Computational Approaches to Reasoning in Structured Argumentation

Tuomo Lehtonen

November 16, 2025

KR Early Career Award 2025

Aalto University, Finland

Aalto University

HIIT | HELSINKI INSTITUTE FOR INFORMATION TECHNOLOGY

Setting

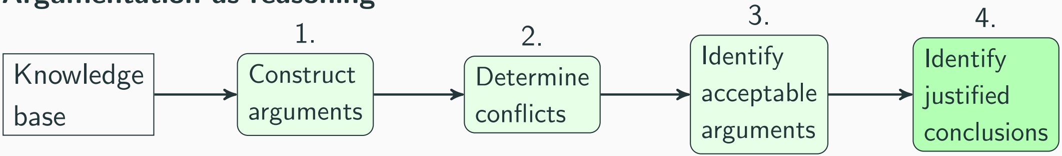
- Argumentation: the process of concluding the most reasonable over contradictory and uncertain viewpoints
  - Daily life, science, law, politics, recreation,...

Setting

- Argumentation: the process of concluding the most reasonable over contradictory and uncertain viewpoints
  - Daily life, science, law, politics, recreation,...
- Computational/formal argumentation
  - Decision support: medical, legal, consumer
  - Debate analysis
  - Explainable artificial intelligence

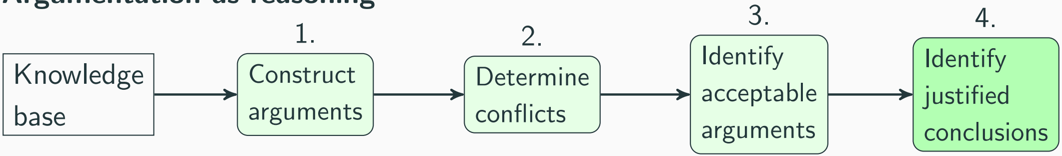
Setting

Argumentation as reasoning



Setting

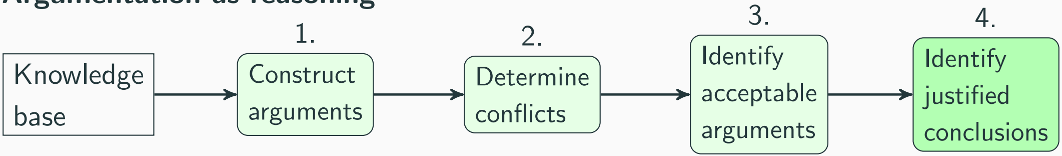
Argumentation as reasoning



- *Abstract* argumentation (AFs): step 3 (and 4), arguments and conflicts are input
- Structured argumentation formalizes all steps
  - Arguments do have a structure
  - AFs might miss dependencies

Setting

Argumentation as reasoning



- *Abstract* argumentation (AFs): step 3 (and 4), arguments and conflicts are input
- Structured argumentation formalizes all steps
  - Arguments do have a structure
  - AFs might miss dependencies
- My focus on two central structured argumentation formalisms
  - Assumption-based argumentation (ABA)
  - Abstract rule-based argumentation (ASPIC+)
  - Here assume atoms and ground rules

Outline

1. Brief introduction to ABA
2. Why to not construct AFs: ABA reasoning
3. “But ASPIC$^+$ *requires* AF construction!”
4. Not anymore

Assumption-based argumentation (ABA) [Bondarenko et al., 1997]

- ABA framework: *Assumptions* $\mathcal{A}$, *rules* $\mathcal{R}$ and *contraries* $\overline{}$ (to assumptions)
- A set of assumptions $S \subseteq \mathcal{A}$ attacks an assumptions a if $S \models_{\mathcal{R}} \overline{a}$

Example

Consider: assumptions $\mathcal{A} = \{a, b, c, d\}$, rules $\mathcal{R} = \{x \leftarrow a, b\}$, and contraries $\overline{c} = x, \overline{d} = c$

Assumption-based argumentation (ABA) [Bondarenko et al., 1997]

- ABA framework: *Assumptions* $\mathcal{A}$, *rules* $\mathcal{R}$ and *contraries* $\overline{}$ (to assumptions)
- A set of assumptions $S \subseteq \mathcal{A}$ attacks an assumptions a if $S \models_{\mathcal{R}} \overline{a}$

Example

Consider: assumptions $\mathcal{A} = \{a, b, c, d\}$, rules $\mathcal{R} = \{x \leftarrow a, b\}$, and contraries $\overline{c} = x, \overline{d} = c$

Task: find a *stable* set of assumptions (i.e.
a set that attacks all other sets)

- $\{a, b, d\}$ is stable

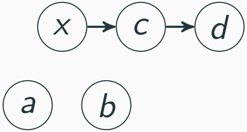
Assumption-based argumentation (ABA) [Bondarenko et al., 1997]

