ScaDS.AI
DRESDEN LEIPZIG

CENTER FOR SCALABLE DATA ANALYTICS AND ARTIFICIAL INTELLIGENCE

Abstract Dialectical Frameworks meet *Binary Decision Diagrams*

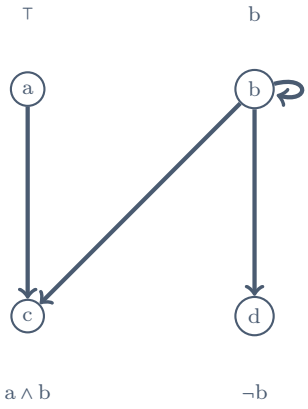
LPNMR [1], COMMA [2], and current research

Stefan Ellmauthaler

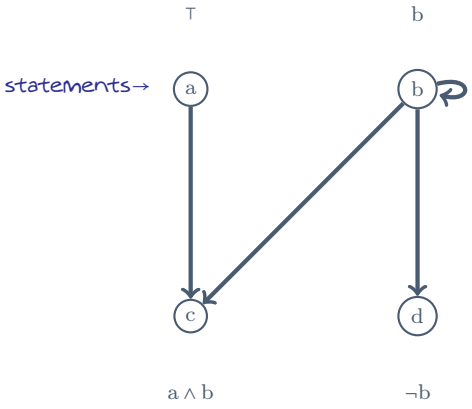
Dresden, May 2nd 2023

GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung
SACHSEN
Diese Maßnahme wird gefördert durch die Bundesregierung aufgrund eines Beschlusses des Deutschen Bundestages. Diese Maßnahme wird mitfinanziert durch Steuermittel auf der Grundlage des von den Abgeordneten des Sächsischen Landtags beschlossenen Haushaltes.

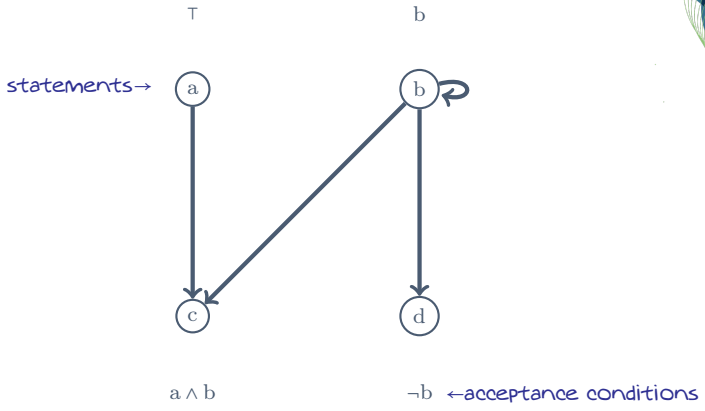
01 Abstract Dialectical Frameworks [3]



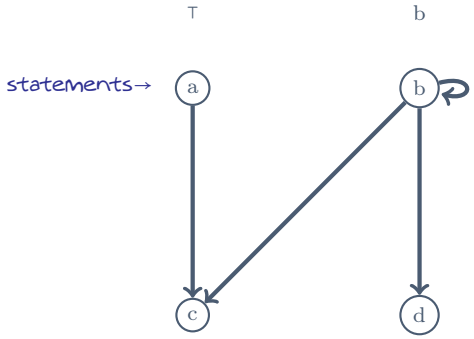
01 Abstract Dialectical Frameworks [3]



