Detecting Conflict-Free Assumption-Based Knowledge Bases

Rolf Haenni
University of Konstanz, Center for Junior Research Fellows
D-78457 Konstanz, Germany
rolf.haenni@uni-konstanz.de

Abstract

This paper presents a simple method that reduces the problem of detecting conflict-free assumption-based knowledge bases to the problem of testing satisfiability.

Keywords: Assumption-Based Reasoning, Argumentation, Abduction, Conflicts, Satisfiability

1 Introduction

Logic-based argumentative and abductive reasoning are closely related tasks of drawing inferences under uncertainty. In both cases, knowledge is expressed in a logic-based or constraint-based language. The complete description of the available knowledge is called *knowledge base*. A particular subset of variables called *assumptions* or *assumables* is used to describe uncertain or unknown circumstances such as possible failure states of system components, different ways of interpreting statements or evidence, unknown outcomes of events, situations of unreliable sensors, witnesses, or information sources, and many more. Because both abduction and argumentation are based on this simple concept, we use *assumption-based reasoning* as a general term that covers both disciplines.[1]

The goal of argumentation is to derive from the given knowledge base *arguments* in favor and *counter-arguments* against certain *hypotheses* about the future or unknown world [15, 12, 14, 3, 4, 1, 17, 18, 7]. Intuitively, every argument provides a possible proof of the hypothesis, while counter-arguments prove the contrary of the hypothesis. Abduction is very similar, as its goal is to derive from the given knowledge base possible *explanations* for some observations [19, 24, 23, 22, 21]. A particular application of abduction is the problem of finding diagnoses for systems with an abnormal behavior [16, 8, 20, 25].

A common feature of argumentation and abduction is that both arguments and explanations are terms (conjunctions of literals) containing assumptions only. These terms are usually assumed to be consistent with the given knowledge base. In order to guarantee consistency, it is important to know the set of inconsistent

[1] In Poole's abductive framework [21], assumptions are called *hypotheses*. As a consequence, Poole speaks about *hypothetical reasoning* rather than assumption-based reasoning.

terms that are in conflict with the knowledge base. As a consequence, prior to computing arguments or explanations, it is usually necessary to compute the set of all such conflicts. However, the set of conflicts is sometimes empty. In such cases, in order to avoid unnecessary computations, it would be advantageous to detect conflict-free assumption-based knowledge bases in advance.

This paper presents a simple method that reduces the problem of detecting conflict-free assumption-based knowledge bases to the problem of testing satisfiability (SAT). Of course, since SAT is a well-known member of the class of computationally intractable NP-complete problems [5, 10], it is unlikely to find a SAT algorithm that has a fast worst-case time behavior. However, there is a wide range of clever SAT algorithms that can rapidly solve many SAT problems of practical interest. As we do not further address SAT in this paper, we refer to the literature and especially to [11] for a comprehensive overview of existing techniques.

2 Argumentative and Abductive Reasoning

Let A and P be two distinct sets of propositions and $\mathcal{L}_{A\cup P}$ the corresponding propositional language.[2] The propositions in A are the assumptions and $\xi \in \mathcal{L}_{A\cup P}$ denotes the knowledge base. Often, ξ is given by a conjunctively interpreted set $\Sigma = \{\xi_1, \ldots, \xi_r\}$ of sentences $\xi_i \in \mathcal{L}_{A\cup P}$ or, more specifically, as a set $\Sigma = \{\gamma_1, \ldots, \gamma_r\}$ of clauses $\gamma_i \in \mathcal{C}_{A\cup P}$, where $\mathcal{C}_{A\cup P}$ denotes the set of all (proper) clauses over $A \cup P$ (including the empty clause $\bot$).

A *term* is a conjunction of non-repeating literals. We use $\mathcal{T}_A$ to denote the set of all terms consisting of assumptions only (including the empty term $\top$). Furthermore, $N_A = \{0,1\}^{|A|}$ denotes the set of all possible configurations relative to A. The elements $s \in N_A$ are called *scenarios* and represent possible states of the unknown or future world.[3] Note that every term $\tau \in \mathcal{T}_A$ has a corresponding set $M_A(\tau) \subseteq N_A$ of possible scenarios called *models* of τ for which τ evaluates to 1. Both argumentation and abduction are based on the idea that one particular scenario $\hat{s} \in N_A$ is the *true* scenario.