- ABA framework: *Assumptions* $\mathcal{A}$, *rules* $\mathcal{R}$ and *contraries* $\overline{}$ (to assumptions)
- A set of assumptions $S \subseteq \mathcal{A}$ attacks an assumptions a if $S \models_{\mathcal{R}} \overline{a}$

Example

Consider: assumptions $\mathcal{A} = \{a, b, c, d\}$, rules $\mathcal{R} = \{x \leftarrow a, b\}$, and contraries $\overline{c} = x, \overline{d} = c$

Equivalently, construct AF:



Task: find a *stable* set of assumptions (i.e. a set that attacks all other sets)

- $\{a, b, d\}$ is stable

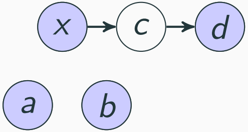
Assumption-based argumentation (ABA) [Bondarenko et al., 1997]

- ABA framework: *Assumptions* $\mathcal{A}$, *rules* $\mathcal{R}$ and *contraries* $^{-}$ (to assumptions)
- A set of assumptions $S \subseteq \mathcal{A}$ attacks an assumptions a if $S \models_{\mathcal{R}} \overline{a}$

Example

Consider: assumptions $\mathcal{A} = \{a, b, c, d\}$, rules $\mathcal{R} = \{x \leftarrow a, b\}$, and contraries $\overline{c} = x, \overline{d} = c$

Equivalently, construct AF:



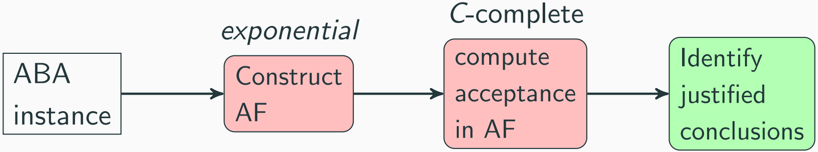
Task: find a *stable* set of assumptions (i.e. a set that attacks all other sets)

- $\{a, b, d\}$ is stable

Find stable extension; standard AF reasoning task

Computing argumentative reasoning

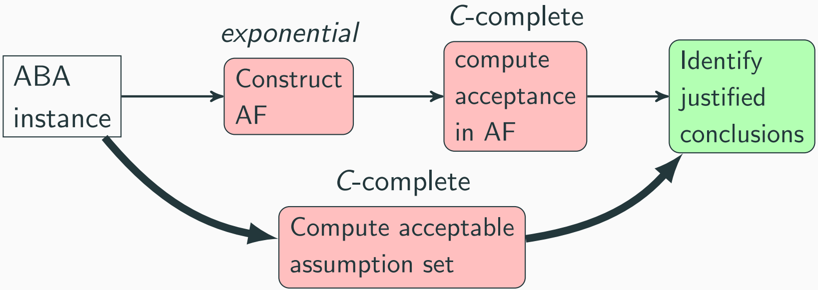
- Number of arguments is not polynomially bounded [Strass et al. 2019]
- Typically acceptance problems are hard (i.e. below C=NP or harder)



Computing argumentative reasoning

- ▷ Number of arguments is not polynomially bounded [Strass et al. 2019]
- ▷ Typically acceptance problems are hard (i.e. below C=NP or harder)
  - ▷ For ABA, same complexity as AFs
  - ▷ For ASPIC$^+$... find out in 5 minutes

→ Construct arguments ≈ grow the input exponentially for no (computational) gain



Contributions: ABA

- First algorithms for ABA with modern constraint solving
  - Here: answer set programming (ASP) and propositional satisfiability (SAT)
- Avoid performance bottleneck of constructing arguments
- This is possible with the original assumption-set definition of ABA
- Open source implementations `bitbucket.org/coreo-group/aspforaba`

Joint work with Wallner and Järvisalo published in AAAI'19, and TPLP and JAIR in 2021

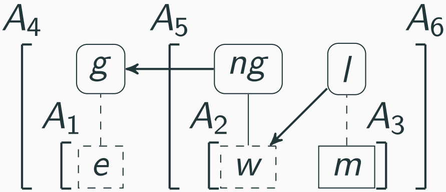
Performance for ABA

Problem	Approach	#timeouts	Running times (s) mean	sum
ABA admissible *credulous acceptance*	**ASPforABA**	**0**	**0.018**	**31**
	ABAGRAPH	200	8.464	12932
	ABA2AF	364	13.990	19078
ABA stable *skeptical acceptance*	**ASPforABA**	**0**	**0.008**	**38**
	ABA2AF	648	10.942	43386
ABA grounded *acceptance*	**ASPforABA**	**0**	**0.127**	**220**
	ABAGRAPH	210	9.979	15148
ABA preferred *solution enumeration*	**ASPforABA**	**0**	**0.333**	**226**
	ABA2AF	255	6.082	2585
ABA complete *solution enumeration*	**ASPforABA**	**0**	**0.005**	**1**
	ABAPLUS	9	15.287	1697
ABA ideal *find the ideal set*	**ASPforABA**	**0**	**0.025**	**3**
	ABAPLUS	18	22.490	2293

