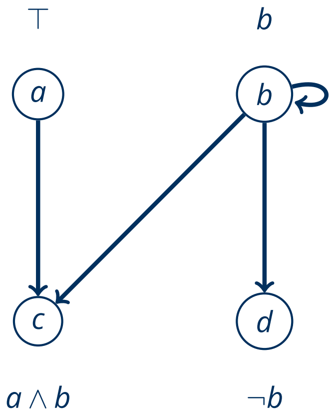
Stefan Ellmauthaler, Sarah A. Gaggl, Dominik Rusovac, Johannes P. Wallner

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

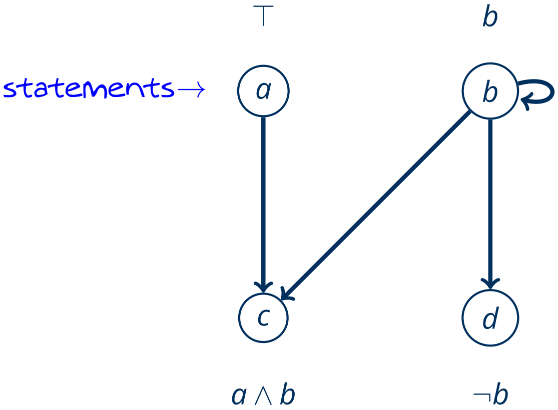
Representing *Abstract Dialectical Frameworks* with *Binary Decision Diagrams*

LPNMR 2022 // Genova Nervi, Italy, September 8, 2022

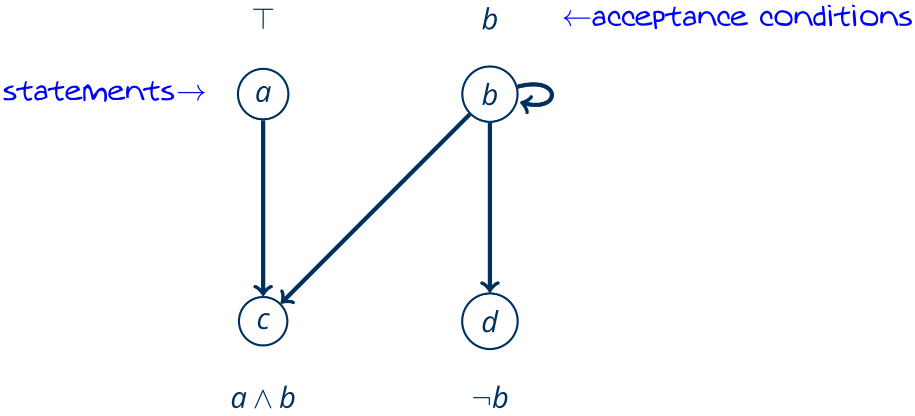
Abstract Dialectical Frameworks [BES+18]



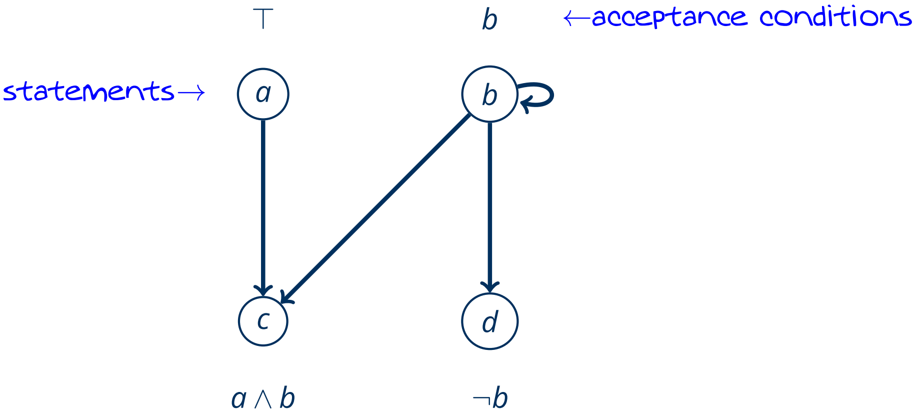
Abstract Dialectical Frameworks [BES⁺18]



