BAss: Symbolic Reasoning in Abstract Dialectical Frameworks

BDD-based solving via the ADF–BN correspondence

Samuel Pastva[1] Van-Giang Trinh[2]

KR 2026

[1]Masaryk University, Brno, Czechia [2]HCMUT, VNU-HCM, Ho Chi Minh City, Vietnam

Motivation & Background

Motivation

Argumentation formalizes reasoning with conflicting information.

Dung's AFs (1995): abstract arguments + binary attack relation.

- Simple and widely studied, but too restrictive: only attacks, no support or complex dependencies.

Abstract Dialectical Frameworks (ADFs): each argument has a propositional acceptance condition.

- Can express attacks, support, and arbitrary logical relationships.
- Applications: legal reasoning, dialog systems, *biology*, ...

The challenge: reasoning is hard — NP-, coNP-, up to Σ_2^P-complete. Enumeration of all solutions is often a practical bottleneck.

Abstract Dialectical Frameworks

Definition. An ADF is $D = (S, C)$; S = arguments, $C = \{\varphi_s\}$ = acceptance conditions (propositional formulas).

3-valued interpretations: $I : S \to \{0, 1, \star\}$ **Information ordering:** $\star <_i 0$, $\star <_i 1$

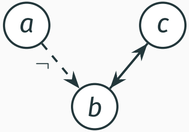
Characteristic operator: $\Gamma_D(I)(s)$ is 1 if $\varphi_s[I]$ is a tautology; is 0 if $\neg\varphi_s[I]$ tautology; (is $\star$ otherwise).

Semantics:

- Admissible: $I \leq_i \Gamma_D(I)$
- Complete: $I = \Gamma_D(I)$
- Preferred: $\leq_i$-maximal admissible
- 2-valued model: complete, no $\star$
- Stable: well-founded 2-valued model

stb $\subseteq$ 2m $\subseteq$ pr $\subseteq$ co $\subseteq$ ad

Running example



$\varphi_a = 1,\ \varphi_b = \neg a \vee c,\ \varphi_c = b$

Semantics: Intuition

Interp.	Type	Rationale
$I_5 = (\star, \star, \star)$	**admissible**	Every belief of I is *justified* by D's conditions; $I \leq_i \Gamma_D(I)$
$I_4 = (\star, 1, 1)$	admissible	
$I_3 = (1, \star, \star)$	**complete**	I contains *all* beliefs following from itself; $I = \Gamma_D(I)$
$I_2 = (1, 1, 1)$	**preferred**	$\leq_i$-maximal admissible, i.e., "fewest" $\star$'s (preferred)
	2-valued	Complete, but with no $\star$'s (2-valued model)
$I_1 = (1, 0, 0)$	**stable**, pref, 2-val	2-valued model, but every 1 is *well-founded* (no circular support).

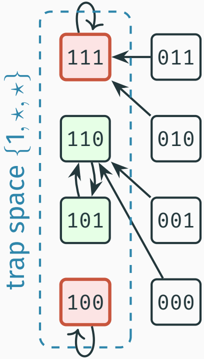
Details: $I_5 = (\star, 1, 1)$ is admissible but *not* complete. Restrict φ_s to I_5 (substitute $b{=}1, c{=}1$): $\varphi_a[I_5]{=}1,\ \varphi_b[I_5]{=}\neg a \vee 1{\equiv}1,\ \varphi_c[I_5]{=}1$ — all tautologies, so $\Gamma_D(I_5){=}(1,1,1) \neq I_5$ (at a: $\star \neq 1$). But $I_5 \leq_i \Gamma_D(I_5)$, hence admissible but not complete.

Boolean Networks

Definition. A Boolean Network is $N = \{f_1, \ldots, f_n\}$: n Boolean variables and n update functions $f_i : \{0,1\}^n \to \{0,1\}$. State $\mathbf{x} \in \{0,1\}^n$ updates to $F(\mathbf{x})$.

Key concepts:

- Fixed point: $F(\mathbf{x}) = \mathbf{x}$ (singleton attractor)
- Attractor: minimal inescapable set of states
- Trap space: inescapable hypercube (some vars fixed, others free)
- Minimal trap space: not contained in any other trap space



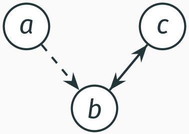
Boolean networks are popular models of gene regulation in biology (1000s of genes, complex logical dependencies, ...).

