Implicit Disagreement in Higher-Order Logic

Same conclusion, different reasons: agents, coalitions and explanations as a LogiKEy domain theory on the Isabelle session `AA_Fixed`

Christoph Benzmüller and Luca Pasetto
joint work with Luca Redondi and Réka Markovich

LogiKEy course, November 2026

From the library to a domain theory

Last lecture: an object logic (L1)

- Abstract argumentation in HOL: sets are predicates, a framework is a universe with an attack relation
- Dung's semantics as predicates on predicates: "preferred" is a quantifier over sets
- The textbook meta-theory proved, extensions generated with Nitpick
- The Isabelle session `AA_Fixed`, after Fuenmayor and Steen[1]

Today: a domain theory (L2) and a case study (L3)

- A question from political and legal debate: two parties reach the **same conclusion for different reasons**
- Agents, coalitions, explanations and disagreement, all written with the predicates of the library
- A case study whose answers are lemmas
- The session `Implicit_Disagreement`: five theories on top of `AA_Fixed`

The lecture follows the paper *Shared Conclusions without Shared Explanations: Mechanizing Implicit Disagreement in Abstract Argumentation with LogiKEy* (Redondi, Pasetto, Benzmüller, Markovich; under review). **All Isabelle text on these slides** is taken from the session.

[1] Fuenmayor and Steen, *A Flexible Approach to Argumentation Framework Analysis using Theorem Proving*, LNGAI 2021; Steen and Fuenmayor, *A formalization of abstract argumentation in higher-order logic*, Journal of Logic and Computation 34(2) 2024.

Outline

1. A case study: agreeing for different reasons
2. Agents, coalitions and three kinds of agreement
3. Implicit disagreement, and where it sits
4. The embedding: frameworks, agents, inference
5. The embedding: explanations and disagreement
6. The case study, lemma by lemma
7. Lessons and outlook

01

A case study: agreeing for different reasons

Case study: Julia and Emma on abortion

Two Members of Parliament, from different parties, both defend abortion rights.

Julia

Abortion should be permissible, **because the fetus is not a person.**

Emma

Abortion should be permissible, **because of women's autonomy over their own bodies.**

Same conclusion: they form a coalition.

What if the fetus is viable?

At a committee hearing the coalition is asked about abortion when the fetus could survive outside the womb.

Emma

When the fetus is viable, delivery should be preferred over abortion: that way we respect **autonomy** and the fetus' **right to life**.

Julia

There is **no right to life** involved, because the fetus is not a person. And if it were a person, the right to life would **outweigh autonomy**!

The coalition fractures on a question it never discussed.

Agreeing that without agreeing because

- Julia and Emma **agree that** abortion is permissible
- They do not **agree because**: they disagree on *why* it is permissible[2]
- **Implicit disagreement** = agreement that + disagreement because

It is not a defect

Coalitions in parliaments, courts and committees are built on shared conclusions. Requiring shared reasons would make most of them impossible.

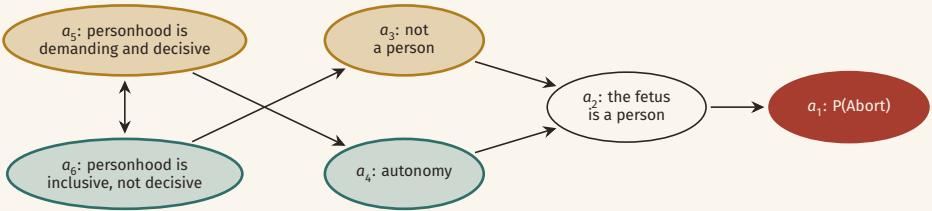
But it is a risk

A new case, or an attack on one party's reasons, can turn the hidden disagreement into an open one. The parties should know where it sits.

A formal tool to identify, locate and analyze implicit disagreement

[2] Baumtrog, *The basic components of agreement*, Informal Logic 2023; Baumtrog, *Deep agreements*, Episteme 2025.