Our algorithms won ABA track of ICCMA 2023 and (more narrowly) 2025

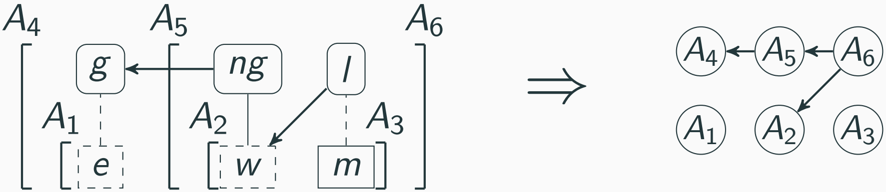
Differences to ABA

- support defeasible rules, and certain premises
- Typically include preferences among premises and rules
- Three attack types, which *succeed* depending on preferences
  - attack on premise (as in ABA),
  - on a (defeasible) rule, and
  - on the head of a (defeasible) rule



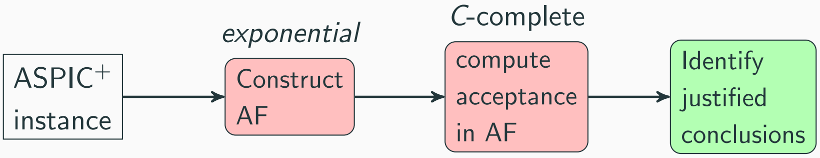
Differences to ABA

- support defeasible rules, and certain premises
- Typically include preferences among premises and rules
- Three attack types, which *succeed* depending on preferences
  - attack on premise (as in ABA),
  - on a (defeasible) rule, and
  - on the head of a (defeasible) rule



Contributions: ASPIC$^+$ without AFs

- Problem: ASPIC$^+$ defined in terms of AFs

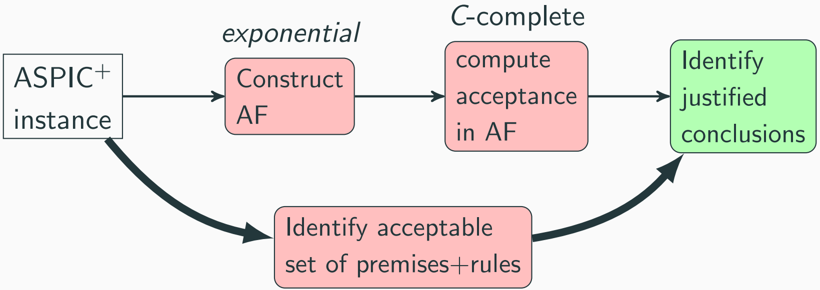


Contributions: ASPIC$^+$ without AFs

- ▷ Problem: ASPIC$^+$ defined in terms of AFs

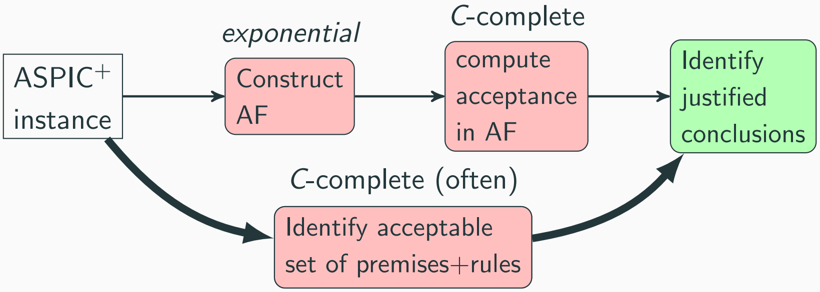
→ We characterized acceptance in terms of set of premises+rules (P, D) rather than arguments

  - ▷ Analogy to assumption-set definition of ABA



Contributions: ASPIC$^+$ without AFs

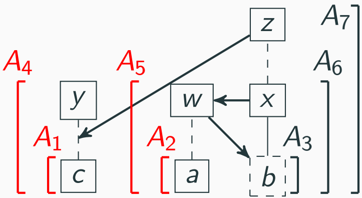
- ▷ Problem: ASPIC$^+$ defined in terms of AFs
- → We characterized acceptance in terms of set of premises+rules (P, D) rather than arguments
  - ▷ Analogy to assumption-set definition of ABA
- ▷ Enables proofs of the computational complexity of claim acceptance



Contributions: ASPIC$^+$ without AFs

We characterized acceptance in terms of set of premises+rules (P, D) instead of arguments

- conditions on (P, D) s.t. the set of arguments based on (P, D) is an extension
- Key property to show: arguments from (P, D) defeating a premise p (resp. defeasible rule d) implies that they defeat any argument using p (resp. d)



Stable extension $\{A_1, A_2, A_4, A_5\}$.

