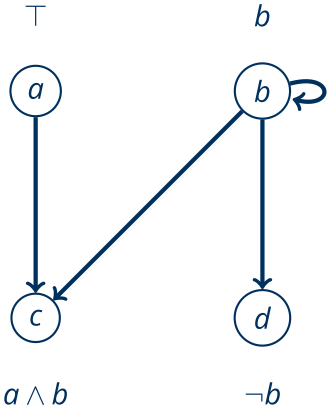
Stefan Ellmauthaler, Lukas Gerlach

Faculty of Computer Science, International Center for Computational Logic, Knowledge-Based Systems Group

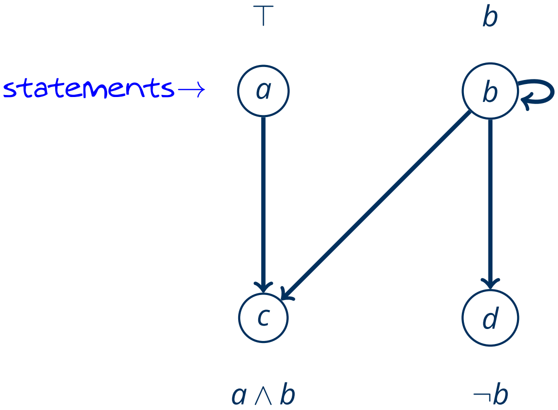
ADF-BDD.DEV: Debug Abstract Dialectical Frameworks with Binary Decision Diagrams

XLoKR 2023 // Rhodes, Greece, September 2, 2023

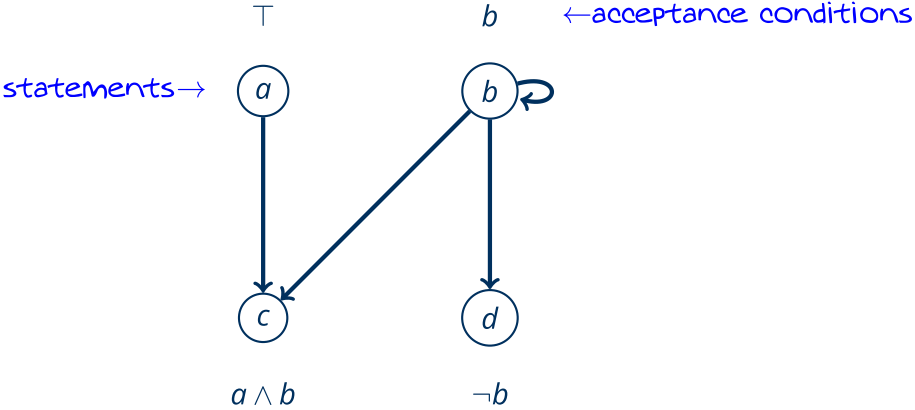
Abstract Dialectical Frameworks [BES⁺18]



Abstract Dialectical Frameworks [BES^{+}18]



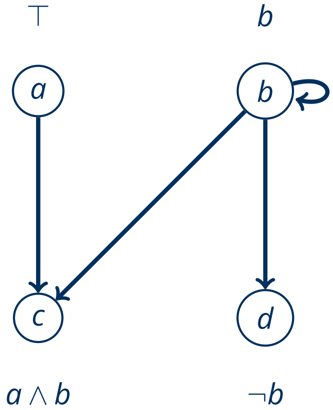
Abstract Dialectical Frameworks [BES$^+$18]



statements→

←acceptance conditions

Abstract Dialectical Frameworks [BES$^+$18]



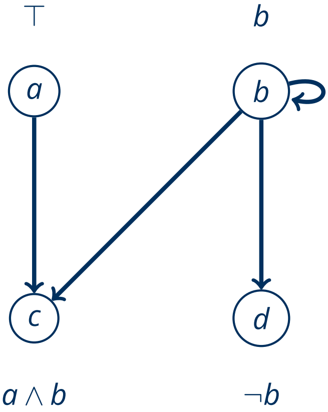
←acceptance conditions

statements→

semantics: $\Gamma_D : \mathcal{V}^3 \mapsto \mathcal{V}^3$

Abstract Dialectical Frameworks [BES+18]

statements→



←acceptance conditions

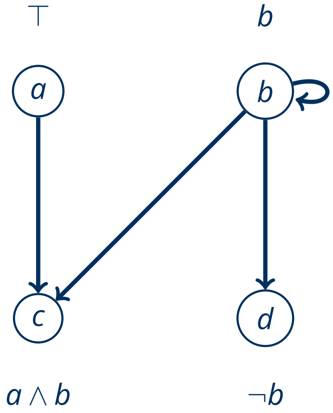
$a \mapsto \boldsymbol{u}, b \mapsto \boldsymbol{u}, c \mapsto \boldsymbol{u}, d \mapsto \boldsymbol{u}$

semantics: $\Gamma_D : \mathcal{V}^3 \mapsto \mathcal{V}^3$

Abstract Dialectical Frameworks [BES+18]