The case study as an argumentation graph



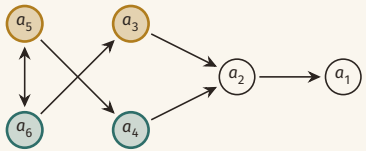
An arrow is an attack. • Julia's arguments • Emma's arguments • the argument for the conclusion at issue

- a_1: by default everything is permissible, so abortion is: $Con(a_1)$ = P(Abort)
- a_2: the fetus is a person, so abortion is not permissible
- a_3 and a_4: the two replies to a_2
- a_5, Julia's stance: a person has a right to life that autonomy cannot outweigh. It attacks a_4
- a_6, Emma's stance: a person's right to life does not always take priority. It attacks a_3
- The two stances are incompatible: $a_5 \leftrightarrows a_6$

Only the shape of the debate is modelled. Real instances: Tooley (1972), Thomson (1971).

What do we want to compute? Four questions

- **Q1** On what do they agree: the conclusion, the argument a_1, or the whole explanation?
- **Q2** Where is the disagreement: which arguments carry it?
- **Q3** Which arguments are **weak spots**, whose loss breaks the coalition? Which ones would make it more robust if they were removed?
- **Q4** Is the disagreement already in the coalition, or only potential?



Julia = $\{a_1, a_2, a_3, a_5\}$

Emma = $\{a_1, a_2, a_4, a_6\}$

The questions are answered three times in this lecture: **on paper** (parts 2 and 3), as **definitions in HOL** (parts 4 and 5), and as **lemmas about the case study** (part 6).

02

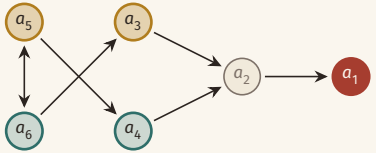
Agents, coalitions and three kinds of agreement

Recap: preferred extensions

A set of arguments E ...

- **defends** a: every attacker of a is attacked by a member of E
- is **admissible**: conflict-free, and it defends its members
- is **complete**: admissible, and it contains all it defends
- is **preferred**: a $\subseteq$-maximal complete set

- The **grounded** extension is **empty**: it accepts nothing
- The preferred extensions are the **alternative stances**. We use preferred semantics throughout[3]



Semantics	Extensions of the example
complete	$\emptyset$, $\{a_1, a_3, a_5\}$, $\{a_1, a_4, a_6\}$
grounded	$\emptyset$
preferred	$\{a_1, a_3, a_5\}$, $\{a_1, a_4, a_6\}$

[3] Dung, *On the acceptability of arguments*, Artificial Intelligence 1995.

Three inference relations

A function *Con* maps each argument to its conclusion. For a framework K and a conclusion φ:

- $K \vdash_{\cup} \varphi$, **credulous**: some preferred extension contains an argument for φ
- $K \vdash_{c} \varphi$, **shared conclusion**: every preferred extension contains an argument for φ
- $K \vdash_{a} \varphi$, **shared argument**: one argument for φ is in every preferred extension

$$\vdash_a \implies \vdash_c \implies \vdash_{\cup}$$

- $\vdash_c$ and $\vdash_a$ are both skeptical. They differ in the order of two quantifiers: *every extension has an argument* versus *one argument is in every extension*

A shared conclusion with no shared argument

Preferred extensions $\{a\}$ and $\{b\}$. Both conclude ψ, so $K \vdash_c \psi$. No argument is in both, so $K \nvdash_a \psi$.

In the running example a_1 is in both preferred extensions: $K \vdash_a$ P(Abort).

Agents: fragments of the graph

Agent. $\alpha = \langle Arg_\alpha, Att_\alpha \rangle$ is an agent in $\langle Arg, Att \rangle$ iff (i) $Arg_\alpha \subseteq Arg$, and (ii) $\langle a, b \rangle \in Att_\alpha$ exactly when $\langle a, b \rangle \in Att$ and $a, b \in Arg_\alpha$.