Of course, it is assumed that the true scenario is not in conflict with the given knowledge base. If $s \in N_A$ is an arbitrary scenario and $\xi_{\leftarrow s}$ the formula obtained from ξ by instantiating all the assumptions according to their values in s, then

$$C_A(\xi) \;=\; \{s \in N_A : \xi_{\leftarrow s} \models \bot\} \tag{1}$$

denotes the set of *conflicting* scenarios of ξ relative to A. Sometimes, the elements of $C_A(\xi)$ are also called *contradictory* or *inconsistent* scenarios. Since the set $C_A(\xi)$ is sometimes intractably large, an appropriate representation is needed. One possibility is to consider the set

$$C(\xi) \;=\; \{\tau \in \mathcal{T}_A : M_A(\tau) \subseteq C_A(\xi)\} \;=\; \{\tau \in \mathcal{T}_A : \tau \wedge \xi \models \bot\} \tag{2}$$

of *conflicting* terms whose models are all conflicting scenarios. Note that $C(\xi)$ is an *upward-closed* set. This means that $\tau \in C(\xi)$ implies $\tau' \in C(\xi)$ for all longer

[2] To simplify matters, we restrict our discussion in this paper to propositional logic.

[3] Note that in Poole's abductive framework [21], terms $\tau \in \mathcal{T}_A$ are called scenarios.

terms $\tau' \in \mathcal{T}_A$ with $\tau' \supseteq \tau$. Furthermore, a term $\tau \in C(\xi)$ is called *minimal* with respect to $C(\xi)$, if $C(\xi)$ contains no shorter term $\tau' \subset \tau$. The corresponding set of minimal conflicting terms

$$\mu C(\xi) \;=\; \{\tau \in C(\xi) : \neg\exists\tau'{\in}C(\xi), \tau'{\subset}\,\tau\} \tag{3}$$

is obtained from $C(\xi)$ by dropping all non-minimal terms. The elements of $\mu C(\xi)$ are also called *minimal conflicts* of ξ. Note that

$$C_A(\xi) \;=\; \cup\{M_A(\tau) : \tau \in C(\xi)\} \;=\; \cup\{M_A(\tau) : \tau \in \mu C(\xi)\}. \tag{4}$$

Thus, in order to exclude conflicting scenarios, it is sufficient to know the minimal conflicts. However, deriving $\mu C(\xi)$ from ξ may be very expensive, even in cases where $\mu C(\xi)$ is small or even empty.

3 Computing Conflicts

The problem of computing minimal conflicts is closely related to the problems of computing *prime implicates* (or *prime implicants*). Conflicts are conjunctions $\tau \in \mathcal{T}_A$ for which $\tau \wedge \xi \models \bot$ holds. This condition can be rewritten as $\xi \models \neg\tau$. Conflicts are therefore negations of implicates of ξ which are in $\mathcal{C}_A$. In other words, if $\gamma \in \mathcal{C}_A$ is an implicate of ξ, then $\neg\gamma \in \mathcal{T}_A$ is a conflict of ξ. Furthermore, if $\gamma \in \mathcal{C}_A$ is a prime implicate of ξ, then $\neg\gamma$ is a minimal conflict. We use $PI(\xi)$ to denote the set of all prime implicates of ξ. If $\neg\Psi$ is the set of conjunctions obtained from a set of clauses Ψ by negating the corresponding clauses, then we can write

$$\mu C(\xi) \;=\; \neg(PI(\xi) \cap \mathcal{C}_A). \tag{5}$$

Since computing prime implicates is known to be NP-complete in general, the above approach is only feasible when ξ is relatively small. However, when A is small enough, many prime implicates of ξ are not in $\mathcal{C}_A$. Such irrelevant prime implicates can be avoided by the method described in [15, 12, 13]. It is assumed that ξ is given as a set $\Sigma \subseteq \mathcal{C}_{A\cup P}$ of clauses over $A \cup P$. The procedure is based on two operations

$$\begin{aligned} Cons_Q(\Sigma) &= Cons_{x_1} \circ \cdots \circ Cons_{x_q}(\Sigma), & (6)\\ Elim_Q(\Sigma) &= Elim_{x_1} \circ \cdots \circ Elim_{x_q}(\Sigma), & (7) \end{aligned}$$