ADF $\leftrightarrow$ BN: A Formal Correspondence

From ADF to Boolean Network (and back)

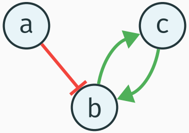
The translation. An acceptance condition φ_s *is* a Boolean update function f_s. Same syntax, same structure.

ADF



$\varphi_a = 1, \; \varphi_b = \neg a \vee c, \; \varphi_c = b$

Boolean Network



$f_a = 1, \; f_b = \neg a \vee c, \; f_c = b$

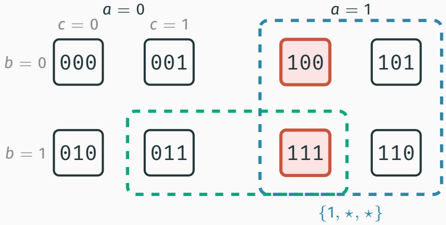
Formal bijections (Heyninck et al. 2024, Azpeitia et al. 2024)

ADF	Boolean Network
2-valued model	Fixed point
Preferred interpretation	Minimal trap space
Complete interpretation	"Fully percolated" trap space
Admissible interpretation	Any trap space

Worked Example: The Correspondence in Action

Running example: $f_a = 1, \quad f_b = \neg a \vee c, \quad f_c = b$

	ADF	BN
$I_1 = (1, 0, 0)$	2-val, pref, stable	fixed, min
$I_2 = (1, 1, 1)$	2-val, pref	fixed, min
$I_3 = (1, \star, \star)$	complete	percolated
$I_4 = (\star, 1, 1)$	admissible	trap
$I_5 = (\star, \star, \star)$	admissible	trap



Details: The trap space $\{1, \star, \star\}$ contains both fixed points *and* a 2-cycle — it's a trap space but not minimal, just like I_3 is complete but not preferred.

Bottom line

Tooling for Boolean networks and Abstract Dialectical Frameworks is *largely* interchangeable.

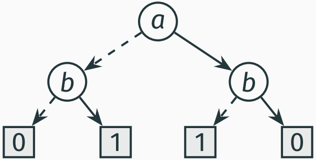
BDDs for Symbolic Computation

Binary Decision Diagrams

A BDD is a rooted DAG encoding a Boolean function $f : \{0,1\}^k \to \{0,1\}$.

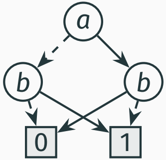
- **Decision nodes:** variable + two successors (positive/negative)
- **Terminals:** **0** and **1**
- **Shannon expansion:** $f = (x \wedge f[x{=}1]) \vee (\neg x \wedge f[x{=}0])$

Decision tree for $f = a \oplus b$



7 nodes (4 terminals)

Reduced BDD for $f = a \oplus b$



5 nodes (2 shared terminals)

Reduced + ordered BDDs are canonical[1] $\Rightarrow$ same f always gives the same BDD.

[1] For a fixed variable ordering.

BDD Operations & Symbolic Computation

Key operations:

- **Binary op:** $A \circ B$ ($\wedge, \vee, \Rightarrow$, ...): $\mathcal{O}(|A| \cdot |B|)$
- **Restriction:** $f[x{=}b]$ (substitute and simplify)
- **Existential quantification:** $\exists x.\, f = f[x{=}1] \vee f[x{=}0]$ (*projection*)

Symbolic computation = sets as BDDs

A set $X \subseteq \{0,1\}^n$ (or relation) is represented by its characteristic function's BDD.
Union = $\vee$, Intersection = $\wedge$, Complement = $\neg$, Projection = $\exists$, ...

"Fully symbolic" = operate on the BDD, never enumerate. A single BDD can often represent large sets compactly — this is the main mechanism behind BAss.

Symbolic Encoding of ADFs

2-valued: each argument s is a Boolean variable. **3-valued:** dual encoding with $(s^\top, s^\perp)$ per argument: $(1,0) \equiv 1$, $(0,1) \equiv 0$, $(1,1) \equiv \star$, $(0,0)$ is invalid.

Dual encoding of the characteristic operator (Γ_D):

$$dual(f) = \exists s_1, \ldots, s_n. \left(f \wedge \bigwedge_{s_i} (s_i \Rightarrow s_i^\top) \wedge (\neg s_i \Rightarrow s_i^\perp)\right)$$

Intuition: $dual(f)$ "lifts" functions from states (2-valued interpretations) to subspaces (3-valued interpretations):

$$\big(dual(f)(hypercube) = 1\big) \textbf{ iff } \big(\exists state.\ f(state) = 1\big)$$

Symbolic characteristic operator: $\Gamma_D \equiv (\varphi_s^\top, \varphi_s^\perp) = (dual(\varphi_s),\ dual(\neg\varphi_s))$

Key optimization: compute $dual(f)$ *directly* on the BDD of f — no DNF translation needed (avoids blowup for complex problem instances).