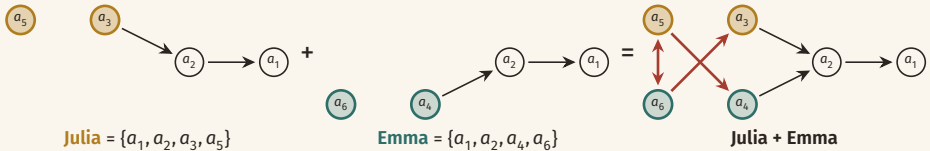
Julia = $\{a_1, a_2, a_3, a_5\}$

Emma = $\{a_1, a_2, a_4, a_6\}$

- Arg_α are the arguments the agent is **aware of**. The attacks come with the arguments: who knows a and b knows whether a attacks b
- **No closure condition**: an agent may miss an attacker of one of its arguments, or hold an argument whose targets it has not met. Julia holds a_5, which attacks nothing in her fragment
- Alone, each agent has a single preferred extension, and it contains a_1: each of them accepts P(Abort)

Coalitions: reasoning together

Coalition. $\alpha + \beta$ has the arguments $Arg_\alpha \cup Arg_\beta$, and $\langle a, b\rangle$ is an attack of $\alpha + \beta$ iff $\langle a, b\rangle \in Att$ and $a, b \in Arg_\alpha \cup Arg_\beta$.

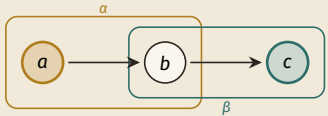


Julia = $\{a_1, a_2, a_3, a_5\}$ **Emma** = $\{a_1, a_2, a_4, a_6\}$ **Julia + Emma**

- The coalition has **all** the attacks among the pooled arguments, also those between arguments that were held by different agents
- The **red attacks** exist only in the coalition: a_5 on a_4, a_6 on a_3, and the conflict between a_5 and a_6. This is where the disagreement of Julia and Emma becomes visible
- A coalition is an agent again, so coalitions of coalitions need no new definition

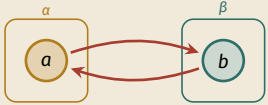
Coalitions gain and lose conclusions

A conclusion gained



- $\alpha = \{a, b\}$ has no argument for $Con(c)$
- $\beta = \{b, c\}$ accepts b, which defeats c
- In $\alpha + \beta$ the unattacked a defeats b, and c is reinstated: the coalition derives $Con(c)$, under $\vdash_c$ and under $\vdash_a$

A conclusion lost



- $\alpha = \{a\}$ derives $Con(a)$, and $\beta = \{b\}$ derives $Con(b)$
- In $\alpha + \beta$ the two arguments attack each other: the preferred extensions are $\{a\}$ and $\{b\}$
- The coalition derives neither conclusion skeptically

- Forming a coalition is **more than merging conclusions**: the agents reason together, on the pooled graph

Agreement that, agreement because

Let X be an agent or a coalition.

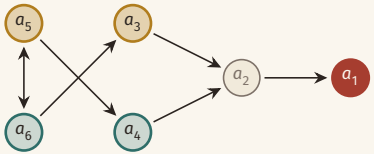
- X **agrees that** φ iff $X \vdash_c \varphi$
- A **proximal explanation** of φ is an argument that concludes φ: the immediate reason
- X **agrees because** on φ iff $X \vdash_a \varphi$: a proximal explanation is shared

Julia + Emma

✓ They agree that P(Abort): both preferred extensions have an argument for it

✓ They agree because: it is the same argument, a_1

? Yet they do not agree on **why a_2 is to be discarded**. Is that not part of the reason?