where $Q = \{x_1, \ldots, x_q\} \subseteq A \cup P$ is subset of propositions appearing in Σ. Both operations repeatedly apply more specific operations $Cons_x(\Sigma)$ and $Elim_x(\Sigma)$, respectively, where x is a proposition in Q. Let Σ_x denote the set of clauses of Σ containing x as a positive literal, $\Sigma_{\bar{x}}$ the set of clauses containing x as a negative literal, and $\Sigma_{\dot{x}}$ the set of clauses not containing x. Of course, we have $\Sigma = \Sigma_x \cup \Sigma_{\bar{x}} \cup \Sigma_{\dot{x}}$. Furthermore, if

$$\rho(\Sigma_x, \Sigma_{\bar{x}}) \;=\; \{\vartheta_1{\vee}\vartheta_2 : \, x{\vee}\vartheta_1 \in \Sigma_x, \, \neg x{\vee}\vartheta_2 \in \Sigma_{\bar{x}}\} \tag{8}$$

denotes the set of all resolvents of Σ relative to x, then the two basic operations are defined by

$$Cons_x(\Sigma) \;=\; \mu(\Sigma \cup \rho(\Sigma_x, \Sigma_{\bar{x}})), \quad Elim_x(\Sigma) \;=\; \mu(\Sigma_{\dot{x}} \cup \rho(\Sigma_x, \Sigma_{\bar{x}})), \tag{9}$$

where the μ-operator means dropping non-minimal clauses. Thus, $Cons_Q(\Sigma)$ computes all the resolvents (consequences) of Σ relative to the propositions in Q and adds them to Σ. Note that if Q contains all the proposition in Σ, then $Cons_Q(\Sigma) = PI(\Sigma)$. In contrast, $Elim_Q(\Sigma)$ eliminates all the propositions in Q from Σ and returns a new set of clauses whose set of models corresponds to the projection of the original set of models to $(A \cup P) \setminus Q$. Elimination is sometimes called *forgetting* and is known to be NP-complete [6].

The set of the minimal conflicts can then be computed in two different ways by

$$\mu C(\xi) \;=\; \neg Cons_A(Elim_P(\Sigma)) \;=\; \neg Elim_P(Cons_A(\Sigma)). \tag{10}$$

In most practical applications, computing the consequences relative to the propositions in A is trivial. In contrast, the elimination of the propositions in P is usually more difficult and becomes even infeasible as soon as Σ has a certain size. Note that from a theoretical point of view, the order in which the propositions in P are eliminated is irrelevant [15], whereas from a practical point of view, it critically influences the efficiency of the procedure. The elimination process is a particular instance of Shenoy's fusion algorithm [26, 27] as well as of Dechter's bucket elimination procedure [9].

Today's state of the art among the methods for computing conflicts, arguments, and abductive explanations is a convenient anytime algorithm that can be interrupted at any time returning the solution found so far [13]. The quality of the approximation increases monotonically when more computational resources are available. The method is based on cost functions [12] and returns lower and upper bounds.

4 Detecting Non-Conflicting Knowledge Bases

Let ξ be an arbitrary assumption-based knowledge base. Of course, we can use the procedure of the previous section to find out whether there are conflicts or not. However, running the complete elimination procedure may possibly be very expensive, even in cases where $\mu C(\xi) = \emptyset$. In the following, we will describe a method that simplifies the detection of conflict-free knowledge bases.

Let $\gamma \in \mathcal{C}_{A \cup P}$ be an arbitrary clause over $A \cup P$. Without loss of generality, it is always possible to split γ into sub-clauses $\gamma_A \in \mathcal{C}_A$ and $\gamma_P \in \mathcal{C}_P$ with $\gamma = \gamma_A \vee \gamma_P$. As we will see below, only the sub-clauses γ_P are relevant for detecting conflict-free knowledge bases. Thus, if $\Sigma \subseteq \mathcal{C}_{A \cup P}$ is an arbitrary set of clauses over $A \cup P$, then

$$\Sigma_P \;=\; \{\gamma_P : \gamma \in \Sigma\} \;\subseteq\; \mathcal{C}_P \tag{11}$$

denotes a new set of clauses obtained by dropping all assumptions. As an effect of this, Σ_P may contain many clauses that are subsumed by others. A corresponding minimal set $\mu\Sigma_P$ is obtained from Σ_P by removing all subsumed clauses. Of course, Σ_P and $\mu\Sigma_P$ are logically equivalent, but $\mu\Sigma_P$ is often considerably smaller.