01 Abstract Dialectical Frameworks [3]

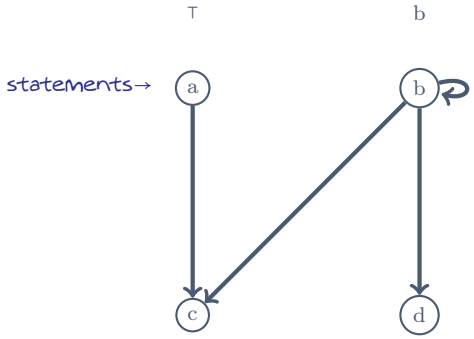


01 Abstract Dialectical Frameworks [3]

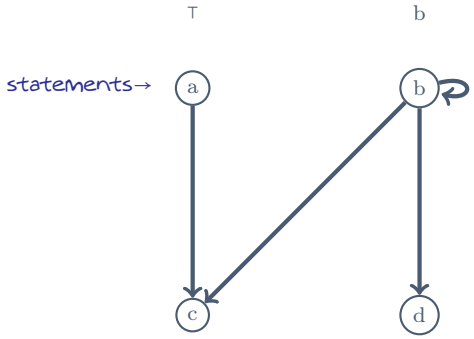


semantics: $\Gamma_D : \mathcal{V}^3 \mapsto \mathcal{V}^3$

01 Abstract Dialectical Frameworks [3]



01 Abstract Dialectical Frameworks [3]

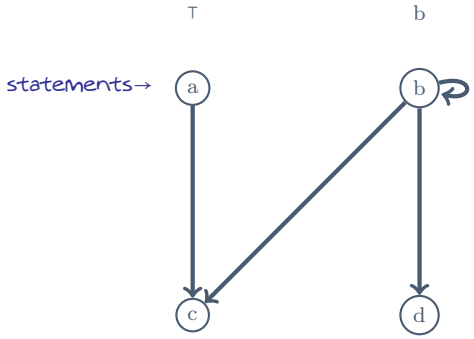


$a \mapsto u, b \mapsto u, c \mapsto u, d \mapsto u$

$a \mapsto t, b \mapsto u, c \mapsto u, d \mapsto u$

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

01 Abstract Dialectical Frameworks [3]



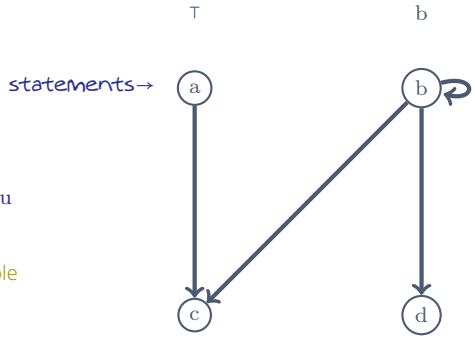
$a \mapsto u, b \mapsto u, c \mapsto u, d \mapsto u$

$a \mapsto t, b \mapsto u, c \mapsto u, d \mapsto u$

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

$\top \wedge b$ $\neg b$ ←acceptance conditions

01 Abstract Dialectical Frameworks [3]

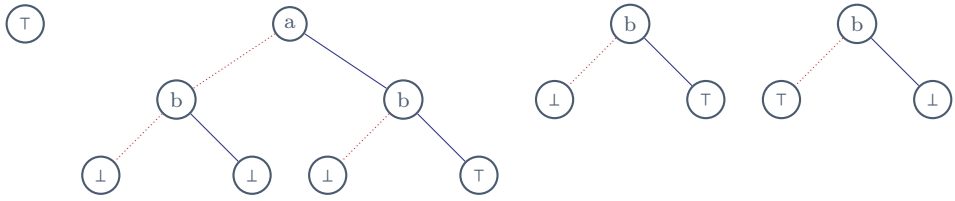


grounded, complete
$a \mapsto t, b \mapsto u, c \mapsto u, d \mapsto u$
complete
$a \mapsto t, b \mapsto t, c \mapsto t, d \mapsto f$
complete, stable
$a \mapsto t, b \mapsto f, c \mapsto f, d \mapsto t$

