunter A (2006) Measuring inconsistency in knowledgebases. *J Intell Inf Syst* 27:159–184
12. Hansson SO (2001) A textbook of belief dynamics. Kluwer Academic Publishers, Norwell
13. Hunter A, Konieczny S (2006) Shapley inconsistency values. In: Proceedings of the 10th International Conference on Knowledge Representation (KR'06), AAAI Press, pp 249–259
14. Kern-Isberner G (2001) Conditionals in nonmonotonic reasoning and belief revision. No. 2087 in lecture notes in computer science. Springer, Berlin
15. Kern-Isberner G, Thimm M (2010) Novel semantical approaches to relational probabilistic conditionals. In: Proceedings of the Twelfth International Conference on the Principles of Knowledge Representation and Reasoning (KR'10), AAAI Press, pp 382–392
16. Kern-Isberner G, Thimm M (2012) A ranking semantics for first-order conditionals. In: Proceedings of the 20th European Conference on Artificial Intelligence (ECAI'12)

17. Krümpelmann P, Kern-Isberner G (2012) Belief base change operations for answer set programming. In: Proceedings of the 13th European conference on Logics in Artificial Intelligence (JELIA'12), Springer, pp 294–306
18. Krümpelmann P, Thimm M, Falappa MA, Garcia AJ, Kern-Isberner G, Simari GR (2011) Selective revision by deductive argumentation. In: Theory and applications of formal argumentation, Proceedings of the First International Workshop (TFAFA'11, revised selected papers). Lecture notes in artificial intelligence, vol 7132, Springer, pp 147–162
19. Lifschitz V, Tang LR, Turner H (1999) Nested expressions in logic programs. *Ann Math Artif Intell* 25:369–389
20. McCabe TJ (1976) A complexity measure. *IEEE Trans Softw Eng* 4:308–320
21. Modgil S, Prakken H (2014) The ASPIC⁺ framework for structured argumentation: a tutorial. *Argum Comput* 5(1):31–62
22. Pearl J (1990) System Z: a natural ordering of defaults with tractable applications to default reasoning. In: Proceedings of the 3rd conference on Theoretical aspects of reasoning about knowledge (TARK'90), Morgan Kaufmann Publishers, pp 121–135
23. Potyka N (2014) Linear programs for measuring inconsistency in probabilistic logics. In: Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR'14)
24. Reiter R (1980) A logic for default reasoning. *Artif Intell* 13:81–132
25. Richardson M, Domingos P (2006) Markov logic networks. *Mach Learn* 62(1–2):107–136
26. Rienstra T, Thimm M, Oren N (2013) Opponent models with uncertainty for strategic argumentation. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13)
27. Rödder W (2000) Conditional logic and the principle of entropy. *Artif Intell* 117:83–106
28. Schweimeier R, Schroeder M (2003) A parameterised hierarchy of argumentation semantics for extended logic programming and its application to the well-founded semantics. *Theory Pract Logic Program* 5(1–2):207–242
29. Stolzenburg F, Garcia A, Chesnevar CI, Simari GR (2003) Computing generalized specificity. *J Non-Class Log* 13(1):87–113
30. Thimm M (2012) A probabilistic semantics for abstract argumentation. In: Proceedings of the 20th European Conference on Artificial Intelligence (ECAI'12)
31. Thimm M (2013) Dynamic preference aggregation under preference changes. In: Proceedings of the Fourth Workshop on Dynamics of Knowledge and Belief (DKB'13)
32. Thimm M (2013) Inconsistency measures for probabilistic logics. *Artif Intell* 197:1–24
33. Thimm M (2014) Coherence and compatibility of markov logic networks. In: Proceedings of the 21st European Conference on Artificial Intelligence (ECAI'14), pp 891–896
34. Thimm M (2014) Tweety—a comprehensive collection of java libraries for logical aspects of artificial intelligence and knowledge representation. In: Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR'14), pp 528–537
35. Thimm M, Garcia AJ (2010) Classification and strategical issues of argumentation games on structured argumentation frameworks. In: van der Hoek W, Kaminka GA, Lespérance Y, Luck M, Sen S (eds) Proceedings of the Ninth International Joint Conference on Autonomous Agents and Multi-Agent Systems 2010 (AAMAS'10)
36. Walsh T (2007) Representing and reasoning with preferences. *AI Mag* 28(4):59–70
37. Weiss G (ed) (2013) Multiagent systems: a modern approach to distributed artificial intelligence, 2nd edn. MIT Press, Cambridge



Matthias Thimm received his PhD from the University of Dortmund in 2011 on the topic of probabilistic reasoning with inconsistencies and his habilitation degree from the University of Koblenz-Landau in 2016. His research interests include reasoning with uncertain and inconsistent information, in particular using approaches from computational models of argumentation and inconsistency measurement.