Making Argument Calculi Proof-Theoretically Robust

Cut-Elimination, Analyticity, and Interpolation

Kees van Berkel[1] Andrea Sabatini[2] Christian Straßer[3]

[1]TU Wien [2]Scuola Normale Superiore di Pisa [3]Ruhr-Universität Bochum

AI^3 2026 – Perugia

Outline

1. Motivation
2. The AC framework
3. Cut-free AC
4. Reasoning by cases
5. Analyticity
6. Conclusion

Outline

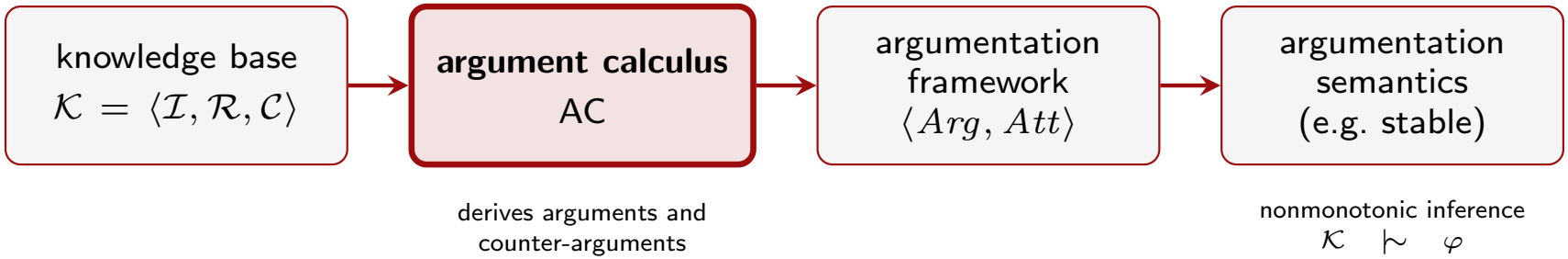
1. Motivation
2. The AC framework
3. Cut-free AC
4. Reasoning by cases
5. Analyticity
6. Conclusion

Defeasible reasoning, argumentatively

Reasoning with defeasible information is nonmonotonic: conclusions may have to be withdrawn when new information becomes available.

Many formalisms: default logic (Reiter 1980), input/output logics (Makinson & van der Torre 2000), autoepistemic, defeasible and adaptive logics, ...

Formal argumentation (Dung 1995) has proven to be a unifying framework for them (Arieli et al. 2021):



In logical argumentation, (counter-)arguments are derived by means of a proof system: the calculus fixes what counts as a reason for what.
The argumentation semantics decides which arguments survive their attackers.

Our question: is the calculus a good proof system w.r.t. analytic proof search?

Argument calculi in a nutshell

Argument Calculi AC (van Berkel & Straßer 2022) are sequent-style systems in which defaults and norms are first-class proof objects. They derive two kinds of arguments:

- detachment: $p^i, (p, q) \Rightarrow q^o$ – input p and rule (p, q) are reasons for output q;
- inapplicability: $p^i, \neg q^c \Rightarrow \neg(p, q)$ – applying (p, q) would violate the constraint $\neg q$.

An argument concluding $\neg(\varphi, \psi)$ attacks every argument having (φ, ψ) among its reasons.
Varying the rules yields normal default logic and all constrained I/O logics.

Example

$\mathcal{K} = \langle \{p^i\}, \{(p, q)\}, \{\neg q^c\} \rangle$:



No rule is among the reasons of b: b is unattacked, a is defeated and q is not inferred.

The problem: Cut

- AC make essential use of Cut: a default is applied by detachment, and its consequent is propagated through the derivation by Cut.

$$\frac{\Gamma \Rightarrow \varphi \qquad \varphi, \Gamma' \Rightarrow \Delta}{\Gamma, \Gamma' \Rightarrow \Delta}\ \text{Cut}$$

- Root-first, one has to guess the cut formula among infinitely many candidates: no subformula property, no syntax-driven proof search.
- Decomposition trees (van Berkel & Sabatini, KR 2026) decide derivability: in a sense, we turn them upside down to get cut-free calculi.

This talk

1. Left-introduction rules for defaults and norms internalize detachment: Cut becomes redundant.
2. A modular Cut-elimination argument by a single induction, reducing everything to Cut in the base logic.
3. Consequences: generalized subformula property, Craig interpolation.

Outline

1 Motivation

2 The AC framework

3 Cut-free AC

4 Reasoning by cases

5 Analyticity

6 Conclusion

Labelled languages and knowledge bases

Let $\mathcal{L}$ be a classical propositional language (with $\top, \bot$) and $x \in \{i, o, c\}$.

- $\mathcal{L}^x = \{\varphi^x \mid \varphi \in \mathcal{L}\}$: input, output and constraint formulas;
- $\mathcal{L}^r = \{(\varphi, \psi) \mid \varphi, \psi \in \mathcal{L}\}$: defeasible rules, read as "if φ, then ψ is plausible / believed / obligatory";
- $\overline{\mathcal{L}^r} = \{\neg(\varphi, \psi) \mid (\varphi, \psi) \in \mathcal{L}^r\}$: inapplicability statements.