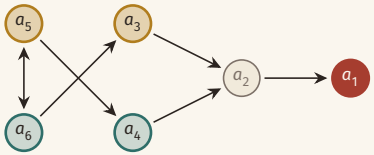
- $\{a_1, a_3, a_5\}$: preferred extension 1
- $\{a_1, a_4, a_6\}$: preferred extension 2
- a_1: in both
- a_2: rejected in both

Explanations

Explanation. A set of arguments $\mathcal{E}$ is an explanation of ψ iff

(i) some $a \in \mathcal{E}$ has $Con(a) = \psi$;
(ii) $\mathcal{E}$ is conflict-free and defends all its arguments;
(iii) no proper subset of $\mathcal{E}$ satisfies (i) and (ii).

- The argument for ψ together with **what it takes to defend it**, and nothing more[4]
- $\{a_1\}$ is not an explanation of P(Abort): nothing answers a_2. Nor is $\{a_1, a_3\}$: nothing answers a_6
- $\{a_1, a_3, a_5\}$ and $\{a_1, a_4, a_6\}$ are the two explanations: a reply to a_2, and the stance that protects it



- Julia's explanation $\{a_1, a_3, a_5\}$
- Emma's explanation $\{a_1, a_4, a_6\}$
- a_1 belongs to both

Here the explanations are the preferred extensions. In general they are smaller.

[4] Rotolo and Sartor, *Argumentation and explanation in the law*, Frontiers in Artificial Intelligence 2023; Fan and Toni, *On computing explanations in argumentation*, AAAI 2015.

Shared explanations: three levels of agreement

- $K \vdash_{\mathcal{E}} \psi$, **shared explanation**: one explanation of ψ is contained in every preferred extension
- X **strongly agrees because** on ψ iff $X \vdash_{\mathcal{E}} \psi$

Level of agreement		In every preferred extension there is ...	Julia + Emma
agrees that	$\vdash_c$	some argument for ψ	✓
agrees because	$\vdash_a$	the same argument for ψ	✓ a_1
strongly agrees because	$\vdash_{\mathcal{E}}$	the same explanation of ψ	✗

$$\vdash_{\mathcal{E}} \implies \vdash_a \implies \vdash_c \implies \vdash_{\cup}$$

- No implication can be reversed: the two mutually attacking arguments separate $\vdash_c$ from $\vdash_a$, the running example separates $\vdash_a$ from $\vdash_{\mathcal{E}}$
- Agreement because always comes with an agreement that, as Baumtrog requires
- **Q1 answered**: Julia and Emma agree on the conclusion and on the argument a_1, not on the explanation

03

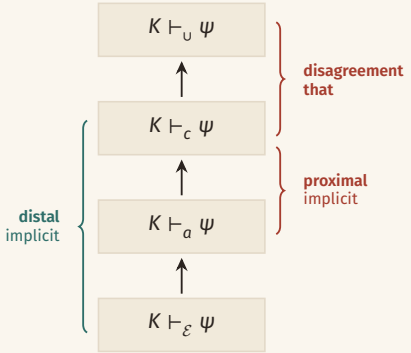
Implicit disagreement, and where it sits

Disagreement, explicit and implicit

For a framework K, such as a coalition $\alpha + \beta$:

- **Disagreement that** ψ: $K \vdash_{\cup} \psi$ and $K \nvdash_{c} \psi$. We write $\psi \in \mathrm{Dis}(K)$. Some stance supports ψ, another does not
- **Proximal implicit disagreement** on ψ: $K \vdash_{c} \psi$ and $K \nvdash_{a} \psi$. We write $\psi \in \mathrm{ImpDis}^{p}(K)$
- **Distal implicit disagreement** on ψ: $K \vdash_{c} \psi$ and $K \nvdash_{\mathcal{E}} \psi$. We write $\psi \in \mathrm{ImpDis}^{d}(K)$

- Proximal implies distal: $\mathrm{ImpDis}^{p}(K) \subseteq \mathrm{ImpDis}^{d}(K)$
- Implicit disagreement and disagreement that exclude each other: one needs $\vdash_{c}$, the other its failure



The inference relations, from weak to strong; an arrow is an implication.