←acceptance conditions

statements→



$a \mapsto \boldsymbol{u}, b \mapsto \boldsymbol{u}, c \mapsto \boldsymbol{u}, d \mapsto \boldsymbol{u}$

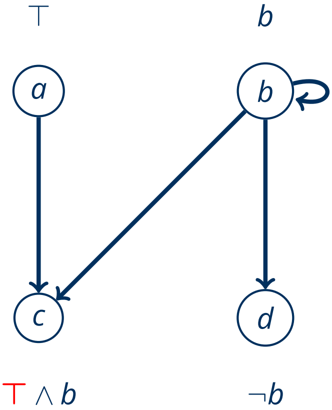
$a \mapsto \boldsymbol{t}, b \mapsto \boldsymbol{u}, c \mapsto \boldsymbol{u}, d \mapsto \boldsymbol{u}$

semantics: $\Gamma_D : \mathcal{V}^3 \mapsto \mathcal{V}^3$

Abstract Dialectical Frameworks [BES$^+$18]

statements→

←acceptance conditions



$a \mapsto \boldsymbol{u}, b \mapsto \boldsymbol{u}, c \mapsto \boldsymbol{u}, d \mapsto \boldsymbol{u}$

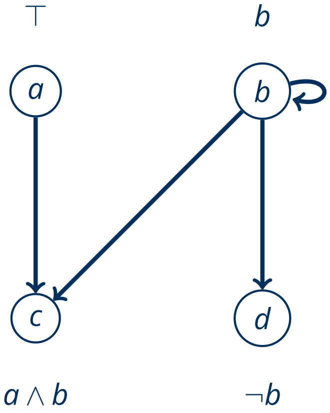
$a \mapsto \boldsymbol{t}, b \mapsto \boldsymbol{u}, c \mapsto \boldsymbol{u}, d \mapsto \boldsymbol{u}$

semantics: $\Gamma_D : \mathcal{V}^3 \mapsto \mathcal{V}^3$

Abstract Dialectical Frameworks [BES+18]

←acceptance conditions

statements→



grounded, complete

$a \mapsto \boldsymbol{t}, b \mapsto \boldsymbol{u}, c \mapsto \boldsymbol{u}, d \mapsto \boldsymbol{u}$

complete

$a \mapsto \boldsymbol{t}, b \mapsto \boldsymbol{t}, c \mapsto \boldsymbol{t}, d \mapsto \boldsymbol{f}$

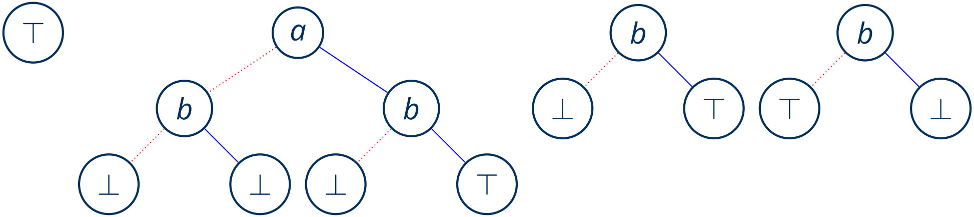
complete, stable

$a \mapsto \boldsymbol{t}, b \mapsto \boldsymbol{f}, c \mapsto \boldsymbol{f}, d \mapsto \boldsymbol{t}$

semantics: $\Gamma_D : \mathcal{V}^3 \mapsto \mathcal{V}^3$

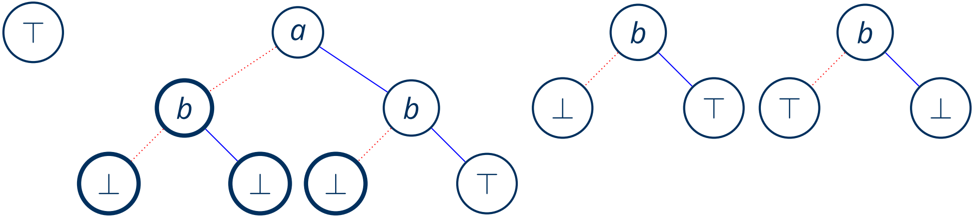
ordered Binary Decision Tree

- Tree: inner nodes are variables and leafs are truth constants $\top$ and $\bot$
- Inner node has lo and hi child
- Every path from root to leaf needs to follow pre-defined strict ordering of variables