A knowledge base is a triple $\mathcal{K} = \langle \mathcal{I}, \mathcal{R}, \mathcal{C} \rangle$, with $\mathcal{I} \subseteq \mathcal{L}^i$, $\mathcal{R} \subseteq \mathcal{L}^r$ and $\mathcal{C} \subseteq \mathcal{L}^c$.

Input/output-style reasoning: input triggers rules, triggered rules detach output, output is controlled by constraints.

An AC-argument is a sequent $\Gamma \Rightarrow \Delta$, where Γ is a finite set and Δ contains at most one formula of

$$\mathcal{L}^{dac} = \mathcal{L}^i \cup \mathcal{L}^o \cup \mathcal{L}^c \cup \mathcal{L}^r \cup \overline{\mathcal{L}^r}.$$

The base logic L (here, classical logic) governs reasoning within each label; $\Gamma^{\downarrow}$ is Γ stripped of its labels.

The rules of AC

$$\frac{\Gamma \vdash_{\mathsf{L}} \bigvee \Delta}{\Gamma^x \Rightarrow \Delta^x}\ \mathsf{Ax} \qquad \frac{}{\Rightarrow (\top, \top)}\ \mathsf{Taut} \qquad \frac{}{\varphi^i, (\varphi, \psi) \Rightarrow \psi^o}\ \mathsf{BDet} \qquad \frac{}{\varphi^i \Rightarrow \varphi^o}\ \mathsf{TP}$$

$$\frac{\Gamma, \varphi^i \Rightarrow \Delta \qquad \Gamma', \psi^i \Rightarrow \Delta}{\Gamma, \Gamma', (\varphi \vee \psi)^i \Rightarrow \Delta}\ \mathsf{Cases}^{(b)} \qquad \frac{\varphi^i, \Gamma \Rightarrow \Delta}{\varphi^o, \Gamma \Rightarrow \Delta}\ \mathsf{SDet}^{(a)} \qquad \frac{\Gamma \Rightarrow \varphi \qquad \varphi, \Gamma' \Rightarrow \Delta}{\Gamma, \Gamma' \Rightarrow \Delta}\ \mathsf{Cut}$$

$$\frac{\Gamma \Rightarrow \varphi^o}{\Gamma, \neg\varphi^c \Rightarrow}\ \mathsf{IO1} \qquad \frac{\Gamma, (\varphi, \psi) \Rightarrow}{\Gamma \Rightarrow \neg(\varphi, \psi)}\ \mathsf{IO2} \qquad \frac{}{\varphi^i, \neg\psi^o \Rightarrow \neg(\varphi, \psi)}\ \mathsf{DL1}$$

$x \in \{i, o, c\}$; (a) $\Gamma \cap \mathcal{L}^r \neq \varnothing$ if $\mathsf{TP} \notin \mathcal{S}$; (b) $\Gamma \cap \mathcal{L}^r \neq \varnothing \neq \Gamma' \cap \mathcal{L}^r$ if $\mathsf{TP} \notin \mathcal{S}$.

- BDet: basic detachment; TP: throughput; SDet: successive detachment (output is reused as input); Cases: reasoning by cases on the input.
- IO1: if Γ yields output φ, then Γ is incoherent with the constraint $\neg\varphi$

  IO2: in an incoherent set of reasons, each rule is inapplicable given the others;

  DL1: a triggered rule is inapplicable if the output refutes its consequent.

The AC family

rules $\mathcal{S}$ added to $\{\mathsf{Ax}, \mathsf{BDet}, \mathsf{Cut}\}$	logic	reasoning style
$\{\mathsf{Taut}, \mathsf{IO1}, \mathsf{IO2}\}$	out_1	simple-minded
$\{\mathsf{Taut}, \mathsf{IO1}, \mathsf{IO2}, \mathsf{Cases}\}$	out_2	basic (reasoning by cases)
$\{\mathsf{Taut}, \mathsf{IO1}, \mathsf{IO2}, \mathsf{SDet}\}$	out_3	reusable (chaining)
$\{\mathsf{Taut}, \mathsf{IO1}, \mathsf{IO2}, \mathsf{SDet}, \mathsf{Cases}\}$	out_4	basic reusable
each of the above $\cup \{\mathsf{TP}\}$	out_n^+	with throughput
$\{\mathsf{Taut}, \mathsf{SDet}, \mathsf{DL1}\}$ $(\cup \{\mathsf{TP}\})$	normal default logic	

Each $\mathcal{K} = \langle \mathcal{I}, \mathcal{R}, \mathcal{C} \rangle$ induces an argumentation framework $\langle Arg, Att \rangle$:

- Arg: all $\mathsf{AC}_\mathcal{S}$-derivable $\Gamma \Rightarrow \Delta$ with $\Gamma \subseteq \mathcal{I} \cup \mathcal{R} \cup \mathcal{C}$;
- a attacks b iff $a = \Gamma \Rightarrow \neg(\varphi, \psi)$ and (φ, ψ) is a premise of b.