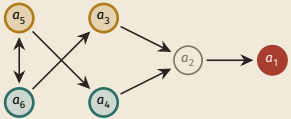
A brace marks a gap: the upper relation holds, the lower one fails.

Proximal versus distal

Proximal: the immediate reasons differ

- Each party has its own argument for ψ, and the two arguments exclude each other
- $K \vdash_c \psi$, but no argument is shared: $\psi \in \text{ImpDis}^p(K)$, and therefore $\psi \in \text{ImpDis}^d(K)$
- It is not a disagreement that: $\psi \notin \text{Dis}(K)$

Distal only: the deeper reasons differ



- The argument a_1 for P(Abort) is shared, so there is no proximal implicit disagreement
- The explanations differ: P(Abort) $\in \text{ImpDis}^d$(Julia + Emma)

- **Proximal**: the disagreement touches the very argument for ψ. Asking "why ψ?" reveals it
- **Distal**: both parties give the same answer, and the disagreement sits further back. It goes unnoticed

Potential versus actual

Implicit disagreement = **agreeing that** + **disagreeing because**. Where is the agreement evaluated: in each agent, or in the coalition?

		Agreeing that	
		Potential: $\alpha \vdash_c \psi$ and $\beta \vdash_c \psi$	**Actual:** $\alpha + \beta \vdash_c \psi$
Disagreeing because	**Distal:** $\alpha + \beta \nvdash_{\mathcal{E}} \psi$	$\psi \in \mathrm{ImpDis}^d_p(\alpha + \beta)$ ✓ Julia + Emma on P(Abort)	$\psi \in \mathrm{ImpDis}^d(\alpha + \beta)$ ✓ Julia + Emma on P(Abort)
	Proximal: $\alpha + \beta \nvdash_a \psi$	$\psi \in \mathrm{ImpDis}^p_p(\alpha + \beta)$ ✗ Julia + Emma: a_1 is shared	$\psi \in \mathrm{ImpDis}^p(\alpha + \beta)$ ✗ Julia + Emma: a_1 is shared

- Superscript: proximal (p) or distal (d). Subscript p: potential. In each column, proximal implies distal
- The **actual** notions depend only on the pooled graph. The **potential** ones keep the two agents apart, and do not require that the coalition still derives ψ

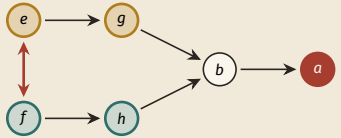
Implicit disagreement as the price of a coalition

Potential and actual are independent. Three situations:

- **Potential only**: each agent derives ψ alone, for its own reasons. Pooled, their arguments defeat each other, and the coalition no longer derives ψ
- **Potential and actual**: the agents agree alone and keep agreeing together. Julia and Emma on P(Abort)
- **Actual only**: neither agent derives ψ alone. They need the coalition to argue for ψ, and pay for it with an implicit disagreement

Potential only, in four arguments: $\alpha = \{a, d\}$, $\beta = \{b, c\}$, with a and b for ψ, c attacking a and d attacking b.

Actual only: the coalition-only example

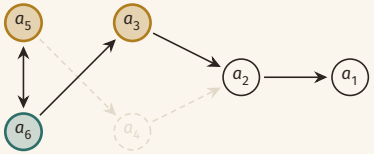


- $\alpha = \{a, b, e, g\}$, $\beta = \{a, b, f, h\}$
- Alone, α's own e defeats its defender g: α accepts b and rejects a. The same for β with f and h
- Together, e and f attack each other. Preferred extensions: $\{a, e, h\}$ and $\{a, f, g\}$
- $\alpha + \beta \vdash_a Con(a)$, with two explanations: distal, actual, not potential

Locating the disagreement: counterfactual deletion

The definitions so far answer yes or no. To find out **where** a coalition is fragile, ask what would happen without one of its arguments.