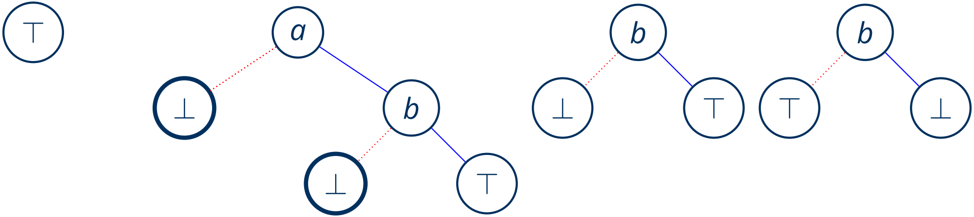
reduced ordered Binary Decision Diagram [Bry86]

- $lo(n) = hi(n)$, then replace n by $hi(n)$
- if $n = v$, then replace v by n globally (violate tree-property)
- Given a variable order, this representation is unique under logical equivalence of formulae



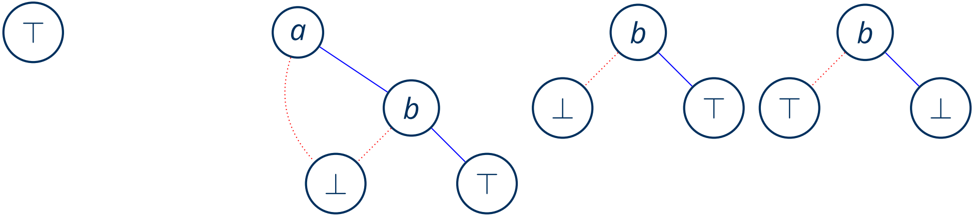
reduced ordered Binary Decision Diagram [Bry86]

- $lo(n) = hi(n)$, then replace n by $hi(n)$
- if $n = v$, then replace v by n globally (violate tree-property)
- Given a variable order, this representation is unique under logical equivalence of formulae



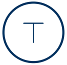
reduced ordered Binary Decision Diagram [Bry86]

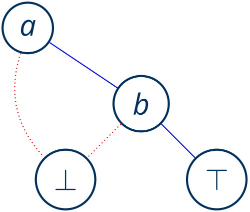
- $lo(n) = hi(n)$, then replace n by $hi(n)$
- if $n = v$, then replace v by n globally (violate tree-property)
- Given a variable order, this representation is unique under logical equivalence of formulae

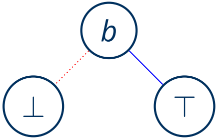


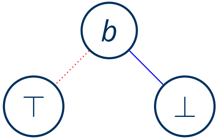
reduced ordered Binary Decision Diagram [Bry86]

- $lo(n) = hi(n)$, then replace n by $hi(n)$
- if $n = v$, then replace v by n globally (violate tree-property)
- Given a variable order, this representation is unique under logical equivalence of formulae







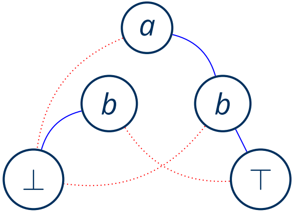


- Restriction linear

- Optimal variable order in NP
- Check for (un-)SAT and TAUT constant [DM02]

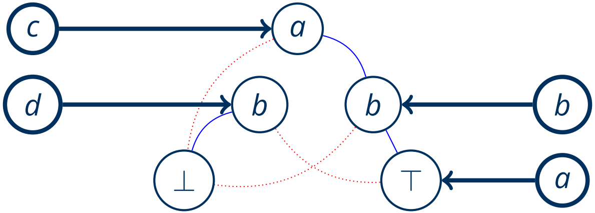
New idea: roBDDs to represent ADFs [EGRW22b]

- To each statement, one BDD is related as the acceptance condition
- More compact representation due to "merging" of nodes



New idea: roBDDs to represent ADFs [EGRW22b]

- To each statement, one BDD is related as the acceptance condition
- More compact representation due to "merging" of nodes



roBDDs to represent ADFs

Theorem

Given the BDD representation of an ADF D, the result of applying Γ_D to any three-valued interpretation $\mathcal{I}$ can be computed in polynomial time.