Abstract Dialectical Frameworks [BES+18]

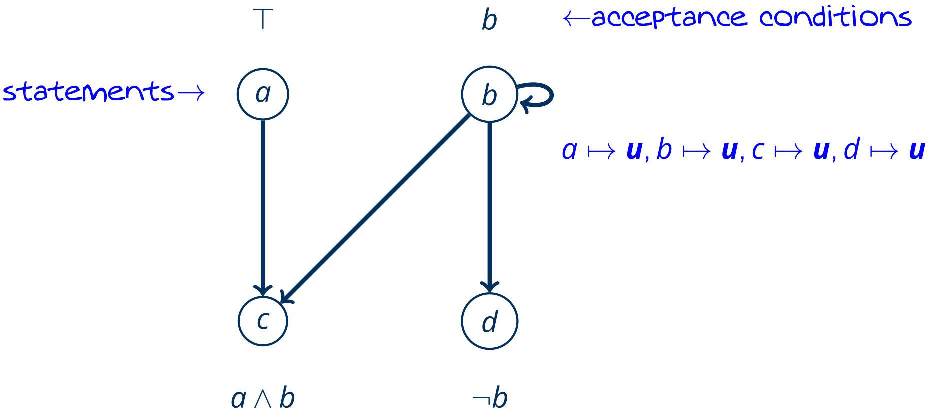


Abstract Dialectical Frameworks [BES+18]



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

Abstract Dialectical Frameworks [BES+18]

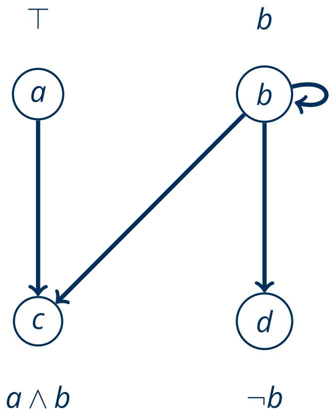


semantics: $\Gamma_D : \mathcal{V}^3 \mapsto \mathcal{V}^3$

Abstract Dialectical Frameworks [BES+18]

←acceptance conditions

statements→



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

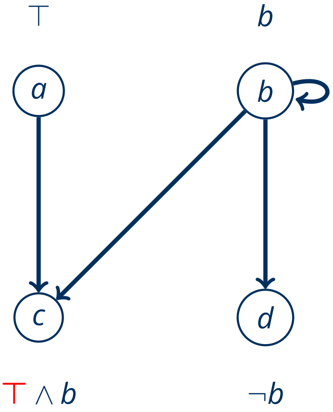
$a \mapsto \boldsymbol{t}, b \mapsto \boldsymbol{u}, c \mapsto \boldsymbol{u}, d \mapsto \boldsymbol{u}$

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

Abstract Dialectical Frameworks [BES+18]

←acceptance conditions

statements→



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

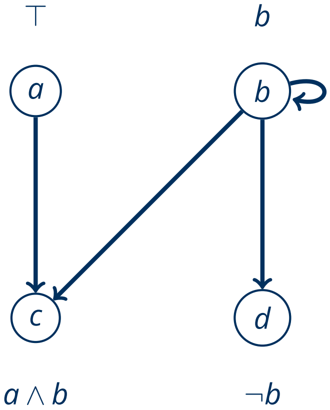
$a \mapsto \boldsymbol{t}, b \mapsto \boldsymbol{u}, c \mapsto \boldsymbol{u}, d \mapsto \boldsymbol{u}$

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

Abstract Dialectical Frameworks [BES+18]