Theorem (van Berkel & Straßer 2022)

$\mathcal{E}$ is a stable extension iff the rules occurring in $\mathcal{E}$ form a maximal consistent set of rules in the corresponding I/O logic or normal default logic.

Cut does all the work

AC have no logical rules: base-logic reasoning is outsourced to Ax and Cut.

Collecting outputs:

$$\dfrac{\dfrac{}{p^i, (p,q) \Rightarrow q^o}\,\text{BDet} \qquad \dfrac{\dfrac{}{p^i, (p,r) \Rightarrow r^o}\,\text{BDet} \qquad \dfrac{}{q^o, r^o \Rightarrow (q \wedge r)^o}\,\text{Ax}}{q^o, p^i, (p,r) \Rightarrow (q \wedge r)^o}\,\text{Cut}}{p^i, (p,q), (p,r) \Rightarrow (q \wedge r)^o}\,\text{Cut}$$

Chaining (out_3):

$$\dfrac{\dfrac{}{p^i, (p,q) \Rightarrow q^o}\,\text{BDet} \qquad \dfrac{\dfrac{}{q^i, (q,r) \Rightarrow r^o}\,\text{BDet}}{q^o, (q,r) \Rightarrow r^o}\,\text{SDet}}{p^i, (p,q), (q,r) \Rightarrow r^o}\,\text{Cut}$$

- The detached outputs q^o and r^o disappear from the conclusion.
- Root-first proof search must guess them – among infinitely many formulas.
- Goal: a calculus in which defaults are applied locally, without Cut.

Outline

1 Motivation

2 The AC framework

3 Cut-free AC

4 Reasoning by cases

5 Analyticity

6 Conclusion

Defaults as left-introduced formulas

Key idea: replace BDet and Cut with a left-introduction rule for defeasible rules:

$$\frac{\Gamma \Rightarrow \varphi^i \qquad \psi^o, \Delta \Rightarrow \Lambda}{\Gamma, \Delta, (\varphi, \psi) \Rightarrow \Lambda}\ \mathsf{Lio}^{\mathsf{i}} \qquad \frac{\Gamma \vdash A \qquad B, \Delta \vdash C}{A \to B, \Gamma, \Delta \vdash C}\ L{\to}$$

- left premise: the rule is triggered by the reasons Γ;
- right premise: the reasoning licensed by the rule, with its consequent ψ^o available as a local hypothesis.

Reusable output (out_3): Lio^x with $x \in \{i, o\}$ – a rule can also be triggered by output.

Lemma

Lio^i is admissible in $\mathsf{AC}_\varnothing$; Lio^o is admissible if $\mathsf{SDet} \in \mathcal{S}$.

$$\dfrac{\dfrac{\Gamma \Rightarrow \varphi^i \qquad \dfrac{}{\varphi^i, (\varphi, \psi) \Rightarrow \psi^o}\ \mathsf{BDet}}{\Gamma, (\varphi, \psi) \Rightarrow \psi^o}\ \mathsf{Cut} \qquad \psi^o, \Delta \Rightarrow \Lambda}{\Gamma, \Delta, (\varphi, \psi) \Rightarrow \Lambda}\ \mathsf{Cut}$$

The same examples, without Cut

Collecting outputs:

$$\dfrac{\dfrac{}{p^i \Rightarrow p^i}\,\mathsf{Ax} \qquad \dfrac{\dfrac{}{p^i \Rightarrow p^i}\,\mathsf{Ax} \qquad \dfrac{}{q^o, r^o \Rightarrow (q \wedge r)^o}\,\mathsf{Ax}}{p^i, r^o, (p,q) \Rightarrow (q \wedge r)^o}\,\mathsf{Lio^i}}{p^i, (p,q), (p,r) \Rightarrow (q \wedge r)^o}\,\mathsf{Lio^i}$$

Chaining (out_3):

$$\dfrac{\dfrac{}{p^i \Rightarrow p^i}\,\mathsf{Ax} \qquad \dfrac{\dfrac{}{q^o \Rightarrow q^o}\,\mathsf{Ax} \qquad \dfrac{}{r^o \Rightarrow r^o}\,\mathsf{Ax}}{q^o, (q,r) \Rightarrow r^o}\,\mathsf{Lio^o}}{p^i, (p,q), (q,r) \Rightarrow r^o}\,\mathsf{Lio^i}$$

- The former cut formulas q^o, r^o reappear as local hypotheses in right premises.
- Every formula is a subformula of the end-sequent (up to labels).
- Base-logic reasoning is confined to the leaves, within a single label.

Cut-free calculi: out_1, out_3 and default logic

The base system is $\mathsf{AC}^{\mathsf{cf}}_{\varnothing} = \{\mathsf{Ax}, \mathsf{Lio}^i\}$: no BDet, no Cut.