Theorem

Given an ADF D in BDD representation, there is a polynomial algorithm that computes the grounded interpretation of D.

Corollary

Verifying whether a three-valued interpretation is a model or is stable in an ADF represented by BDDs is in P. Moreover credulous reasoning is in NP and sceptical reasoning in coNP.

ADF-BDD solver [EGRW22a]

- Written in `Rust`
- BDDs
  - own implementation
  - biodivine-bdd for faster instantiation
- Various BDD-modes (own, biodivine, hybrid)
- Grounded, complete, and stable semantics
- Github, Library, and Binary available
  - hub: `https://github.com/ellmau/adf-obdd`
  - lib: `https://crates.io/crates/adf-bdd`
  - bin: `https://crates.io/crates/adf-bdd-bin`

ADF-BDD.dev

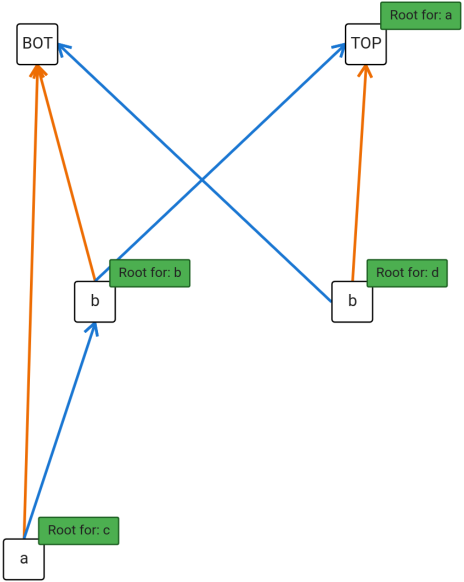
- Web-application for adf-bdd
- Visualisation of results as BDDs
  - Parsed state
  - Every semantics model
- G6 graph visualisation library used [WBL^{+}21]
  - Dagre algorithm for graph representation
  - Ranks nodes into hierarchy
  - Minimises number of crossing edges
- Colour-coded BDD
  - Orange is lo
  - Blue is hi
  - Green labels are the roots for each statement

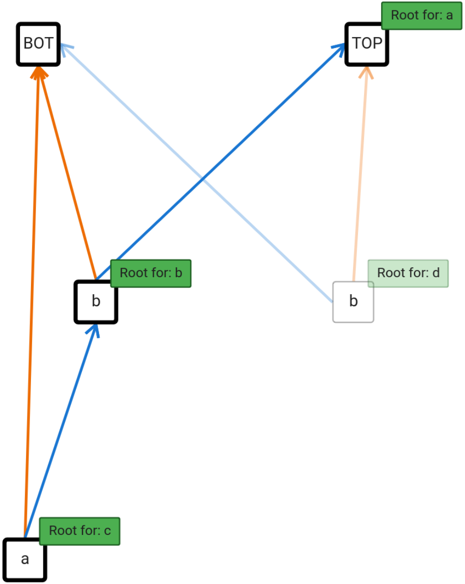
ADF-BDD.dev
Visualisation Insights

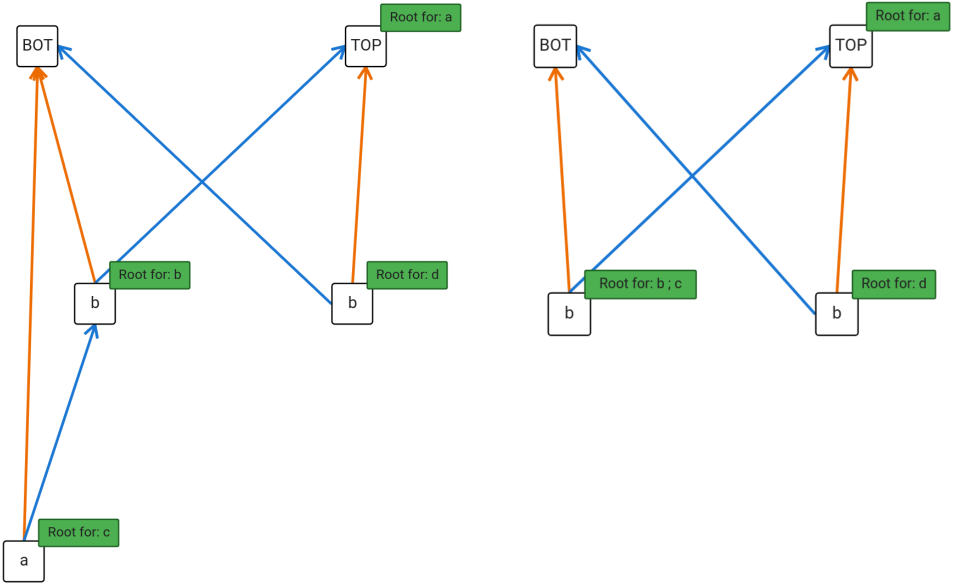
- ***t*** and ***f*** already decided by fixpoints
- Reason for each ***u*** is given by corresponding BDD
- Focus onto sub-diagrams
- No redundant links, only variables that have an impact

