Learning Brave Assumption-Based Argumentation Frameworks via ASP

Emanuele De Angelis[1[0000−0002−7319−8439]],
Maurizio Proietti[1[0000−0003−3835−4931]], and
Francesca Toni[2[0000−0001−8194−1459]]

[1] IASI-CNR, Rome, Italy
emanuele.deangelis,maurizio.proietti@iasi.cnr.it
[2] Imperial College London, UK
ft@ic.ac.uk

Assumption-based Argumentation (ABA) frameworks ABA frameworks have been advocated as unifying frameworks for various forms of non-monotonic reasoning, including logic programming [BDKT97,CFST17].

Formally, an ABA framework is a tuple $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ such that:

- $\langle \mathcal{L}, \mathcal{R} \rangle$ is a deductive system, where $\mathcal{L}$ is a *language* and $\mathcal{R}$ is a set of *(inference) rules* of the form $s_0 \leftarrow s_1, \ldots, s_m$ ($m \geq 0, s_i \in \mathcal{L}$, for $1 \leq i \leq m$);
- $\mathcal{A} \subseteq \mathcal{L}$ is a (non-empty) set of *assumptions*;
- $\overline{}$ is a total mapping from $\mathcal{A}$ into $\mathcal{L}$, where $\overline{a}$ is the *contrary* of a, for $a \in \mathcal{A}$.

Elements of $\mathcal{L}$ can be any sentences, but in this paper $\mathcal{L}$ is a set of ground atoms. Also, we focus on *flat* ABA frameworks where assumptions are not heads of rules. In the spirit of logic programming, we will use variables to represent compactly all instances over some underlying universe. For instance, an ABA framework of the kind we consider may have $\mathcal{L} = \{p(X), a(X), q(X), b(X) | X \in \{1\}\}$, $\mathcal{A} = \{a(X), b(X) | X \in \{1\}\}$, with $\overline{a(X)} = q(X)$, $\overline{b(X)} = p(X)$ and $\mathcal{R} = \{p(X) \leftarrow a(X),\ \ q(X) \leftarrow b(X)\}$.

The semantics of flat ABA frameworks is given in terms of "acceptable" extensions, i.e. sets of *arguments* able to "defend" themselves against *attacks*. Intuitively, arguments are deductions of claims using rules and supported by assumptions, and attacks are directed at the assumptions in the support of arguments. An argument is represented as $S \vdash c$, where S is a set of assumptions from which the claim c can be deduced using the available rules. Argument $S_1 \vdash c_1$ attacks argument $S_2 \vdash c_2$ if c_1 is the contrary of an assumption in S_2. In the earlier example, argument $\{a(1)\} \vdash p(1)$ attacks argument $\{b(1)\} \vdash q(1)$ (and vice versa), and $\{a(1)\} \vdash a(1)$ attacks $\{b(1)\} \vdash b(1)$ (and vice versa).

Given an ABA framework $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ of the restricted form we consider, let $Args$ be the set of all arguments and $Att = \{(\alpha, \beta) \in Args \times Args \mid \alpha$ attacks $\beta\}$. Then, the notion of "acceptable" extensions we will focus on in this paper is as follows: $\Delta \subseteq Args$ is a *stable extension* iff (i) $\nexists \alpha, \beta \in \Delta$ such that $(\alpha, \beta) \in Att$ (i.e. Δ is *conflict-free*) and (ii) $\forall \beta \in Args \setminus \Delta, \exists \alpha \in \Delta$ such that $(\alpha, \beta) \in Att$ (i.e. Δ "attacks" all arguments it does not contain, thus pre-emptively "defending" itself against potential attacks). We will consider the *brave* (a.k.a. *credulous*) consequences of (flat) ABA frameworks $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$, i.e. the sets of sentences in $\mathcal{L}$ that are claims of arguments in at least one stable extension for $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$. In our earlier simple illustration, there are two stable extensions: $\{\{a(1)\} \vdash p(1), \{a(1)\} \vdash a(1)\}$ and $\{\{b(1)\} \vdash q(1), \{b(1)\} \vdash b(1)\}$. Thus, both $p(1)$ and $q(1)$ are brave consequences.

Learning Brave ABA frameworks In recent work we have presented an approach for learning ABA frameworks from background knowledge and positive and negative examples [PT23,DPT23]. In this paper we address the problem of automating the learning of ABA frameworks in the case where their semantics is defined in terms of brave consequences.