- out_1: $\mathsf{AC}^{\mathsf{cf}}_{\varnothing} + \{\mathsf{Taut}, \mathsf{IO1}, \mathsf{IO1}', \mathsf{IO2}\}$;
- out_3: $\mathsf{AC}^{\mathsf{cf}}_{\varnothing} + \{\mathsf{Taut}, \mathsf{IO1}, \mathsf{IO1}', \mathsf{IO2}, \mathsf{SDet}, \mathsf{Lio}^o\}$;
- normal default logic: $\mathsf{AC}^{\mathsf{cf}}_{\varnothing} + \{\mathsf{Taut}, \mathsf{SDet}, \mathsf{Lio}^o, \mathsf{DL1}, \mathsf{IO2}'\}$.

Two rules take over interactions that used to be mediated by Cut:

$$\frac{\Gamma^o, \Delta^o \Rightarrow}{\Gamma^o, \Delta^c \Rightarrow}\ \mathsf{IO1}' \qquad\qquad \frac{\Gamma, (\varphi, \psi) \Rightarrow}{\Gamma \Rightarrow \neg(\varphi, \psi)}\ \mathsf{IO2}'$$

- IO1′: constraints clash jointly with a set of outputs (formerly: Cut between Ax and IO1);
- IO2′: IO2 restricted to premises with an SDet-free derivation (default logic).

Lemma

The new rules are admissible in the corresponding AC systems. Conversely, BDet is derivable in $\mathsf{AC}^{\mathsf{cf}}_{\varnothing}$: apply Lio^i to $\varphi^i \Rightarrow \varphi^i$ and $\psi^o \Rightarrow \psi^o$.

Cut elimination: the core

Theorem

If Cut is admissible in the base logic L, then Cut is admissible in $\mathsf{AC}^{\mathsf{cf}}_{\varnothing}$.

Proof idea: induction on the number n of Lio^i instances in the derivations of the Cut premises. If $n = 0$, both premises are Ax instances: apply Cut in L. If $n > 0$, Cut is permuted upwards into the right premise of Lio^i, where the cut formula lives.

$$\dfrac{\dfrac{\Gamma' \Rightarrow \psi^i \qquad \theta^o, \Gamma'' \Rightarrow \varphi}{\Gamma', \Gamma'', (\psi, \theta) \Rightarrow \varphi}\,\mathsf{Lio^i} \qquad \dfrac{\Delta' \Rightarrow \xi^i \qquad \zeta^o, \varphi, \Delta'' \Rightarrow \Lambda}{\varphi, \Delta', \Delta'', (\xi, \zeta) \Rightarrow \Lambda}\,\mathsf{Lio^i}}{\Gamma', \Gamma'', (\psi, \theta), \Delta', \Delta'', (\xi, \zeta) \Rightarrow \Lambda}\,\mathsf{Cut}$$

$$\rightsquigarrow \qquad \dfrac{\Delta' \Rightarrow \xi^i \qquad \dfrac{\Gamma' \Rightarrow \psi^i \qquad \dfrac{\theta^o, \Gamma'' \Rightarrow \varphi \qquad \zeta^o, \varphi, \Delta'' \Rightarrow \Lambda}{\theta^o, \Gamma'', \zeta^o, \Delta'' \Rightarrow \Lambda}\,\mathsf{Cut}}{\Gamma', \Gamma'', (\psi, \theta), \zeta^o, \Delta'' \Rightarrow \Lambda}\,\mathsf{Lio^i}}{\Gamma', \Gamma'', (\psi, \theta), \Delta', \Delta'', (\xi, \zeta) \Rightarrow \Lambda}\,\mathsf{Lio^i}$$

No double induction on cut rank and height: the cut formula is never decomposed, it travels to the leaves where the base logic takes over.

Cut elimination: constraints, chaining and defaults

Theorem

Let $\mathcal{S}$ be the cut-free system for out_1, out_3 or normal default logic. If Cut is admissible in L, so is every Cut on a formula other than $(\top, \top)$ in $\mathsf{AC}^{\mathsf{cf}}_{\mathcal{S}}$.

- Input is pure: if $\Gamma \Rightarrow \varphi^i$ is derivable, then $\Gamma \subseteq \mathcal{L}^i$ and $\Gamma^{\downarrow} \vdash_{\mathsf{L}} \varphi$.
- Chaining and defaults: a cut into DL1 is simulated by Lio^i and $\mathsf{IO2}'$.
- Constraints are pushed up to the leaves, where $\mathsf{IO1}'$ lets them clash with outputs:

$$\dfrac{\dfrac{}{\Pi^c \Rightarrow \neg\psi^c}\,\mathsf{Ax} \qquad \dfrac{\dfrac{\Delta' \Rightarrow \theta^x \qquad \dfrac{}{\xi^o, \Delta'' \Rightarrow \psi^o}\,\mathsf{Ax}}{\Delta', \Delta'', (\theta, \xi) \Rightarrow \psi^o}\,\mathsf{Lio}^{\mathsf{x}}}{\Delta', \Delta'', (\theta, \xi), \neg\psi^c \Rightarrow}\,\mathsf{IO1}}{\Pi^c, \Delta', \Delta'', (\theta, \xi) \Rightarrow}\,\mathsf{Cut} \quad \rightsquigarrow \quad \dfrac{\Delta' \Rightarrow \theta^x \qquad \dfrac{\dfrac{}{\Pi^o, \xi^o, \Delta'' \Rightarrow}\,\mathsf{Ax}}{\Pi^c, \xi^o, \Delta'' \Rightarrow}\,\mathsf{IO1}'}{\Pi^c, \Delta', \Delta'', (\theta, \xi) \Rightarrow}\,\mathsf{Lio}^{\mathsf{x}}$$