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

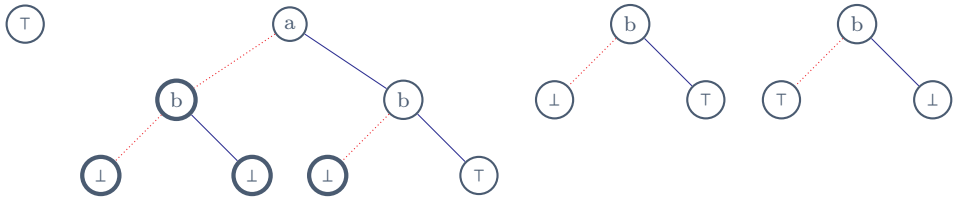
02 ordered Binary Decision Tree

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



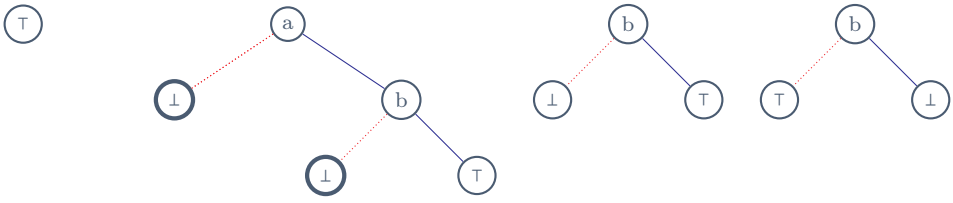
02 reduced ordered Binary Decision Diagram [4]

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



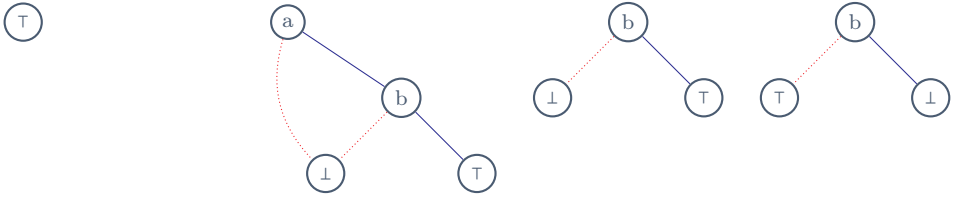
02 reduced ordered Binary Decision Diagram [4]

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



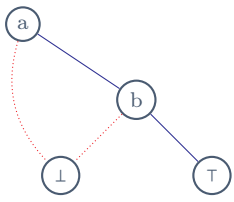
02 reduced ordered Binary Decision Diagram [4]

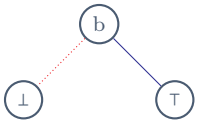
- $\mathrm{lo}(\mathrm{n}) = \mathrm{hi}(\mathrm{n})$, then replace n by $\mathrm{hi}(\mathrm{n})$
- if $\mathrm{n} = \mathrm{v}$, then replace v by n globally (violate tree-property)
- Given a variable order, this representation is unique under logical equivalence of formulae

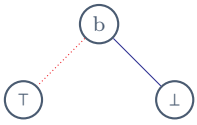


02 reduced ordered Binary Decision Diagram [4]

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





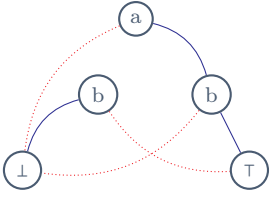


- Restriction linear

- Optimal variable order in NP
- Check for (un-)SAT and TAUT constant [5]

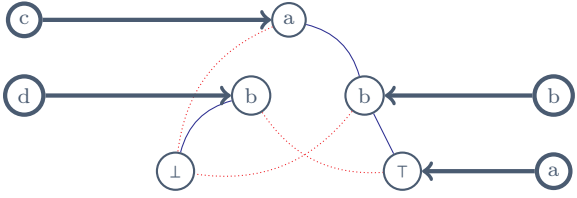
03 New idea: roBDDs to represent ADFs

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



03 New idea: roBDDs to represent ADFs

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



03 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.