←acceptance conditions

statements→



grounded, complete

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

complete

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

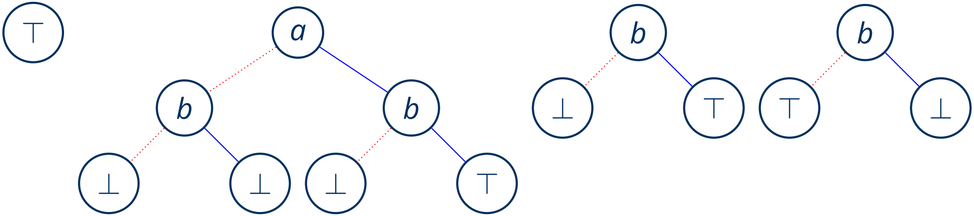
complete, stable

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

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

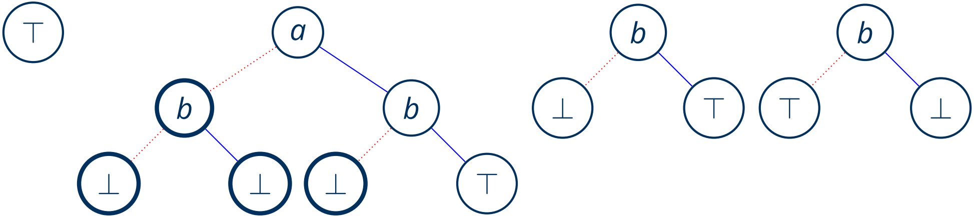
ordered Binary Decision Tree

- 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



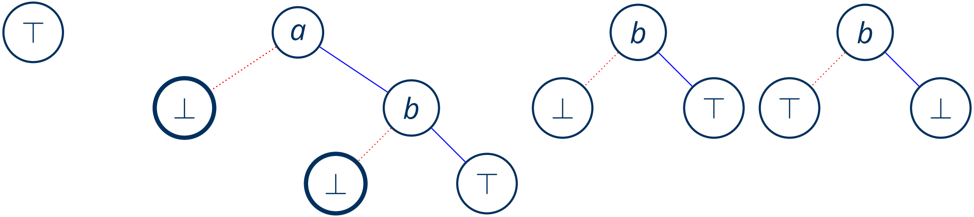
reduced ordered Binary Decision Diagram [Bry86]

- $lo(n) = hi(n)$, 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



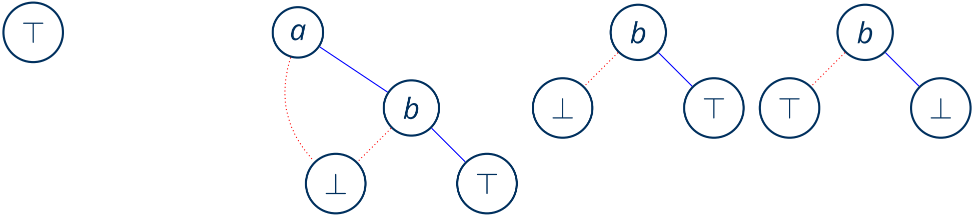
reduced ordered Binary Decision Diagram [Bry86]

- $lo(n) = hi(n)$, 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



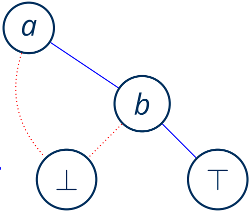
reduced ordered Binary Decision Diagram [Bry86]

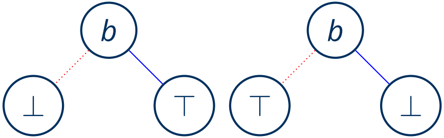
- $lo(n) = hi(n)$, then replace n by $hi(n)$
- if $n = v$, then replace v by n globally (violate tree-property)
- Given a variable order, this representation is unique under logical equivalence of formulae