Corollary

$\vdash_{\mathsf{AC}_{\mathcal{S}}} \Gamma \Rightarrow \Lambda$ iff $\vdash_{\mathsf{AC}^{\mathsf{cf}}_{\mathcal{S}'}} \Gamma, \Theta \Rightarrow \Lambda$ for some $\Theta \subseteq \{(\top, \top)\}$. (why $(\top, \top)$: later)

Outline

1 Motivation

2 The AC framework

3 Cut-free AC

4 Reasoning by cases

5 Analyticity

6 Conclusion

Reasoning by cases, cut-free

In out_2 and out_4, Cases splits the input, BDet triggers rules and Cut propagates their consequents. A cut-free rule must do all three at once: a parallel composition of Lio^i instances.

Left introduction by cases

$$\frac{\Gamma^i \Rightarrow (\bigwedge \Gamma_1 \vee \cdots \vee \bigwedge \Gamma_n)^i \qquad \Gamma_1^i \Rightarrow \varphi_1^i \quad \psi_1^o, \Lambda_1 \Rightarrow \Lambda \qquad \cdots \qquad \Gamma_n^i \Rightarrow \varphi_n^i \quad \psi_n^o, \Lambda_n \Rightarrow \Lambda}{\Gamma^i, \Lambda_1, \ldots, \Lambda_n, (\varphi_1, \psi_1), \ldots, (\varphi_n, \psi_n) \Rightarrow \Lambda}\ \mathsf{LiOR}$$

- $\Gamma^i \Rightarrow (\bigwedge \Gamma_1 \vee \cdots \vee \bigwedge \Gamma_n)^i$: the input is covered by the cases $\Gamma_1, \ldots, \Gamma_n$;
- $\Gamma_j^i \Rightarrow \varphi_j^i$: in case j, the rule (φ_j, ψ_j) is triggered;
- $\psi_j^o, \Lambda_j \Rightarrow \Lambda$: in case j, Λ follows from the consequent ψ_j (and Λ_j).

For $n = 1$ we get back Lio^i. The system $\mathsf{AC}^{\mathsf{cf}}_{out_2}$ consists of Ax, Taut, LiOR, IO1, IO1′, IO2; $\mathsf{AC}^{\mathsf{cf}}_{out_4}$ adds SDet.

LiOR at work

In normal form (next slide) the triggering premises are identities $\varphi_j^i \Rightarrow \varphi_j^i$: we omit them.

Example (Reasoning by cases (out_2 and out_4))

$$\frac{(p \vee q)^i \Rightarrow (p \vee q)^i \qquad r^o \Rightarrow r^o \qquad r^o \Rightarrow r^o}{(p \vee q)^i, (p, r), (q, r) \Rightarrow r^o}\ \text{LiOR}$$

Example (Cases and chaining (out_4 only; π_q is symmetric to π_p))

$$\frac{(p \vee q)^i \Rightarrow (p \vee q)^i \qquad \overset{\pi_p}{q^o, (q, p) \Rightarrow (p \wedge q)^o} \qquad \overset{\pi_q}{p^o, (p, q) \Rightarrow (p \wedge q)^o}}{(p \vee q)^i, (p, q), (q, p) \Rightarrow (p \wedge q)^o}\ \text{LiOR}$$

$$\pi_p\text{:}\quad \frac{\dfrac{q^i \Rightarrow q^i \qquad p^o, q^o \Rightarrow (p \wedge q)^o}{q^i, q^o, (q, p) \Rightarrow (p \wedge q)^o}\ \text{LiOR}}{q^o, (q, p) \Rightarrow (p \wedge q)^o}\ \text{SDet}$$

Without SDet, i.e. in out_2, the latter sequent is not derivable.

Structure of derivations and Cut elimination

Let $\mathcal{S}$ be the cut-free system for out_2 or out_4.

Lemma

(i) Purity: $\Gamma^x \Rightarrow \Lambda^x$ is derivable iff $\Gamma \vdash_{\mathsf{L}} \Lambda$; if $\Gamma \Rightarrow \varphi^i$ is derivable, then $\Gamma \subseteq \mathcal{L}^i$.

(ii) Normal form: every LiOR instance can be replaced by one with $\Gamma_j = \{\varphi_j\}$, i.e. with case split $\Gamma^i \Rightarrow (\varphi_1 \vee \cdots \vee \varphi_n)^i$ and identities as triggering premises.