04 ADF-BDD solver

- 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`

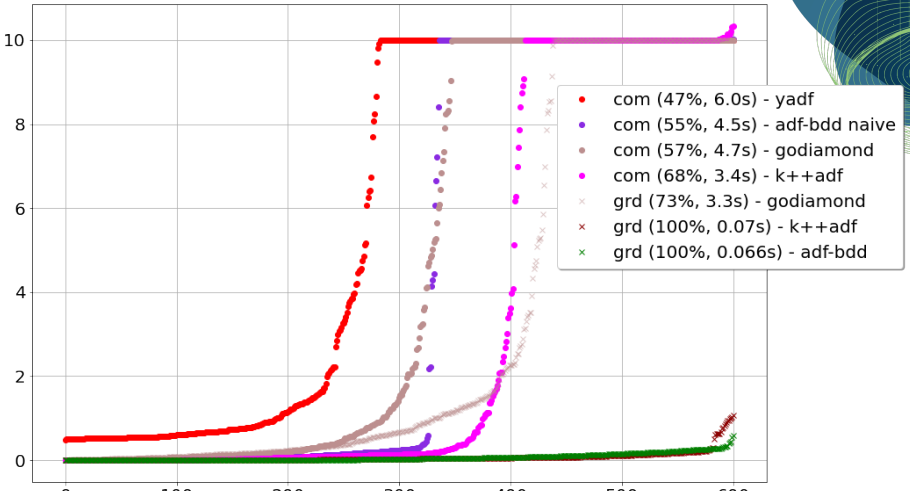
04 ADF-BDD solver

- 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`

Evaluation

- `goDIAMOND`, `k++adf`, `yadf`
- 600 instances
- timeout 10 seconds
- `hyperfine` evaluation

04 ADF-BDD Evaluation: grounded and complete



04 Search Space Exploitation
with Faceted Navigation

- Use Faceted Navigation measures to describe (Sub-)Search-Space
- Allows for an easy framework of properties for heuristics, like
  - Number of Models
  - Number of Facets
  - BDD Paths to $\top$ resp. $\bot$
  - Variable impact
  - …
- Heuristics and Facet Navigation-based Algorithm for Stable Models
  - Recursive, one set of NoGood-like constraints per recursion path
  - Based on a heuristic, identify the optimal facet to activate
  - Propagates truth values, based on the facets and construct fixpoints
  - Explore activated facet recursively
  - Add the inverse facet to the NoGoods and continue the recursion

04 Search Space Exploitation
with Faceted Navigation

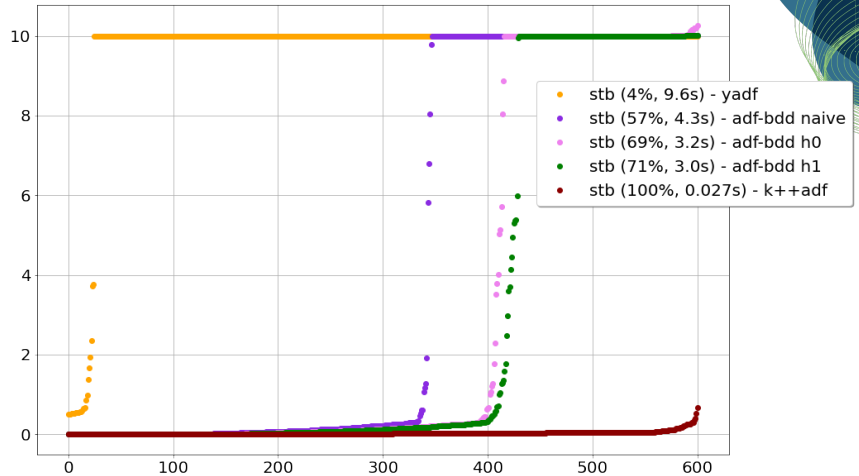
- Use Faceted Navigation measures to describe (Sub-)Search-Space
- Allows for an easy framework of properties for heuristics, like
  - Number of Models
  - Number of Facets
  - BDD Paths to $\top$ resp. $\bot$
  - Variable impact
  - …

