Precise Exponential-Time Complexity of Non-Monotonic Reasoning

Mohamed Maizia[1,2]

[1]*Department of Computer Science and Informatics, School of Engineering, Jönköping University, Jönköping, Sweden*
[2]*Department of Computer and Information Science, Linköping University, Linköping, Sweden*

Abstract

NP-hard problems frequently occur in many real-world situations due to their rich modeling power. Even though only superpolynomial algorithms are currently known, there is still a large practical incentive to find faster algorithms, and significant improvements over brute force search can in general be achieved. In this project we investigate superpolynomial algorithms for non-monotonic reasoning problems, with a particular focus on propositional abduction. Despite seeing many real-world applications, the precise exponential time complexity of abduction is currently a blind spot, and no improved algorithms are known for the NP-hard cases. We will issue a systematic attack on the complexity of abduction, and other forms of non-monotonic reasoning, by constructing faster algorithms and simultaneously investigate how close to optimal our algorithms are by proving new lower bounds under the exponential-time hypothesis. To study the complexity in a systematic way we use the constraint based framework employing the algebraic approach based on (partial) polymorphisms. To the best of our knowledge we are the first to launch a systematic attack on the exponential time complexity of a problem complete for the second level of the polynomial hierarchy. We will hereby advance tools for exponential time complexity analysis of NP-complete problems to problems beyond NP.

Keywords

Non-monotonic reasoning, Abductive reasoning, Exponential time complexity, Constraint satisifaction

1. Introduction and problem description

NP-hard problems frequently occur in many real-world situations due to their rich modeling power. Even though only superpolynomial algorithms are currently known, there is still a large practical incentive to find faster algorithms, and significant improvements over brute force search can in general be achieved. In this project we will investigate superpolynomial algorithms for non-monotonic reasoning problems, with a particular focus on the propositional *abduction* problem. Despite seeing many applications areas, such as scientific discovery [1], network security [2], logic programming [3], computational biology [4], medical diagnosis [5], knowledge base updates [6], and explainability in machine learning [7], the precise exponential time complexity of abduction is currently a blind spot, and *no* improved algorithms are known for the NP-hard cases. We will issue a systematic attack on the complexity of abduction, and other forms of non-monotonic reasoning problems, by constructing faster algorithms and simultaneously investigate how close to optimal our algorithms are by proving new lower bounds under the *exponential-time hypothesis*.

The classical complexity of abduction, and non-monotonic reasoning problems in general, is well understood. The predominant method for characterizing problems in this framework is to use a *constraint* based framework, and analyze the complexity when only certain types of constraints are allowed. Unfortunately, tractable variants are often heavily reduced in expressiveness, and posses significantly reduced practical relevance. Yet, the exact algorithmic complexity of the intractable, yet practically relevant, variants is completely unknown, and has received little attention. Beyond completeness for a specific level of the polynomial hierarchy, *no* precise upper and lower bounds of the exponential time complexity are known. According to Cygan et al., tools to precisely analyze the exponential time complexity of NP-complete problems are in its infancy [8]. For problems at higher levels of the polynomial hierarchy the situation is even more dire. Are algorithmic approaches for

ICLP DC 2025 - 21st Doctoral Consortium (DC) on Logic Programming, University of Calabria, Rende, Italy | September 12-19, 2025
mohamed.maizia@ju.se (M. Maizia)

problems in NP still usable? Are the tools to obtain lower bounds still usable? How can these tools be adapted and advanced? Why are no sharp upper bounds known for problems in non-monotonic reasoning? Are these problems fundamentally different from e.g. satisfiability problems? Addressing these questions can lead the way to both improved practical solvers, and an increased understanding of the relative complexity for *all* intractable abduction problems.