Theorem 1 *Let $\xi \in \mathcal{L}_{A\cup P}$ be a knowledge base given as a set of clauses $\Sigma \subseteq \mathcal{C}_{A\cup P}$. If $\Sigma_P \subseteq \mathcal{C}_P$ is the set of clauses obtained from Σ as defined above, then $\Sigma_P \not\models \bot$ (or $\mu\Sigma_P \not\models \bot$) implies $C(\xi) = \emptyset$ (and $\mu C(\xi) = \emptyset$).*

Proof: Let $\xi_P = \wedge\Sigma_P$ be the conjunction of clauses of Σ_P. From $\gamma_P \models \gamma$ for all $\gamma \in \Sigma$ follows $\xi_P \models \xi$. Now, suppose $C(\xi) \neq \emptyset$ and $\tau \in C(\xi)$. This implies $\tau \wedge \xi \models \bot$ and thus $\tau \wedge \xi_P \models \bot$. Since ξ_P contains no assumptions, this is only possible if $\xi_P \models \bot$. Thus, $C(\xi) \neq \emptyset$ implies $\xi_P \models \bot$, and the other way round, $\xi_P \not\models \bot$ implies $C(\xi) = \emptyset$.
□

Note that $\Sigma_P \models \bot$ does not necessarily mean that $C_A(\xi) \neq \emptyset$. For example, if $A = \{a\}$ and $P = \{p\}$, then $\xi = (a \vee p) \wedge (\neg a \vee \neg p)$ is a conflict-free knowledge base that implies $\xi_P = p \wedge \neg p$ and thus $\xi_P \models \bot$. The above theorem can thus be used to detect some (but not all) cases of conflict-free assumption-based knowledge bases.

Example 1 *Let $A = \{a_1, \ldots, a_n, b_1, \ldots, b_n\}$ and $P = \{p_1, \ldots, p_n\}$ be the given sets of propositions. Furthermore, suppose that ξ is given as a set clauses*

$$\Sigma = \left\{ \begin{array}{ll} a_1 \vee p_1, & b_1 \vee p_1, \\ a_2 \vee \neg p_1 \vee p_2, & b_2 \vee \neg p_1 \vee p_2, \\ a_3 \vee \neg p_2 \vee p_3, & b_3 \vee \neg p_2 \vee p_3 \\ \quad\vdots & \quad\vdots \\ a_n \vee \neg p_{n-1} \vee p_n, & b_n \vee \neg p_{n-1} \vee p_n \end{array} \right\}.$$

Obviously, ξ is a conflict-free knowledge base. By running the elimination procedure of the previous section, we observe a worst-case scenario with an exponentially increasing number of clauses during the process. Before eliminating the last proposition, we always get a total number of 2^n clauses for which no further resolutions are possible (independently of the actual elimination ordering). Of course, this is not feasible if n exceeds a certain limit. In contrast, by applying Theorem 1, we get

$$\Sigma_P = \{p_1, \neg p_1 \vee p_2, \neg p_2 \vee p_3, \ldots, \neg p_{n-1} \vee p_n\}$$

for which $\Sigma_P \not\models \bot$ and thus $C_A(\xi) = \emptyset$ is easily proved in linear time.

Even if SAT is NP-complete in general, this example demonstrates how the method of Theorem 1 sometimes tremendously reduces the complexity of detecting conflict-free knowledge bases from $O(2^n)$ to $O(n)$. This is typical in many practical examples from the author's domain of interest [2].

5 Conclusion

Even if the set of minimal conflicts of an assumption-based knowledge base is small or empty, effectively computing the minimal conflicts is not feasible in the worst case. However, we have shown how conflict-free knowledge bases, which are common in many practical applications, can sometimes be detected more easily by testing the satisfiability of a corresponding simplified knowledge base. In certain cases, this reduces tremendously the necessary time of computation, even if SAT is also NP-complete in the worst case.

Acknowledgements

Research supported by (1) Alexander von Humboldt Foundation, (2) German Federal Ministry of Education and Research, (3) German Program for the Investment in the Future.

References

[1] L. Amgoud and C. Cayrol. A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence*, 34(1–3):197–215, 2002.

[2] B. Anrig, R. Bissig, R. Haenni, J. Kohlas, and N. Lehmann. Probabilistic argumentation systems: Introduction to assumption-based modeling with ABEL. Technical Report 99-1, Institute of Informatics, University of Fribourg, 1999.