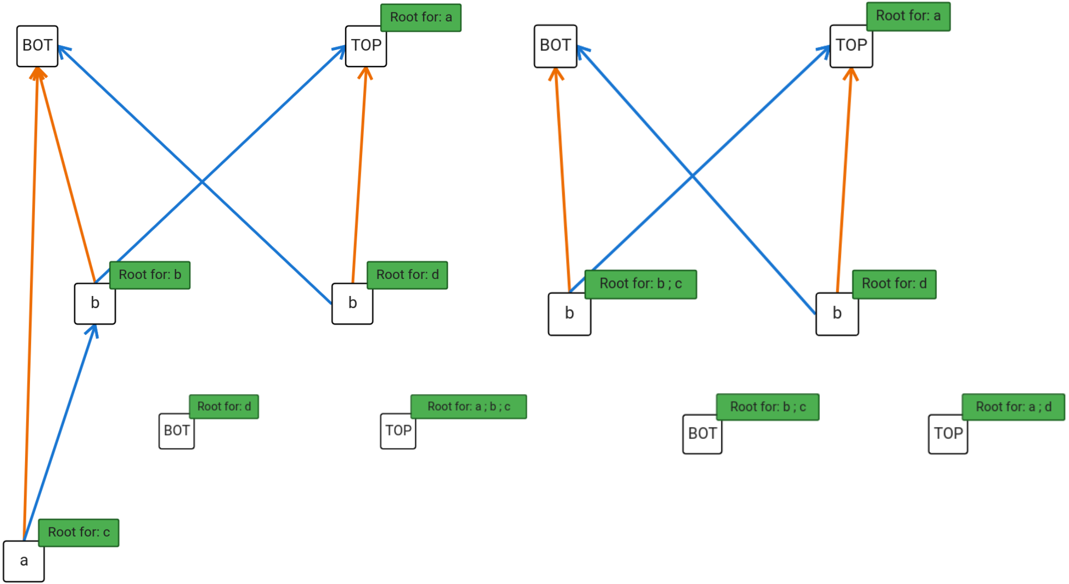
ADF-BDD.dev

Online presentation of ADF-BDD.dev or further slides


Root for: a
BOT
TOP
Root for: b
Root for: d
b
b
Root for: c
a


Root for: a
BOT
TOP
Root for: b
Root for: d
b
b
Root for: c
a


BOT
TOP
Root for: a
b
Root for: b
b
Root for: d
a
Root for: c
BOT
TOP
Root for: a
b
Root for: b ; c
b
Root for: d


Root for: a
BOT
TOP
Root for: b
Root for: d
b
b
Root for: d
BOT
Root for: a ; b ; c
TOP
Root for: c
a
Root for: a
BOT
TOP
Root for: b ; c
Root for: d
b
b
Root for: b ; c
BOT
Root for: a ; d
TOP

Future Work

- Direct editing of BDDs
  - ADF-Design
  - Enforcement
- Import and export of BDDs
- More assistance for non-familiar users
- More analysis options for experienced users

Thank you for your interest!

Visit ADF-BDD at

https://adf-bdd.dev

https://ellmau.github.io/adf-obdd/

Abstract Dialectical Frameworks

Definition (Abstract Dialectical Framework (ADF))

An ADF is a tuple $\langle S, C\rangle$, where

- S is a fixed finite set of statements and
- $C := \{\varphi_s\}_{s\in S}$ is a set acceptance conditions for statements, which corresponds to propositional formulae whose variable signature is S.