(iii) Shape: if $\Gamma \cap \mathcal{L}^r \neq \varnothing$, every derivation of $\Gamma \Rightarrow \theta^o$ ends with a LiOR instance followed by a (possibly empty) sequence of SDet instances.

Theorem

If Cut is admissible in L, then every Cut on a formula other than $(\top, \top)$ is admissible in $\mathsf{AC}^{\mathsf{cf}}_{\mathcal{S}}$.

Proof idea: induction on the sum of the heights of the Cut premises.

Recovering AC – modulo $(\top, \top)$

Lemma

In $\mathsf{AC}^{\mathsf{cf}}_{\mathcal{S}}$, Cases (with rules on both sides) is admissible. With Cut, Cases is what yields admissibility of the generalized rule

$$\frac{\Gamma^i \Rightarrow (\bigwedge \Gamma_1 \vee \cdots \vee \bigwedge \Gamma_n)^i \qquad \{\Gamma_j^i, \Pi_j \Rightarrow \varphi_j^x\}_{1 \leq j \leq n} \qquad \{\psi_j^o, \Lambda_j \Rightarrow \Lambda\}_{1 \leq j \leq n}}{\Gamma^i, \Pi_1, \ldots, \Pi_n, \Lambda_1, \ldots, \Lambda_n, (\varphi_1, \psi_1), \ldots, (\varphi_n, \psi_n) \Rightarrow \Lambda}\ \mathsf{LiOR}^{\mathsf{x}}$$

with $x = i$, and $x = o$ in out_4. In particular, BDet and Lio^x are admissible.

Theorem (equivalence)

$\vdash_{\mathsf{AC}_{\mathcal{S}}} \Gamma \Rightarrow \Lambda$ iff $\vdash_{\mathsf{AC}^{\mathsf{cf}}_{\mathcal{S}'}} \Gamma, \Theta \Rightarrow \Lambda$ for some $\Theta \subseteq \{(\top, \top)\}$.

Why $(\top, \top)$? AC derives $(p \vee q)^i, (p, r) \Rightarrow (q \vee \neg q)^o$: Cases on $p^i, (p, r) \Rightarrow (q \vee \neg q)^o$ and $q^i, (\top, \top) \Rightarrow (q \vee \neg q)^o$, then Cut with Taut. In the cut-free case, we only get $(p \vee q)^i, (p, r), (\top, \top) \Rightarrow (q \vee \neg q)^o$.

Harmless: $(\top, \top)$ is in every maximal consistent set (for consistent $\mathcal{C}$): let $(\top, \top) \in \mathcal{R}$.

Throughput

Throughput (TP) has a cut-free counterpart: the inverse of SDet, with no side condition.

$$\frac{\varphi^o, \Gamma \Rightarrow \Lambda}{\varphi^i, \Gamma \Rightarrow \Lambda}\ \text{iSDet}$$

For out_1^+, out_3^+ and default logic with TP, add iSDet; for $out_2^+ = out_4^+$, allow additional cases in LiOR:

$$\frac{\Gamma^i \Rightarrow (\bigwedge \Gamma_1 \vee \cdots \vee \bigwedge \Gamma_n)^i \qquad \{\Gamma_j^i \Rightarrow \varphi_j^i \quad \psi_j^o, \Lambda_j \Rightarrow \Lambda\}_{1 \le j \le m} \qquad \{\Gamma_k^i, \Lambda_k \Rightarrow \Lambda\}_{m < k \le n}}{\Gamma^i, \Lambda_1, \ldots, \Lambda_n, (\varphi_1, \psi_1), \ldots, (\varphi_m, \psi_m) \Rightarrow \Lambda}\ \text{LiOR}^+$$

Example ($\neg q^i, (p, q) \Rightarrow \neg p^o$: derivable in out_4^+, not in out_4)

$$\frac{\Rightarrow (p \vee \neg p)^i \qquad \dfrac{\dfrac{q^o, \neg q^o \Rightarrow \neg p^o}{q^o, \neg q^i \Rightarrow \neg p^o}\ \text{iSDet} \qquad \dfrac{\neg p^o \Rightarrow \neg p^o}{\neg p^i \Rightarrow \neg p^o}\ \text{iSDet}}{\neg q^i, (p, q) \Rightarrow \neg p^o}\ \text{LiOR}^+$$

Cut elimination and the equivalence results extend to all throughput variants.

Outline

1 Motivation

2 The AC framework

3 Cut-free AC

4 Reasoning by cases

5 Analyticity

6 Conclusion

Generalized subformula property

Definition (generalized subformulas)

$gsub(\Gamma^{\downarrow})$ is the least set containing $\Gamma^{\downarrow}$ and closed under: $\neg\varphi \mapsto \varphi$; $(\varphi, \psi) \mapsto \varphi, \psi$; $\neg(\varphi, \psi) \mapsto \varphi, \neg\psi, (\varphi, \psi)$; $(\varphi_1, \psi_1), \ldots, (\varphi_n, \psi_n) \mapsto \varphi_1 \vee \cdots \vee \varphi_n$.