Given background knowledge $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$, positive examples $\mathcal{E}^+$ and negative examples $\mathcal{E}^-$ with $\mathcal{E}^+ \cap \mathcal{E}^- = \emptyset$, the *goal of brave ABA Learning* is to construct $\langle \mathcal{L}', \mathcal{R}', \mathcal{A}', \overline{}' \rangle$ such that $\mathcal{R} \subseteq \mathcal{R}'$, $\mathcal{A} \subseteq \mathcal{A}'$, and the following three conditions hold:
(*Existence*) $\langle \mathcal{L}', \mathcal{R}', \mathcal{A}', \overline{}' \rangle$ admits at least one stable extension Δ,
(*Completeness*) for all $e \in \mathcal{E}^+$, $S \vdash e \in \Delta$, for some set S of assumptions,
(*Consistency*) for all $e \in \mathcal{E}^-$, $S \vdash e \notin \Delta$, for any S.
$\langle \mathcal{L}', \mathcal{R}', \mathcal{A}', \overline{}' \rangle$ is called a *brave solution* of the given ABA Learning problem. A solution is *intensional* when $\mathcal{R}' \setminus \mathcal{R}$ comprises non-ground rule schemata, thus obtaining fairly general rules. As an example of a learning problem, consider a version of the Nixon-diamond example [RC81].

Background Knowledge. We only show the rules, where we use equalities for a uniformity of presentation.

$$\mathcal{R} = \{\rho_1 : \ \mathit{quaker}(X) \leftarrow X = a, \quad \rho_2 : \mathit{republican}(X) \leftarrow X = a, \\ \rho_3 : \ \mathit{quaker}(X) \leftarrow X = b, \quad \rho_4 : \mathit{republican}(X) \leftarrow X = b\}.$$

Positive Examples: $\mathcal{E}^+ = \{\mathit{pacifist}(a)\}$, *Negative Examples:* $\mathcal{E}^- = \{\mathit{pacifist}(b)\}$.

Note that this example can be seen as capturing a form of *noise*, whereby two rows of the same table (one for a and one for b) are characterised by exactly the same attributes (amounting to being quakers and republicans) but have different labels (one is pacifist, the other is not). In non-monotonic reasoning terms, this requires reasoning with contradictory rules [RC81].

Learning brave ABA frameworks via transformation rules and Answer Set Programming (ASP) In order to automate the learning of ABA frameworks under brave reasoning, we consider the approach based *transformation rules* presented in previous work [PT23]. In what follows, by means of our Nixon-diamond example, we briefly explain (some of) the transformation rules (namely, *Rote Learning*, *Folding*, and *Assumption Introduction*) and we outline an algorithm for their application, which is guided by the use of an ASP solver [GKKS12]

The set $\mathcal{R}'$ of learnt rules is initialised by adding to $\mathcal{R}$ the rule:

$$\rho_5 : \ \mathit{pacifist}(X) \leftarrow X = a$$

This is done by an application of the Rote Learning transformation, which indeed allows us to add to $\mathcal{R}$ rules corresponding to positive examples. Now, $\mathcal{R}' = \mathcal{R} \cup \{\rho_5\}$ is a brave solution of the learning problem. However, this solution is not satisfactory as rule ρ_5 is specific to the individual a, and hence it is not an intensional solution. In order to obtain an intensional solution, the learning algorithm proceeds by iterating a sequence of applications of Folding, Assumption Introduction, and Rote Learning, in this order.

First, we generalise ρ_5 by applying the Folding transformation using rule ρ_1 from the background knowledge, that is, by replacing $X = a$ in the body of ρ_5 with its consequence $quaker(X)$:

$\rho_6: \ \mathit{pacifist}(X) \leftarrow quaker(X)$

Note that other Foldings are possible, leading to different ABA frameworks. Now, $\mathcal{R}' = \mathcal{R} \cup \{\rho_6\}$ has the positive example $\mathit{pacifist}(a)$ as a brave consequence. However, every stable extension that accepts an argument for $\mathit{pacifist}(a)$, also accepts an argument for the negative example $\mathit{pacifist}(b)$. Thus, the ABA framework obtained so far is not a solution of the learning problem. In order to get an ABA framework with a stable extension that accepts $\mathit{pacifist}(a)$ and not $\mathit{pacifist}(b)$, we apply the Assumption Introduction rule and we introduce a new assumption $normal\_quaker(X)$ with contrary $\overline{normal\_quaker(X)} = abnormal\_quaker(X)$, where $abnormal\_quaker(X)$ is a new predicate symbol. Then we replace ρ_6 with:

$\rho_7: \ \mathit{pacifist}(X) \leftarrow quaker(X), normal\_quaker(X)$

and we look for suitable facts for $abnormal\_quaker(X)$ that, when added to $\mathcal{R}'$ obtain a new ABA framework with a stable extension that accepts $\mathit{pacifist}(a)$ and not $\mathit{pacifist}(b)$. This task is translated into an ASP program consisting of $\mathcal{R} \cup \{\rho_7\}$ along with the following rules:

$abnormal\_quaker(X) \leftarrow quaker(X), not\ normal\_quaker(X)$
$normal\_quaker(X) \leftarrow quaker(X), not\ abnormal\_quaker(X)$
$\leftarrow abnormal\_quaker(X),\ normal\_quaker(X)$
$\leftarrow not\ \mathit{pacifist}(a)$
$\leftarrow \mathit{pacifist}(b)$

