MS-DIS: Multi-Shot ASP-Driven ABA Disputes*

Martin Diller
Logic Programming and Argumentation Group
TU Dresden, Germany
martin.diller@tu-dresden.de

Piotr Gorczyca
Computational Logic Group
TU Dresden, Germany
piotr.gorczyca@tu-dresden.de

***Abstract*—We describe MS-DIS, a system implementing dispute derivations for assumption-based argumentation (ABA) via multi-shot answer set programming (ASP). It participates in the ICCMA 2025 ABA tracks for credulous acceptance under the complete and stable semantics. While primarily designed to support dialectical explication rather than performance, MS-DIS also showcases the use of the process of argumentation itself as the core reasoning mechanism.**

I. Introduction

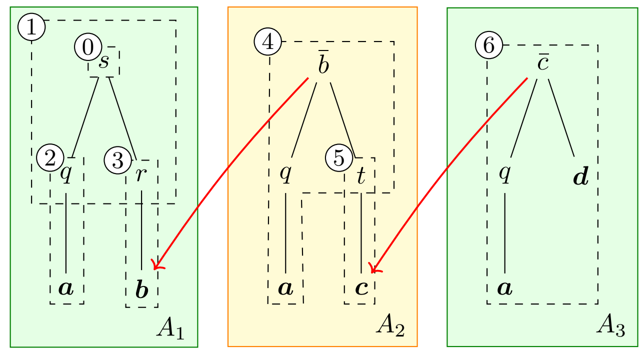
Dispute derivations are among the primary native reasoning methods for assumption-based argumentation (ABA) [1]. Inspired by games for abstract argumentation [2], they are conceptualized as structured disputes in which arguments supporting or challenging a given claim are exchanged between a proponent and an opponent. Beyond serving as a reasoning mechanism for determining the acceptability of claims, dispute derivations also offer a framework for dialectical explication.

MS-DIS implements the rule-based representation of (flexible) ABA disputes first introduced in [3]. This version of ABA disputes further simplifies the earlier graph-based dispute model from [4]—itself based on [5]—by having the proponent and opponent exchange rules, rather than arguments. While arguments are not represented explicitly, they can be reconstructed from the rule-based exchange.

The implementation makes use of multi-shot answer set programming (ASP) using the `clingo` solver [6]. This enables the incremental grounding and solving of logic programs across multiple iterations, while reducing some of the overhead typically associated with repeated invocations of an ASP solver.

In preliminary experiments, we found that although MS-DIS —unsurprisingly, given its focus on explication—still clearly lags behind inference-focused systems such as `aspforaba` [7], it outperforms `flexABle`, a direct implementation of rule-based ABA disputes. Notably, `flexABle` was the only dispute-based system to participate in the most recent ICCMA competition (2023), and has been shown to be more efficient than earlier Prolog-based implementations of ABA disputes [8]. Nevertheless, as with our decision to submit `flexABle` to ICCMA'23, we submitted MS-DIS to ICCMA'25 not primarily for its efficiency, but because we consider it valuable to also include systems that use the process of argumentation itself as the basis for reasoning.

The authors are funded by the German Federal Ministry of Research, Technology and Space (BMFTR) via SEMECO (grant no. 03ZU1210B) and KIMEDS (grant no. GW0552B) respectively.



m(1,**p**,pb1,$s \leftarrow q, r$) m(2,**p**,pb1,$q \leftarrow a$) m(3,**p**,pb1,$r \leftarrow b$)
m(4,**o**,ob2,$\bar{b} \leftarrow q, t$) m(5,**o**,ob1,$t \leftarrow c$) m(6,**p**,pb2,$\bar{c} \leftarrow q, d$)

Fig. 1. Visualisation of a dispute as produced by MS-DIS. The output of MS-DIS is shown at the bottom, while the visualisation is at the top. In the top figure green and yellow boxes indicate proponent and opponent arguments, respectively. Rules appear as subtrees, with heads as roots and bodies as leaves, connected by solid lines. Red arrows mark attacks, assumptions are in boldface, and dashed boxes with numbers indicate the order of rule application in the dispute.