Corresponding $(P, D) = (\emptyset, \{c \Rightarrow y, a \Rightarrow w\})$. Arguments from (P, D), i.e. $\{A_1, A_2, A_3, A_4\}$ defeats b and any argument using it

Contributions: ASPIC$^+$ complexity of credulous/skeptical acceptance (DC/DS)

semantics	preferences		
	no[1]	weakest-link[2]	last-link
complete (CO)	NP/coNP-c	?	NP/coNP-c
preferred (PR)	NP/Π_2^P-c	?	NP/Π_2^P-c
stable (ST)	NP/coNP-c	Σ_2^P/Π_2^P-c	NP/coNP-c
grounded (GR)	in P	?	in P

[Lehtonen, Wallner, Järvisalo, KR'20]

[Lehtonen, Wallner, Järvisalo, KR'22]

[Lehtonen, Odekerken, Wallner, Järvisalo, KR'24]

[1] Reductions from ASPIC$^+$ to ABA were shown under some semantics, implying complexity results.

[2] The "key thing" from last slide does not hold under weakest-link principle.

Performance for ASPIC$^+$ versus an AF construction approach

PyArg (construct+solve AF)

#atoms	**#solved** (mean run time over solved (s))							
	DC-ST		DC-CO		DC-AD		DS-ST	
50	2	(20.8)	2	(20.0)	1	(34.7)	2	(19.5)
>50	0	—	0	—	0	—	0	—

Our ASP approach

#atoms	**#solved** (mean run time over solved (s))							
	DC-ST		DC-CO		DC-AD		DS-ST	
50	5	(0.1)	5	(0.2)	5	(0.2)	5	(0.1)
100	5	(0.3)	5	(0.5)	5	(0.5)	5	(0.3)
200	5	(1.7)	5	(2.9)	5	(3.0)	5	(1.7)
400	5	(11.6)	5	(14.7)	5	(16.9)	5	(9.6)
800	5	(64.8)	5	(87.6)	5	(97.9)	5	(59.3)
1200	5	(175.5)	5	(226.4)	5	(239.0)	5	(181.6)
1600	5	(422.6)	4	(518.9)	4	(543.4)	5	(450.0)
1700	3	(473.8)	2	(587.9)	1	(591.1)	3	(486.4)
1800	2	(569.5)	0	—	0	—	3	(568.8)
1900	0	—	0	—	0	—	0	—

`bitbucket.org/coreo-group/aspforaspic`

Contributions: ASPIC$^+$ application

- A customer-facing ASPIC$^+$ application in use by the Netherlands Police

  [Odekerken, Bex, Borg, Testerink. Intelligent Systems with Applications 2022]

- We developed efficient algorithms for problems related to incomplete information; enabled by recharacterizing ASPIC$^+$

  [Odekerken, Lehtonen, Borg, Wallner, and Järvisalo, KR 2023]
  [Odekerken, Lehtonen, Wallner, and Järvisalo, JAIR 2025]

`bitbucket.org/coreo-group/raspic`

Summary

Contributions

- **Characterization of central ASPIC$^+$ semantics** without constructing AF
  - **Complexity results**
- **Declarative algorithms** for ASPIC$^+$ and ABA, open source implementations

Summary

Contributions

- **Characterization of central ASPIC$^+$ semantics** without constructing AF
  - **Complexity results**
- **Declarative algorithms** for ASPIC$^+$ and ABA, open source implementations

- Arguments have a structure
- You have to pay computationally if you want to construct an AF
- Omitted ABA topics: preferences, non-flat, default logic fragment, more efficient ABA$\rightarrow$AF translations

Summary

Contributions

- **Characterization of central ASPIC$^+$ semantics** without constructing AF
  - **Complexity results**
- **Declarative algorithms** for ASPIC$^+$ and ABA, open source implementations

- Arguments have a structure
- You have to pay computationally if you want to construct an AF
- Omitted ABA topics: preferences, non-flat, default logic fragment, more efficient ABA$\rightarrow$AF translations

Hear more in **Constraints session** after this: SAT approach to ABA

Thank you for your attention!

Thanks to

- PhD supervisors: Matti Järvisalo and Johannes P. Wallner
- Other coauthors on ABA/ASPIC$^+$: Daphne Odekerken, Anna Rapberger, Markus Ulbricht, Francesca Toni, Andreas Niskanen, Masood Feyzbakhsh Rankooh, AnneMarie Borg