There exists a wealth of existing algorithmic strategies in the realm of exponential time algorithms. These form a good foundation but tailored methods will be needed as well. To study the complexity of all possible constraint-based restrictions we will employ the so-called *algebraic approach* based on partial polymorphisms, where computational properties of abduction problems in a very fine-grained way can be related to partially defined, higher-order homomorphisms. Crucially, the algebraic approach makes it possible to study complexity in a *general* setting and hints at the possibility of obtaining improved algorithms for entire classes of hard problems. Lower bounds, important for proving optimality or limitations of certain algorithmic schemes, will be explored in the context of the highly influential (strong) *exponential time hypothesis* ((S)ETH).

2. Background and overview of the existing literature

Non-monotonic reasoning and its computational complexity Non-monotonic reasoning formalisms originate in the 1970's as a formal model for human reasoning. It was recognized that monotonic logics are not well suited to model real-world reasoning: one needs to be able to reason with incomplete knowledge, and hence learning new information may invalidate previously valid conclusions (non-monotonicity). Applications today lie in e.g. artificial intelligence, human reasoning, database theory, and knowledge representation. Many forms of non-monotonic reasoning have been developed, and the most prominent ones are Abduction, Circumscription, Default logic, Autoepistemic logic, and, more recently, Argumentation. Most forms of non-monotonic logics build on top of a monotonic logic where they borrow notions of consistency and entailment in order to define a higher level of conclusion relation. One can thus model a non-monotonic logic with different kinds of monotonic logics (e.g. first order logic, propositional logic), resulting in different expressive richness and different computational costs to perform reasoning. In this project we primarily focus on propositional logic. It delivers rich enough expressiveness for many applications while the computational cost is usually confined to PSPACE.

When modeled in propositional logic, many reasoning tasks in non-monotonic logics are at least NP-hard, but quite frequently even harder. We have, for instance, completeness for the second level of the polynomial hierarchy for the problems of explanation-existence in Abduction [9], deduction in Circumscription [10], extension- and expansion-existence in Default logic and Autoepistemic logic [11], and argument-existence in logic-based Argumentation [12]. These intractability results have led to extensive investigations of the computational complexity of these formalisms with the objective to understand more precisely the sources of intractability. One line of research has looked into fragments of propositional logic. Often this has been done in a systematic way, that is, *all* fragments of propositional logic within a defined framework have been considered. The constraint approach, also known as the *algebraic approach*, or *Schaefer's framework*, has been considered, where one considers formulas in generalized conjunctive normal form. This approach covers well-known fragments such as Horn, bijunctive, and affine constraints, see e.g. [13, 14]. Another systematic approach is known as *Post's framework* where one considers formulas built from composition of a restricted set of Boolean functions. This approach covers different fragments such as monotonic, c-separating, and self-dual functions, see e.g. [15]. Another line of research has looked into the parametrized complexity of these formalisms. Besides treewidth, problem specific parameters have been considered, see e.g. [16, 17].

The synthesis of these extensive efforts is that the identified tractable fragments (polynomial time solvable) are surprisingly often of *low* expressiveness and admit relatively simple algorithms. Apart from the powerful treewidth parameter, the same holds for fixed-parameter tractability [16, 17]. A notable exception to this is the affine fragments which typically involve solving a system of linear equations over GF(2), see e.g. [14]. Fragments of *interesting* expressiveness remain intractable, i.e., they are at least (co-)NP-hard (W[1]-hard). Despite these insights, to date, nothing is known about the precise exponential time complexity of the intractable fragments.

Abduction The process of abduction seeks to identify explanations for an observed manifestation, given some background knowledge base and a set of allowed hypotheses to build an explanation from. It dates back to Peirce [18] and has fundamentally influenced several areas in and around artificial intelligence [15, 9]. Numerous application areas exist, including scientific discovery [1], network security [2], logic programming [3], computational biology [4], medical diagnosis [5], knowledge base updates [6], and explainability in machine learning [7]. We follow the formalization of logic-based abduction from Eiter and Gottlob [9]. We are given the knowledge base KB as a set of propositional formulas, the hypotheses as a set of variables H, and the manifestation M as a propositional formula. An *explanation* for an instance (KB, H, M) is a set $E \subseteq H$ such that $\mathrm{KB} \cup E$ is consistent and logically implies the M. This type of abduction is often referred to as *positive* abduction since an explanation is formed upon positive literals only. If one allows negative literals as well, one speaks of *symmetric* abduction. Another point of variation is the type of manifestation. In this project we stick to the most commonly studied manifestation type: *positive term*. We denote the corresponding explanation-existence problem by Abd for symmetric abduction, and by P-Abd for positive abduction.