Proposition (subformula property)

If π is a $\mathsf{AC}^{\mathsf{cf}}_{\mathcal{S}}$-proof of $\Gamma \Rightarrow \Delta$ (with LiOR in normal form), every formula occurring in $\pi^{\downarrow}$ belongs to $gsub(\Gamma^{\downarrow} \cup \Delta^{\downarrow})$. (The disjunctive clause serves the case splits of LiOR; for $out_2^+ = out_4^+$ a finite variant holds.)

Corollary (decidability)

Derivability of arguments is decidable in each $\mathsf{AC}^{\mathsf{cf}}_{\mathcal{S}}$ – and hence in each $\mathsf{AC}_{\mathcal{S}}$.

Root-first search ranges over a finite pool of labelled formulas, leaves are decided in L: arguments are generated and checked on demand.

Craig interpolation

Theorem

Let $\mathcal{S}$ be the cut-free system for out_1, out_3, out_1^+ or out_3^+. If $\Gamma \Rightarrow \theta^x$ is derivable in $\mathsf{AC}^{\mathsf{cf}}_{\mathcal{S}}$, then there is $\sigma \in \mathcal{L}^{dac}$ such that

$$At(\sigma^{\downarrow}) \subseteq At(\Gamma^{\downarrow}) \cap At(\theta), \qquad \vdash \Gamma \Rightarrow \sigma, \qquad \vdash \sigma \Rightarrow \theta^x.$$

Proof idea: induction on the derivation.

- Ax: interpolation in L, within a single label.
- Lio^x: the interpolant γ of the right premise works for the conclusion:

$$\frac{\Gamma_1 \Rightarrow \varphi^x \qquad \psi^o, \Gamma_2 \Rightarrow \gamma}{\Gamma_1, \Gamma_2, (\varphi, \psi) \Rightarrow \gamma}\ \mathsf{Lio}^{\mathsf{x}} \qquad \gamma \Rightarrow \theta^x$$

Limits: attacks need not have interpolants – e.g. $p^i, \neg q^c \Rightarrow \neg(p, q)$, since no connective combines input and constraint information.

Outline

1 Motivation

2 The AC framework

3 Cut-free AC

4 Reasoning by cases

5 Analyticity

6 Conclusion

Conclusion and future work

Summary

- Defaults and norms as left-introduced formulas: Lio and LiOR internalize detachment, chaining and reasoning by cases.
- Modular Cut elimination for the calculi characterizing normal default logic and all constrained I/O logics: a single induction, relative to the base logic.
- Analyticity: generalized subformula property and decidable derivability; Craig interpolation for the non-disjunctive systems.

Future work

- cut-free calculi vs. decomposition trees: which method for which task?
- proof search for nonmonotonic inference (credulous and skeptical): monotonic search plus consistency checks; underivability via refutation calculi;
- dialogues and explanations in which arguments are generated, queried and retracted on the fly;
- other base logics; disjunctive and prioritized defaults, full default logic, causal reasoning.

References I

[1] Arieli O., Borg A., Heyninck J., Straßer C., *Logic-based approaches to formal argumentation*, in *Handbook of Formal Argumentation*, vol. 2, College Publications, 2021.

[2] van Berkel K., Straßer C., *Reasoning with and about norms in logical argumentation*, *COMMA 2022*.

[3] van Berkel K., *A logical analysis of normative reasoning: agency, action, and argumentation*, PhD thesis, TU Wien, 2023.

[4] van Berkel K., Straßer C., Zhou Z., *Towards an argumentative unification of default reasoning*, *COMMA 2024*.

[5] van Berkel K., Straßer C., Zhou Z., *Hypothesis-driven disjunctive reasoning in logical argumentation*, *LORI 2025*.

[6] van Berkel K., Sabatini A., *Proof-search for normative and doxastic reasoning and its use in logical argumentation*, *KR 2026*.

References II

[1] Bonatti P., Olivetti N., *Sequent calculi for propositional nonmonotonic logics*, *ACM Transactions on Computational Logic*, 3, 2002.

[2] Ciabattoni A., Rozplokhas D., *Streamlining input/output logics with sequent calculi*, *KR 2023*.

[3] Dung P.M., *On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games*, *Artificial Intelligence*, 77, 1995.

[4] Makinson D., van der Torre L., *Input/output logics*, *Journal of Philosophical Logic*, 29, 2000.

[5] Makinson D., van der Torre L., *Constraints for input/output logics*, *Journal of Philosophical Logic*, 30, 2001.

[6] Piazza M., Sabatini A., *Hypersequent calculi for propositional default logics*, *ACM Transactions on Computational Logic*, 26, 2025.

[7] Piazza M., Sabatini A., *A unified framework for input/output and default logics via hypersequents*, *Journal of Logic and Computation*, 36, 2026.

[8] Reiter R., *A logic for default reasoning*, *Artificial Intelligence*, 13, 1980.

Thank you for your attention!