Abstract Dialectical Frameworks

Definition (Γ_D)

Let $D = \langle S, C\rangle$ be an ADF, $\mathcal{I} : S \mapsto \{\boldsymbol{t}, \boldsymbol{f}, \boldsymbol{u}\}$ be a three-valued interpretation, and $\Gamma_D(\mathcal{I}) : S \mapsto \{\boldsymbol{t}, \boldsymbol{f}, \boldsymbol{u}\}$ with

$$s = \begin{cases} \mathbf{t} & \text{if } \models \varphi_s(\mathcal{I}); \\ \mathbf{f} & \text{if } \varphi_s(\mathcal{I}) \models \bot; \\ \mathbf{u} & \text{otherwise.} \end{cases}$$

Definition (Semantics)

$\mathcal{I}$ is

- complete if $\mathcal{I} = \Gamma_D(\mathcal{I})$
- grounded if $\mathcal{I} = \mathit{lfp}(\Gamma_D)$

Abstract Dialectical Frameworks

Definition (Semantics)

$\mathcal{I}$ is

- complete if $\mathcal{I} = \Gamma_D(\mathcal{I})$
- grounded if $\mathcal{I} = \mathit{lfp}(\Gamma_D)$
- stable if $\mathcal{I}$ is two-valued, complete, and the for the grounded interpretation $\mathcal{W}$ of $D^{\mathcal{I}}$ it holds that $\mathcal{I}(s) = \boldsymbol{t}$ implies $\mathcal{W}(s) = \boldsymbol{t}$

Abstract Dialectical Frameworks

Definition (Semantics)

$\mathcal{I}$ is

- complete if $\mathcal{I} = \Gamma_D(\mathcal{I})$
- grounded if $\mathcal{I} = \mathit{lfp}(\Gamma_D)$
- stable if $\mathcal{I}$ is two-valued, complete, and the for the grounded interpretation $\mathcal{W}$ of $D^{\mathcal{I}}$ it holds that $\mathcal{I}(s) = \boldsymbol{t}$ implies $\mathcal{W}(s) = \boldsymbol{t}$

Definition (Reduction)

Let $D = \langle S, C \rangle$ be an ADF, $\mathcal{I} : S \mapsto \{\boldsymbol{t}, \boldsymbol{f}\}$ be a two-valued interpretation.
$D^{\mathcal{I}} = \langle S^{\mathcal{I}}, C^{\mathcal{I}} \rangle$, where

- $S^{\mathcal{I}} = \{s \in S \mid \mathcal{I}(s) = \boldsymbol{t}\}$
- $C^{\mathcal{I}} = \{\varphi_s[s'/\bot : \mathcal{I}(s') = \boldsymbol{f}]\}$

References I

Gerhard Brewka, Stefan Ellmauthaler, Hannes Strass, Johannes P. Wallner, and Stefan Woltran.
Abstract dialectical frameworks.
In Pietro Baroni, Dov Gabbay, Massimiliano Giacomin, and Leendert van der Torre, editors, *Handbook of Formal Argumentation*, chapter 5, pages 237–285. College Publications, 2018.

Randal E Bryant.
Graph-based algorithms for boolean function manipulation.
IEEE Trans. Computers, 100(8):677–691, 1986.

Adnan Darwiche and Pierre Marquis.
A knowledge compilation map.
J. Artif. Intell. Res., 17:229–264, 2002.

References II

- Stefan Ellmauthaler, Sarah Alice Gaggl, Dominik Rusovac, and Johannes P. Wallner.
  Adf - BDD : An ADF solver based on binary decision diagrams.
  In Francesca Toni, editor, *Proceedings of the 9th International Conference on Computational Models of Argument (COMMA 2022)*, volume 220146 of *FAIA*, pages 355–356. IOS Press, September 2022.

- Stefan Ellmauthaler, Sarah Alice Gaggl, Dominik Rusovac, and Johannes P. Wallner.
  Representing abstract dialectical frameworks with binary decision diagrams.
  In G. Gottlob, D Inclezan, and M. Maratea, editors, *Proceedings of the 16th International Conference on Logic Programming and Non-monotonic*

References III

Reasoning (LPNMR 2022), volume 13416 of *Lecture Notes in Computer Science*, pages 177–198. Springer, 2022.

Yanyan Wang, Zhanning Bai, Zhifeng Lin, Xiaoqing Dong, Yingchaojie Feng, Jiacheng Pan, and Wei Chen.
G6: A web-based library for graph visualization.
Visual Informatics, 5(4):49–55, December 2021.