We define the abduction problem parametrized by a constraint language as Nordh and Zanuttini [14]. A *constraint language* Γ is a set of finitary Boolean relations. A Γ-constraint is an application of a k-ary relation $R \in \Gamma$ to a k-tuple of variables, written $R(x_1, \ldots, x_k)$. A Γ-formula is a finite set of Γ-constraints (interpreted as conjunction). A constraint $C = R(x_1, \ldots, x_k)$ is *satisfied* by an assignment σ, if $(\sigma(x_1), \ldots, \sigma(x_k)) \in R$. A Γ-formula φ is *satisfiable* if there exists an assignment σ that satisfies all constraints in φ simultaneously. In this case σ is called a *model* of φ. A Γ-formula φ *entails* a Γ-formula ψ if every model of φ satisfies ψ. We identify a set of variables E (or M) with the conjunction of its members, that is, a positive term. The symmetric abduction problem parametrized by Γ, denoted $\textsc{Abd}(\Gamma)$, is defined as follows. We are given (KB, H, M) as above, where the knowledge base KB is now a Γ-formula. An *explanation* is a set $E \subseteq \mathrm{Lit}(H)$ such that (1) $\mathrm{KB} \wedge E$ is satisfiable, and (2) $\mathrm{KB} \wedge E$ entails M, where $\mathrm{Lit}(H)$ denotes the set of literals formed upon variables from H. The positive abduction problem, denoted P-$\textsc{Abd}(\Gamma)$, is defined analogously, but an explanation E is required to be positive, that is, $E \subseteq H$.

Let us also remark that the straightforward exhaustive search algorithm for (P-)$\textsc{Abd}(\Gamma)$ goes through all possible sets of explanations ($2^{|H|}$ many) and then verifies conditions 1 and 2 (by going through 2^n many assignments). If we omit polynomial factors in the input size then this leads to a running time of $2^{|H|} \cdot 2^n$. Again, we stress that *no* better upper bounds are known, for *any* Γ such that (P-)$\textsc{Abd}(\Gamma)$ is intractable.

3. Goal of the research

The main goal of this project is to determine the precise exponential time complexity of non-monotonic reasoning, and in particular of propositional abduction. A secondary goal is to develop tools to analyze exponential time complexity of problems beyond NP. This will compound exploring in how far current tools can be extended as well as developing new tools and methods.

4. Current status and first results

We currently explore the exponential time complexity of propositional abduction parameterized by a constraint language Γ, that is, $\textsc{Abd}(\Gamma)$ and P-$\textsc{Abd}(\Gamma)$. This allows to investigate the problem's complexity systematically in *all* fragments of propositional logic in the above mentioned *Schaefer's framework*. The classical complexity of (P-)$\textsc{Abd}(\Gamma)$ is completely detemermined [14]. Constraint languages Γ such that (P-)$\textsc{Abd}(\Gamma)$ is intractable (at least NP- or coNP-hard) are the particular focus of this investigation.

The current status is that for several such Γ we have either proven lower bounds under (S)ETH or developed non-trivial algorithms, improving on the above-mentioned exhaustive search approach. These first results have been submitted as a paper to IJCAI 2025 and have been accepted for publication. Some of the main results of the paper can be summarized as follows. We use the O^*-notation to suppress polynomial factors, and denote by PPL propositional logic.