This ASP program has an answer set that contains the atom $abnormal\_\ quaker(b)$. By Rote Learning we add the rule

$\rho_8: \ abnormal\_quaker(X) \leftarrow X = b$

and we get a new ABA framework with set of rules $\mathcal{R} \cup \{\rho_7, \rho_8\}$. Thus, the effect of Assumption Introduction and Rote Learning is to introduce an exception to the applicability of rule ρ_6. Again, the current ABA framework is a non-intensional, brave solution of our Nixon-diamond learning problem, and the algorithm proceeds with further iterations. After two iterations we replace ρ_8 with

$\rho_9: \ \ abnormal\_quaker(X) \leftarrow republican(X), normal\_republican(X)$
$\rho_{10}: \ abnormal\_republican(X) \leftarrow quaker(X), normal\_quaker(X)$

where the atom $normal\_republican(X)$ is a new assumption with contrary $\overline{normal\_republican(X)} = abnormal\_republican(X)$.

The final learnt set of rules is $\mathcal{R} \cup \{\rho_7, \rho_9, \rho_{10}\}$, which is an intensional brave solution of the Nixon-diamond learning problem. Indeed, the final ABA framework has (among others) a stable extension including the following arguments:

$\{normal\_quaker(a)\} \vdash \mathit{pacifist}(a)$,
$\{normal\_quaker(a)\} \vdash abnormal\_republican(a)$,

$\{normal\_republican(b)\} \vdash abnormal\_quaker(b),$
$\emptyset \vdash quaker(a), \emptyset \vdash republican(a), \emptyset \vdash quaker(b), \emptyset \vdash republican(b)$
Note that there are three other stable extensions of the resulting ABA framework (one where b is pacifist and a is not, one where both are pacifist and one where neither is), and thus sceptical reasoning would not work.

Related Work Some work on learning argumentation frameworks has been done, e.g. [DK95,PGNK22], but the specificity of our approach is that it learns ABA frameworks and uses ASP. Our work is also related to various techniques that learn various kinds of non-monotonic rules. Among these we mention: approaches that combine abductive and inductive learning [IH00,Ray09], non-monotonic logic programs [KY97,Sak00,SSG17], and ASP programs [LRB14,Sak05,SI09]. A formal comparison with these methods is left for future work.

References

BDKT97. A. Bondarenko, P.M. Dung, R.A. Kowalski, and F. Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artif. Intell.*, 93:63–101, 1997.

CFST17. K. Cyras, X. Fan, C. Schulz, and F. Toni. Assumption-based argumentation: Disputes, explanations, preferences. *FLAP*, 4(8), 2017.

DK95. Y. Dimopoulos and A. C. Kakas. Learning non-monotonic logic programs: Learning exceptions. In *ECML 1995*, pages 122–137, 1995.

DPT23. E. De Angelis, M. Proietti, and F. Toni. ABA learning via ASP. *EPTCS*, 385:1–8, aug 2023. Technical Communications, ICLP 2023.

GKKS12. M. Gebser, R. Kaminski, B. Kaufmann, and T. Schaub. *Answer Set Solving in Practice*. Morgan & Claypool Publishers, 2012.

IH00. K. Inoue and H. Haneda. Learning abductive and nonmonotonic logic programs. In *Abduction and Induction: Essays on their Relation and Integration*, pages 213–231. Kluwer Academic, 2000.

KY97. K.Inoue and Y.Kudoh. Learning extended logic programs. In *IJCAI*, pages 176–181. Morgan Kaufmann, 1997.

LRB14. M. Law, A. Russo, and K. Broda. Inductive learning of answer set programs. In *JELIA 2014*, LNCS 8761, pages 311–325, 2014.

PGNK22. N. Prentzas, A. Gavrielidou, M. Neofytou, and A. C. Kakas. Argumentation-based explainable machine learning (argeml): A real-life use case on gynecological cancer. In *ArgML 2022*, CEUR-WP, 3208, pages 1–13, 2022.

PT23. M. Proietti and F. Toni. Learning assumption-based argumentation frameworks. *CoRR*, 2023. To appear in Proc. ILP '22, LNCS, Springer.

Ray09. O. Ray. Nonmonotonic abductive inductive learning. *J. Appl. Log.*, 7(3):329–340, 2009.

RC81. R. Reiter and G. Criscuolo. On interacting defaults. In *IJCAI*, pages 270–276. William Kaufmann, 1981.

Sak00. C. Sakama. Inverse entailment in nonmonotonic logic programs. In *Proc. ILP 2000*, 2000.

Sak05. C. Sakama. Induction from answer sets in nonmonotonic logic programs. *ACM TOCL*, 6(2):203–231, 2005.

SI09. C. Sakama and K. Inoue. Brave induction: A logical framework for learning from incomplete information. *Mach. Learn.*, 76(1):3–35, 2009.

SSG17. F. Shakerin, E. Salazar, and G. Gupta. A new algorithm to automate inductive learning of default theories. *TPLP*, 17(5-6):1010–1026, 2017.