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



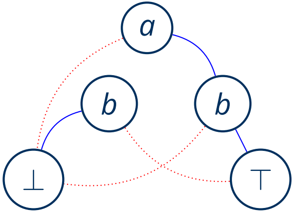


- Restriction linear

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

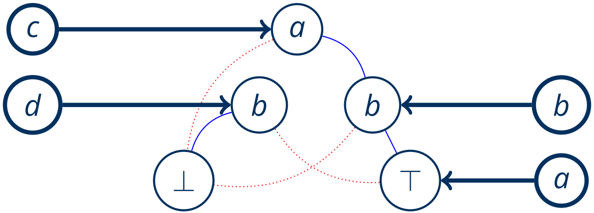
New idea: roBDDs to represent ADFs

- 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

- 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

- 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`

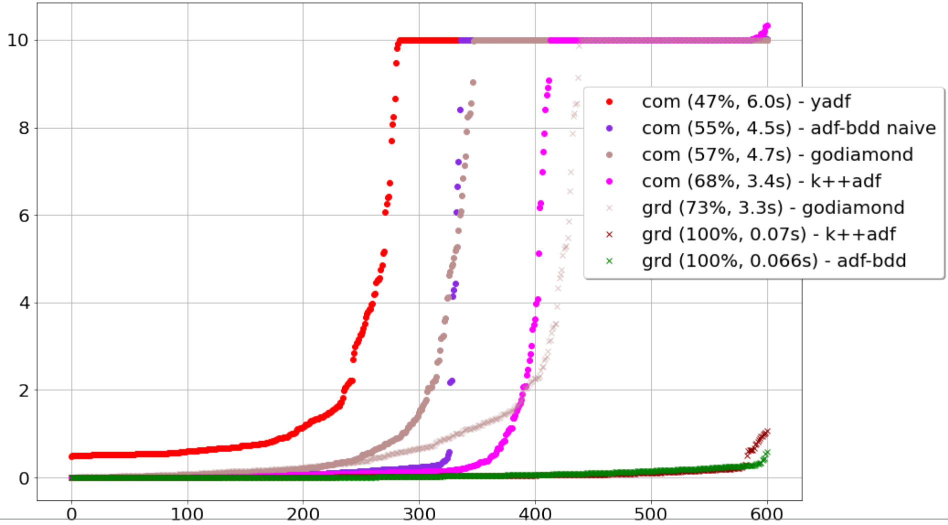
ADF-BDD solver

- 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

ADF-BDD Evaluation: grounded and complete



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. $\perp$
  - 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

Search Space Exploitation
with Faceted Navigation

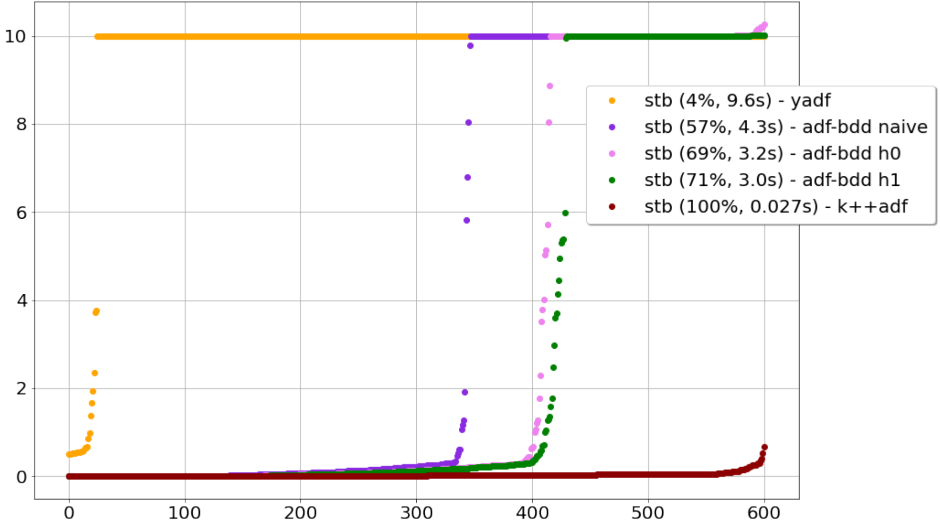
- Use Faceted Navigation measures to describe (Sub-)Search-Space
- Allows for an easy framework of properties for heuristics, like
  - Number of Models
  - Number of Facets
  - BDD Paths to $\top$ resp. $\perp$
  - Variable impact
  - ...

- 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

ADF-BDD Evaluation: stable



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

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,...)

Future Work

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

Thank you for your interest!

Visit ADF-BDD at

`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.