- An close analysis of the exhaustive search scheme for Abd (full PPL) delivers the base-line of $O^*(2^n)$.
- An close analysis of the exhaustive search scheme for P-Abd (full PPL) delivers the base-line of $O^*(3^{|H|} \cdot 2^{n-|H|}) = O^*(1.5^{|H|} \cdot 2^n)$ or $O^*(3^n)$.
- A much more sophisticated exhaustive search scheme for P-Abd (full PPL) improves this to $O^*(2^n)$.
- If for a constraint language Γ all models of a given Γ-formula can be exhaustively enumerated in *improved time*, that is, in time $O^*(c^n)$ for a $c < 2$, then Γ is called *sparse*. We show that if Γ is sparse, then $\text{Abd}(\Gamma)$ can be solved in improved time $O^*(c^n)$, but trading in exponential space of $O^*(min(c^n, 2^{|H|}))$.
- Similarly, but with a very different algorithm, also $\text{P-Abd}(\Gamma)$ can be solved in improved time $O^*(c^n)$ for sparse Γ, but trading in exponential space of $O^*(c^n)$.
- We show the existence of sparse languages Γ where $\text{(P-)Abd}(\Gamma)$ is intractable. In particular this contains Σ_2^P-complete cases. This might be, to the best of our knowledge, the first example in the literature of a Σ_2^P-complete problem that can be solved in improved time.
- We show that under SETH, there is no $c < 1$ such that Abd(4-CNF) is solvable in 2^{cn} time. For P-Abd(4-CNF) we obtain the SETH lower bound of 1.4142^{cn}
- We also consider several NP- and coNP-complete languages and establish lower bounds as well as improved algorithms (not only based on sparse languages). For instance, we show that NP-complete Abd(k-CNF$^+$) can be solved in improved time $O^*(c^{|H|})$, while Abd(CNF$^+$) can not be solved in time 1.2599^{cn} or $1.4142^{c|H|}$ for any $c < 1$, assuming SETH.

5. Open issues and expected achievements

Several open questions and future directions arise from the current status.

- Degree bounded variables. Bounding the degree of a variable in the knowledge base (input formula) can potentially be exploited to obtain improved algorithms. Several examples suggest that this should be feasible for abduction as well though advanced exhaustive enumeration schemes. However, current attempts have been unsuccessful.
- 3-CNF formulas. While we have a sharp SETH lower bound for Abd(4-CNF), we have no lower bound for Abd(3-CNF), and neither an improved algorithm. This is unsatisfying. Given that both problems are finite-ary and Σ_2^P-complete, one would expect them to show similar behaviour. But currently the question is wide open.
- While we have several upper and lower bounds for specific languages, the results often do not extend to larger classes of constraint languages in the vein of Schaefer's framework. Here, a class is defined as a set of relations that all share the common property of being 'closed' under a certain polymorphism. For instance, Horn languages are characterized by the polymorphism $\wedge$, that is, the binary AND function. The classical complexity is usually the same for all languages within a class. This does generally not hold for precise exponential time complexity. The deeper reason for this is that any two languages from the same class allow for polynomial many-one reductions between each other, but this type of reduction does generally not preserve exact exponential running time. While this burden can not be circumvented in general, it would still be desirable to obtain more general characterizations of the precise exponential running time behaviour of all languages with the same class.

- More generally, other forms of non-monotonic reasoning shall be explored. Also for e.g. Argumentation or Circumscription the precise exponential time complexity is a blind spot and deserves a closer look.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] K. Inoue, T. Sato, M. Ishihata, Y. Kameya, H. Nabeshima, Evaluating abductive hypotheses using an EM algorithm on bdds, in: Proc. 21st Internat. Joint Conf. on Artificial Intelligence (IJCAI'09), 2009, pp. 810–815.

[2] M. Alberti, F. Chesani, M. Gavanelli, E. Lamma, P. Mello, P. Torroni, Security protocols verification in abductive logic programming: A case study, in: Engineering Societies in the Agents World VI, 6th Internat. Workshop, ESAW'05, 2005, pp. 106–124.