- **Deletion** $K(\bar{a})$: remove a and all attacks from and to a, then evaluate the framework again
- The simplest operation of argumentation dynamics. What it tells us is a contrastive, counterfactual explanation of an acceptance status[5]
- It has the same effect as **adding** an unattacked attacker of a, which is how a new objection enters a debate[6]
- Next: weak spots, antagonists, the locus



$K(\bar{a}_4)$: the preferred extensions are $\{a_1, a_3, a_5\}$ and $\{a_2, a_6\}$.

P(Abort) is now only credulously supported.

[5] Doutre and Mailly, *Constraints and changes: A survey of abstract argumentation dynamics*, Argument & Computation 2018; Čyras et al., *Argumentative XAI: A survey*, IJCAI 2021.

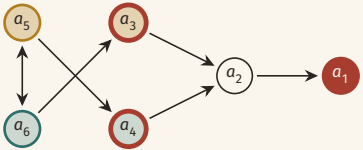
[6] Sakama, *Counterfactual Reasoning in Argumentation Frameworks*, COMMA 2014.

Weak spots

Weak spot. An argument a is a weak spot of $\alpha + \beta$ with respect to ψ iff

$$\alpha + \beta \vdash_c \psi \quad \text{and} \quad \psi \in \mathrm{Dis}\big((\alpha + \beta)(\bar{a})\big).$$

- a_3 **and** a_4 are the weak spots for P(Abort). Without a_4 the extension $\{a_2, a_6\}$ rejects a_1, while $\{a_1, a_3, a_5\}$ keeps it: the coalition splits on the issue itself
- a_1 is **not** a weak spot. Without it the issue disappears, the coalition does not fracture: credulous support must survive
- This is the committee hearing: viability is an unanswered attacker of Emma's a_4, which has the effect of deleting it. The implicit disagreement has become a disagreement that



thick red ring: weak spot

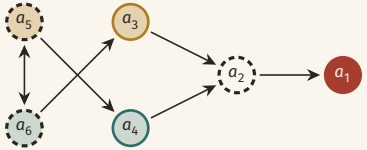
$K(\bar{a}_3)$: $\{a_1, a_4, a_6\}$ and $\{a_2, a_5\}$

Antagonists

Antagonist. An argument a of $\alpha \dashrightarrow \beta$ is an antagonist with respect to ψ iff

$$\psi \in \mathrm{ImpDis}^{d}(\alpha \dashrightarrow \beta) \quad \text{and} \quad (\alpha \dashrightarrow \beta)(\bar{a}) \vdash_{\mathcal{E}} \psi.$$

- Deleting an antagonist turns distal implicit disagreement into a **shared explanation**: the coalition becomes more robust
- a_2, a_5 **and** a_6 are the antagonists for P(Abort)
- Without a_2 nothing attacks a_1. Without a_5 Emma's reply a_4 is unattacked and settles the matter, and symmetrically without a_6
- An antagonist is never a weak spot: after its deletion $\vdash_{\mathcal{E}}$ holds, hence $\vdash_{c}$, so there is no disagreement that



dashed ring: antagonist

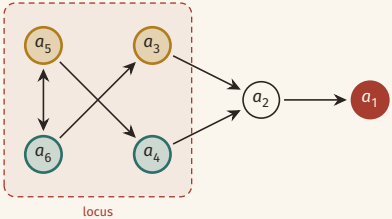
Delete	Shared explanation of P(Abort)
a_2	$\{a_1\}$
a_5	$\{a_1, a_4\}$
a_6	$\{a_1, a_3\}$