Backup: cuts on $(\top, \top)$

In AC, a Cut with Taut acts as a hidden weakening:

$$\dfrac{\dfrac{}{\Rightarrow (\top,\top)}\ \text{Taut} \qquad \dfrac{p^i \Rightarrow \top^i \qquad \top^o, q^o \Rightarrow q^o}{p^i, q^o, (\top,\top) \Rightarrow q^o}\ \text{Lio}^{\text{i}}\ \text{(adm.)}}{p^i, q^o \Rightarrow q^o}\ \text{Cut}$$

- Cut-free, $p^i, q^o \Rightarrow q^o$ is not derivable: an argument with mixed labels and a labelled conclusion needs a rule among its reasons, while $p^i, q^o, (\top, \top) \Rightarrow q^o$ is derivable.
- Disjunctive analogue: $(p \vee q)^i, (p, r) \Rightarrow (q \vee \neg q)^o$ vs. $(p \vee q)^i, (p, r), (\top, \top) \Rightarrow (q \vee \neg q)^o$.
- Cuts on $(\top, \top)$ are the only ones that may resist elimination: hence the equivalences hold modulo $(\top, \top)$.
- No loss for argumentation: $(\top, \top)$ detaches only tautologies and belongs to every maximal consistent set (for consistent $\mathcal{C}$), so we may assume $(\top, \top) \in \mathcal{R}$.
- With throughput, iSDet derives $p^i, q^o \Rightarrow q^o$ from $p^o, q^o \Rightarrow q^o$ directly.

Backup: Cases and inapplicability

Lemma

Let $\mathcal{S}$ be the cut-free system for out_2 or out_4.

(i) Inversion: if $\Gamma \Rightarrow \neg(\varphi, \psi)$ is derivable, so is $\Gamma, (\varphi, \psi) \Rightarrow$.

(ii) Constraints: if $\Gamma \Rightarrow \varphi^o$ is derivable, so is $\Gamma \setminus \mathcal{L}^c \Rightarrow \varphi^o$ (height-preserving); if $\Gamma \Rightarrow$ is derivable, so is $\Gamma \setminus \mathcal{L}^c \Rightarrow \varphi^o$ for some $\varphi \vdash_{\mathsf{L}} \neg \bigwedge(\Gamma \cap \mathcal{L}^c)^{\downarrow}$.

Cases is permuted upwards; e.g. when a premise ends with IO2:

$$\dfrac{\dfrac{\varphi^i, \Gamma, (\chi, \xi) \Rightarrow}{\varphi^i, \Gamma \Rightarrow \neg(\chi, \xi)}\,\mathsf{IO2} \qquad \psi^i, \Delta \Rightarrow \neg(\chi, \xi)}{(\varphi \vee \psi)^i, \Gamma, \Delta \Rightarrow \neg(\chi, \xi)}\,\mathsf{Cases} \quad \rightsquigarrow \quad \dfrac{\dfrac{\varphi^i, \Gamma, (\chi, \xi) \Rightarrow \qquad \dfrac{\psi^i, \Delta \Rightarrow \neg(\chi, \xi)}{\psi^i, \Delta, (\chi, \xi) \Rightarrow}\,(i)}{(\varphi \vee \psi)^i, \Gamma, \Delta, (\chi, \xi) \Rightarrow}\,\mathsf{Cases}}{(\varphi \vee \psi)^i, \Gamma, \Delta \Rightarrow \neg(\chi, \xi)}\,\mathsf{IO2}$$

- IO1: the constraint is moved below Cases, using (ii).
- Both premises end with LiOR: the two instances are merged into a single LiOR.

Backup: attackers in default vs. I/O reasoning

Let $\mathcal{I} = \{p^i\}$, $\mathcal{R} = \{(p, q), (q, r), (p, \neg r)\}$, $\mathcal{C} = \{\top^c\}$: the three rules are jointly incoherent (r vs. $\neg r$).

- Both styles derive $p^i, (p, q), (q, r) \Rightarrow \neg(p, \neg r)$ and $p^i, (p, q), (p, \neg r) \Rightarrow \neg(q, r)$.
- Only the I/O style (IO2) derives $p^i, (p, \neg r), (q, r) \Rightarrow \neg(p, q)$: in default logic only triggered rules may serve as reasons, and (q, r) is not triggered by p and the output of $(p, \neg r)$.

A cut-free attacker in the default-logic system:

$$\dfrac{\dfrac{p^i \Rightarrow p^i \qquad \dfrac{\dfrac{q^o \Rightarrow q^o \qquad r^o, \neg r^o \Rightarrow}{q^o, \neg r^o, (q, r) \Rightarrow}\,\text{Lio}^\circ}{p^i, \neg r^o, (p, q), (q, r) \Rightarrow}\,\text{Lio}^{\text{i}}}{p^i, (p, q), (q, r), (p, \neg r) \Rightarrow}\,\text{Lio}^{\text{i}}}{p^i, (p, q), (q, r) \Rightarrow \neg(p, \neg r)}\,\text{IO2}'$$

The premise of IO2′ has an SDet-free derivation, as required.