[3] F. Lin, J. You, Abduction in logic programming: A new definition and an abductive procedure based on rewriting, Artif. Intell. 140 (2002) 175–205. URL: https://doi.org/10.1016/S0004-3702(02)00227-8.

[4] O. Ray, A. Antoniades, A. C. Kakas, I. Demetriades, Abductive logic programming in the clinical management of HIV/AIDS, in: Proc. 17th European Conf. on Artificial Intelligence, 2006, pp. 437–441.

[5] M. Obeid, Z. Obeid, A. Moubaiddin, N. Obeid, Using description logic and abox abduction to capture medical diagnosis, in: Proc. 32nd Internat. Conf. on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2019, 2019, pp. 376–388.

[6] C. Sakama, K. Inoue, An abductive framework for computing knowledge base updates, Theory Pract. Log. Program. 3 (2003) 671–713.

[7] A. Ignatiev, N. Narodytska, J. Marques-Silva, Abduction-based explanations for machine learning models, in: Proc. 33rd AAAI Conf. on Artificial Intelligence (AAAI'19), 2019, pp. 1511–1519.

[8] M. Cygan, H. Dell, D. Lokshtanov, D. Marx, J. N. et al., On problems as hard as CNF-SAT, ACM Trans. Algorithms 12 (2016) 41:1–41:24. URL: https://doi.org/10.1145/2925416. doi:10.1145/2925416.

[9] T. Eiter, G. Gottlob, The complexity of logic-based abduction, J. ACM 42 (1995) 3–42. URL: https://doi.org/10.1145/200836.200838. doi:10.1145/200836.200838.

[10] T. Eiter, G. Gottlob, Propositional circumscription and extended closed-world reasoning are Πp2-complete, Theor. Comput. Sci. 114 (1993) 231–245. URL: https://doi.org/10.1016/0304-3975(93)90073-3. doi:10.1016/0304-3975(93)90073-3.

[11] G. Gottlob, Complexity results for nonmonotonic logics, J. Log. Comput. 2 (1992) 397–425. URL: https://doi.org/10.1093/logcom/2.3.397. doi:10.1093/logcom/2.3.397.

[12] S. Parsons, M. Wooldridge, L. Amgoud, Properties and complexity of some formal inter-agent dialogues, J. Log. Comput. 13 (2003) 347–376.

[13] N. Creignou, U. Egly, J. Schmidt, Complexity classifications for logic-based argumentation, ACM Trans. Comput. Log. 15 (2014) 19:1–19:20. URL: http://doi.acm.org/10.1145/2629421. doi:10.1145/2629421.

[14] G. Nordh, B. Zanuttini, What makes propositional abduction tractable, Artif. Intell. 172 (2008) 1245–1284. URL: https://doi.org/10.1016/j.artint.2008.02.001. doi:10.1016/j.artint.2008.02.001.

[15] N. Creignou, J. Schmidt, M. Thomas, Complexity classifications for propositional abduction in post's framework, J. Log. Comput. 22 (2012) 1145–1170. URL: https://doi.org/10.1093/logcom/exr012. doi:10.1093/logcom/exr012.

[16] Y. Mahmood, A. Meier, J. Schmidt, Parameterized complexity of abduction in schaefer's framework, J. Log. Comput. 31 (2021) 266–296. URL: https://doi.org/10.1093/logcom/exaa079. doi:10.1093/logcom/exaa079.

[17] Y. Mahmood, A. Meier, J. Schmidt, Parameterized complexity of logic-based argumentation in

schaefer's framework, in: Proc. 35th AAAI Conf. on Artificial Intelligence (AAAI'21), 2021, pp. 6426–6434.
[18] C. S. Peirce, C. Hartshorne, P. Weiss, The Collected Papers of Charles Sanders Peirce., Cambridge Press (1931-1958).