II. An example

Consider the ABA framework $\mathcal{F} = (\mathcal{L}, \mathcal{A}, \overline{}, \mathcal{R})$, with assumptions $\mathcal{A} = \{a, b, c, d\}$, a contrary relation defined by $\overline{}(x) = \{\bar{x}\}$ for all $x \in \mathcal{A}$, and the set of rules $\mathcal{R} = \{s \leftarrow q, r; q \leftarrow a;\ r \leftarrow b;\ r \leftarrow c;\ \bar{b} \leftarrow a, t; q \leftarrow a; t \leftarrow c; \bar{c} \leftarrow q, d\}$. A dispute demonstrating the goal statement s to be acceptable with respect to the complete semantics is shown in Figure 1. The top portion of the figure illustrates the arguments implicitly constructed during the dispute, while the bottom part displays the output of MS-DIS at each step.

Each entry in the output has the form `m(S, P, T, R)`, indicating that at step `S` of the dispute, player `P` performed a move of type `T`, involving a rule `R`. Players are either the proponent (`p`) or the opponent (`o`). All moves in this example are backward moves, meaning the player either justifies a previously asserted claim by introducing a rule (types `pb1` or `ob1`), or initiates an attack against a statement introduced by the opposing player (types `pb2` or `ob2`).

III. System description, reasoning tasks

MS-DIS consists of (multi-shot) ASP encodings that closely follow the definition of rule-based ABA disputes from [3] to compute possible moves at each step. The second component is a simple Python script that manages multi-shot execution. MS-DIS supports both an interactive mode and an automatic mode, the latter being used in the ICCMA competition. Two calls to the ASP solver are needed at each step of a dispute: one to check whether the proponent can make a move that wins the dispute, while the second is to check whether there is a move where the opponent does *not* win. If the first check fails and the second succeeds, the dispute continues.

MS-DIS participates in the ABA subtracks addressing the credulous acceptance problem for the admissible and stable semantics. Since credulous acceptance under complete semantics is equivalent to credulous acceptance under the admissible semantics, the simpler encoding for the admissible semantics is used for the complete semantics. The encoding for stable semantics is a slight extension of that for the admissible semantics.

In contrast to flexABle [8], MS-DIS currently implements only a basic reasoning strategy. Specifically, the version submitted to ICCMA employs a strategy that prioritizes moves which do not involve choices in the search for a winning dispute for the proponent over those that introduce branching.

References

[1] K. Cyras, X. Fan, C. Schulz, and F. Toni, "Assumption-based argumentation: Disputes, explanations, preferences," in *Handbook of Formal Argumentation*, P. Baroni, D. Gabbay, and M. Giacomin, Eds., 2018, pp. 365–408.

[2] M. Caminada, "Argumentation semantics as formal discussion," in *Handbook of Formal Argumentation*, P. Baroni, D. Gabbay, and M. Giacomin, Eds., 2018, pp. 487–518.

[3] M. Diller, S. A. Gaggl, and P. Gorczyca, "Flexible dispute derivations with forward and backward arguments for assumption-based argumentation," in *CLAR*, ser. LNCS, vol. 13040, 2021, pp. 147–168.

[4] R. Craven and F. Toni, "Argument graphs and assumption-based argumentation," *Artif. Intell.*, vol. 233, pp. 1–59, 2016.

[5] F. Toni, "A generalised framework for dispute derivations in assumption-based argumentation," *Artif. Intell.*, vol. 195, pp. 1–43, 2013.

[6] M. Gebser, R. Kaminski, B. Kaufmann, and T. Schaub, "Multi-shot ASP solving with clingo," *Theory Pract. Log. Program.*, vol. 19, no. 1, pp. 27–82, 2019.

[7] T. Lehtonen, J. P. Wallner, and M. Järvisalo, "Declarative algorithms and complexity results for assumption-based argumentation," *J. Artif. Intell. Res.*, vol. 71, pp. 265–318, 2021.

[8] M. Diller, S. A. Gaggl, and P. Gorczyca, "Strategies in flexible dispute derivations for assumption-based argumentation," in *SAFA@COMMA*, ser. CEUR Workshop Proceedings, vol. 3236, 2022, pp. 59–72.