[3] Ph. Besnard and A. Hunter. Towards a logic-based theory of argumentation. In *Proceedings of the National Conference on Artificial Intelligence (AAAI'2000)*, pages 411–416. MIT Press, 2000.

[4] Ph. Besnard and A. Hunter. A logic-based theory of deductive arguments. *Artificial Intelligence*, 128:203–235, 2001.

[5] S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the* 3^{rd} *Annual ACM Symposium on Theory of Computing*, pages 151–158, 1971.

[6] A. Darwiche and P. Marquis. A knowledge compilation map. *Journal of Artificial Intelligence Research*, 17:229–264, 2002.

[7] J. de Kleer. An assumption-based TMS. *Artificial Intelligence*, 28:127–162, 1986.

[8] J. de Kleer, A. K. Mackworth, and R. Reiter. Characterizing diagnosis and systems. *Artificial Intelligence*, 56:197–222, 1992.

[9] R. Dechter. Bucket elimination: a unifying framework for reasoning. *Artificial Intelligence*, 113(1–2):41–85, 1999.

[10] M. Garey and D. Johnson. *A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, 1979.

[11] J. Gu, P. W. Purdom, J. Franco, and B. W. Wah. Algorithms for the satisfiability (SAT) problem: a survey. In D. Du, J. Gu, and P. M. Pardalos, editors, *Satisfiability Problem: Theory and Applications*, volume 35 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 19–152. American Mathematical Society, 1997.

[12] R. Haenni. Cost-bounded argumentation. *International Journal of Approximate Reasoning*, 26(2):101–127, 2001.

[13] R. Haenni. A query-driven anytime algorithm for argumentative and abductive reasoning. In D. Bustard, W. Liu, and R. Sterrit, editors, *Soft-Ware 2002, 1st International Conference on Computing in an Imperfect World*, LNCS 2311, pages 114–127. Springer-Verlag, 2002.

[14] R. Haenni, B. Anrig, J. Kohlas, and N. Lehmann. A survey on probabilistic argumentation. In *Proceedings of the 6th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'01)*, pages 19–25, Toulouse, France, 2001.

[15] R. Haenni, J. Kohlas, and N. Lehmann. Probabilistic argumentation systems. In J. Kohlas and S. Moral, editors, *Handbook of Defeasible Reasoning and Uncertainty Management Systems, Volume 5: Algorithms for Uncertainty and Defeasible Reasoning*, pages 221–288. Kluwer, Dordrecht, 2000.

[16] J. Kohlas, B. Anrig, R. Haenni, and P. A. Monney. Model-based diagnostics and probabilistic assumption-based reasoning. *Artificial Intelligence*, 104:71–106, 1998.

[17] J. Kohlas and P. A. Monney. *A Mathematical Theory of Hints. An Approach to the Dempster-Shafer Theory of Evidence*, volume 425 of *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag, 1995.

[18] D. McAllester. Truth maintenance. In R. Smith and T. Mitchell, editors, *Proceedings of the Eighth National Conference on Artificial Intelligence*, volume 2, pages 1109–1116. AAAI Press, 1990.

[19] G. Paul. Approaches to abductive reasoning – an overview. *Artificial Intelligence Review*, 7(2):109–152, 1993.

[20] Y. Peng and J. Reggia. *Aductive Inference Model for Diagnostic Problem Solving*. Springer-Verlag, 1990.

[21] D. Poole. A logical framework for default reasoning. *Artificial Intelligence*, 36(1):27–47, 1988.

[22] D. Poole. Explanation and prediction: An architecture for default and abductive reasoning. *Computational Intelligence*, 5(2):97–110, 1989.

[23] D. Poole. A methodology for using a default and abductive reasoning system. *International Journal of Intelligent Systems*, 5(5):521–548, 1990.

[24] D. Poole. Probabilistic horn abduction and bayesian networks. *Artificial Intelligence*, 64:81–129, 1993.

[25] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–95, 1987.

[26] P. P. Shenoy. Binary join trees. In E. Horvitz and F. Jensen, editors, *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence (UAI-96)*, pages 492–499. Morgan Kaufmann Publishers, 1996.

[27] P. P. Shenoy. Binary join trees for computing marginals in the shenoy-shafer architecture. *International Journal of Approximate Reasoning*, 17(2–3):239–263, 1997.