- high impact
- min variables
- min paths

- min paths
- high impact

- Heuristics and Facet Navigation-based Algorithm for Stable Models
  - Recursive, one set of NoGood-like constraints per recursion path
  - Based on a heuristic, identify the optimal facet to activate
  - Propagates truth values, based on the facets and construct fixpoints
  - Explore activated facet recursively
  - Add the inverse facet to the NoGoods and continue the recursion

04 ADF-BDD Evaluation: stable



04 Further Improvements after the Evaluation

- Implemented two-valued NoGood-reasoning for Stable Models
- Launched Web-Service `https://adf-bdd.dev`
  - Visualisation of BDD structures and results
  - Managing own problem-sets
  - Easy to use
  - A new approach to Explainable Result
  - Submitted to ICLP 23 Systems Demonstration Track

04 Contributions

ADF with BDD

- Use BDDs to represent ACs of ADFs
- Complexity analysis of this BDD-based ADFs
  - Drop of complexity - one level on polynomial hierarchy
  - Same as (easier) Dung AF and (less expressive) BADFs
- Unique representation of maximal information with respect to Γ_{D}

Facet Navigation for Search Space Exploitation

- Use Facet Navigation to navigate in the search space
- Represent Properties, Weights, and Heuristics in an uniform model

04 Contributions II

Software Tool

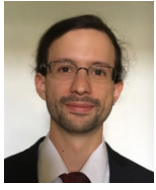
- Comparable to fastest solver for grounded semantics
- Comparable to 2nd fastest solver for complete semantics with a naive approach
- In between fastest and 2nd fastest for stable semantics with improved performance with faceted heuristics algorithm
- Easy usability through `Rust`-Ecosystem
  (e.g. `cargo install adf-bdd-bin` to try it out, lib-crate to use ADF-BDD in your own project,…)
- Visualisation and Insights into the computation via the web-application `adf-bdd.dev`

04 Future Work

- Implement full three-valued NoGood-reasoning
- Implement further Heuristics
- Improve the BDD methods
- Investigate optimal BDD-variable-orders
- Improve on UX and Visualisation

04 Further Acknowledgements

Sarah A. Gaggl
Lukas Gerlach
Dominik Rusovac
Johannes P. Wallner



CENTER FOR PERSPICUOUS COMPUTING

INNOSALE

TU Graz

05 References I

[1] Stefan Ellmauthaler et al. „Representing Abstract Dialectical Frameworks with Binary Decision Diagrams“. In: *Proceedings of the 16th International Conference on Logic Programming and Non-monotonic Reasoning (LPNMR 2022)*. Ed. by G. Gottlob, D Inclezan, and M. Maratea. Vol. 13416. Lecture Notes in Computer Science. Springer, 2022, pp. 177–198. DOI: 10.1007/978-3-031-15707-3_14.

[2] Stefan Ellmauthaler et al. „ADF - BDD : An ADF Solver Based on Binary Decision Diagrams“. In: *Proceedings of the 9th International Conference on Computational Models of Argument (COMMA 2022)*. Ed. by Francesca Toni. Vol. 220146. FAIA. IOS Press, Sept. 2022, pp. 355–356. DOI: 10.3233/FAIA220170.

[3] Gerhard Brewka et al. „Abstract Dialectical Frameworks“. In: *Handbook of Formal Argumentation*. Ed. by Pietro Baroni et al. College Publications, 2018. Chap. 5, pp. 237–285.

[4] Randal E Bryant. „Graph-based algorithms for boolean function manipulation“. In: *IEEE Trans. Computers* 100.8 (1986), pp. 677–691.

[5] Adnan Darwiche and Pierre Marquis. „A Knowledge Compilation Map“. In: *J. Artif. Intell. Res.* 17 (2002), pp. 229–264.