Computing ADF Semantics Symbolically

2-valued, Admissible, and Complete Interpretations

Once we have the dual encoding, characterization is straightforward:

$$f_{\mathtt{2m}(D)} = \bigwedge_{s \in S} \left(s \Leftrightarrow \varphi_s\right)$$

$$f_{\mathtt{ad}(D)} = \bigwedge_{s \in S} \left((\varphi_s^{\top} \Rightarrow s^{\top}) \wedge (\varphi_s^{\perp} \Rightarrow s^{\perp})\right)$$

$$f_{\mathtt{co}(D)} = f_{\mathtt{ad}(D)} \wedge \bigwedge_{s \in S} \left((s^{\top} \wedge s^{\perp}) \Rightarrow (\varphi_s^{\top} \wedge \varphi_s^{\perp})\right)$$

- Each formula's BDD represents the entire solution set as a single object.
- Greedy conjunction: minimize intermediate BDD sizes by always merging the smallest pair first (simple but works!).

Preferred Interpretations and Stable Models

Problem: preferred = admissible and $\leq_i$-maximal. Naive: up to $|\mathrm{co}(D)|$ iterations.

Key insight: group by number of $\star$ values

Same $\star$-count $\Rightarrow$ $\leq_i$-incomparable $\Rightarrow$ process all at once.
At most $|S|$ $\star$-counts $\Rightarrow$ $\mathcal{O}(|S|)$ symbolic iterations.

Algorithm (initiated with set $ad(D)$):

1. Find any interpretation with the most $\star$ values overall (k).
2. Filter out all interpretations with k $\star$ values $\Rightarrow$ Save these into $prf(D)$.
3. Remove any I' such that $I' \leq_i I$ for some $I \in prf(D)$; repeat until empty.

Stable models

Initiated with $2m(D)$, swaps $\leq_i$ for $\leq_t$ ($0 \leq_t 1$), and select models with the most 0 values (correctness: [Strauss2013]).

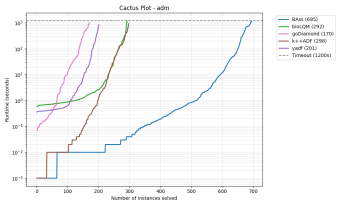
Evaluation & Impact

Full solution search

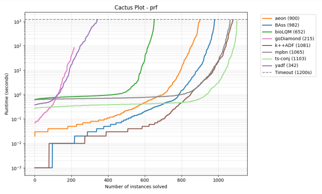
Setup: 1,237 instances (ICCMA ADFs + BBM BNs) • 1,200s timeout • 11 tools

	BAss	k++adf	goDiamond	yadf	adf-bdd	ts-conj	bioLQM
admiss.	**695 (338)**	298 (0)	170 (0)	201 (0)	N/A	N/A	292 (0)
complete	**974 (85)**	883 (23)	190 (0)	535 (21)	476 (0)	N/A	619 (0)
preferred	982 (39)	1081 (0)	215 (0)	342 (0)	N/A	**1103 (1)**	652 (0)
2-valued	**1142 (3)**	1116 (0)	1130 (0)	N/A	950 (0)	1104 (0)	1038 (0)
stable	**1149 (50)**	1115 (16)	202 (0)	65 (0)	571 (0)	N/A	N/A

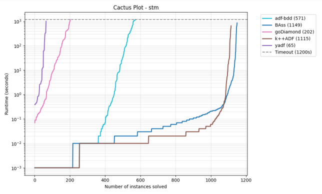
Each cell: #solved (uniquely solved). Bold = best.



Admissible



Preferred



Stable

Conclusion

Conclusion

BAss: BDD-based ADF solver leveraging the ADF–BN correspondence.

- Novel fully symbolic characterization of Γ_D, enabling in particular algorithms for preferred and stable semantics
- First BDD-based ADF solver supporting these symbolically
- Dramatically outperforms prior BDD tools — `adf-bdd` (ADFs), `bioLQM` (BNs)
- Competitive with SAT/ASP when full solution set is needed
- Uniquely solves biological networks beyond reach of previous tools

Future work:

- Hybrid symbolic-SAT strategies
- Further transfer of BN theory into ADFs (model reductions, ...)
- Additional semantics (naive, stage, semi-stable)

Thank you!

Questions?