The locus

Locus. Let $\psi \in \mathrm{ImpDis}^d(\alpha + \beta)$, and let $\mathbb{E}(\psi)$ be the explanations of ψ that are contained in some preferred extension of $\alpha + \beta$. The locus of the disagreement is

$$\bigcup \mathbb{E}(\psi) \setminus \bigcap \mathbb{E}(\psi).$$

- The arguments on which the accepted explanations **differ**. No deletion is involved
- Here $\mathbb{E}(\mathrm{P(Abort)}) = \big\{\{a_1, a_3, a_5\}, \{a_1, a_4, a_6\}\big\}$, so the locus is $\{a_3, a_4, a_5, a_6\}$. The shared a_1 drops out
- In a finite framework the locus is never empty under distal implicit disagreement
- **Q2 answered**: the disagreement sits in the replies to a_2 and in the stances behind them

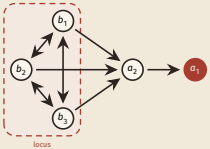


The locus is the symmetric difference of the two explanations.

Locus and weak spots are independent

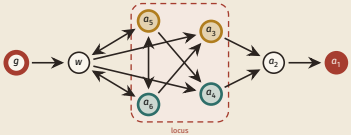
In the running example the locus is the weak spots a_3, a_4 plus the antagonists a_5, a_6. A coincidence:

In the locus, neither weak spot nor antagonist



- Three pairwise conflicting defenders of a_1 against a_2. Preferred extensions $\{a_1, b_i\}$; locus $\{b_1, b_2, b_3\}$
- Delete b_1: two disagreeing explanations remain. So b_1 is no weak spot, and no antagonist

A weak spot outside the locus



- A **wrecker** w attacks a_3 to a_6. An unattacked **gatekeeper** g keeps it out
- Delete g: $\{w, a_2\}$ becomes a preferred extension, so g is a weak spot
- Yet g is in no explanation: a_5 and a_6 already answer w. Both examples are mechanized

04

The embedding: frameworks, agents, inference

The LogiKEy layers: L0, the meta-logic

L3	**Applications** the running example and five small graphs: `ID_Examples`
L2	**Domain theory** agents, coalitions, inference, explanations, disagreement: `ID_Frameworks`, `ID_Inference`, `ID_Explanations`, `ID_Disagreement`
L1	**Object logic** abstract argumentation with Dung's semantics: session `AA_Fixed`
L0	**Meta-logic** classical higher-order logic: Isabelle/HOL, with Sledgehammer and Nitpick

- LogiKEy:[7] classical higher-order logic as a **universal meta-logic** for object logics, domain theories and applications
- Here: Isabelle/HOL, with Sledgehammer and Nitpick
- Sets and relations are predicates. Inclusion and equality of sets also come **relative to a universe**
- **Design decision**: every notion takes an explicit universe U, so that graphs, agents, coalitions and deletions are values of one type

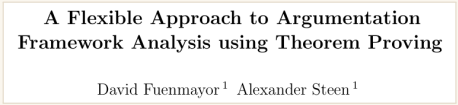
```
type_synonym 'a Set = ‹'a ⇒ bool›
type_synonym 'a Rel = ‹'a ⇒ 'a ⇒ bool›
A ⊆ B  ≡ ∀x. (A x ⟶ B x)
A ⊆ᵁ B ≡ ∀x. U x ⟶ (A x ⟶ B x)
A ≈ᵁ B ≡ ∀x. U x ⟶ (A x ⟷ B x)
```

[7] Benzmüller et al., *Designing normative theories for ethical and legal reasoning*, Artificial Intelligence 2020.

The LogiKEy layers: L1, the object logic

L3	**Applications**	the running example and five small graphs: `ID_Examples`
L2	**Domain theory**	agents, coalitions, inference, explanations, disagreement: `ID_Frameworks`, `ID_Inference`, `ID_Explanations`, `ID_Disagreement`
L1	**Object logic**	abstract argumentation with Dung's semantics: session `AA_Fixed`
L0	**Meta-logic**	classical higher-order logic: Isabelle/HOL, with Sledgehammer and Nitpick

- Dung's extension semantics as HOL predicates over a universe: the first lecture


A Flexible Approach to Argumentation Framework Analysis using Theorem Proving

David Fuenmayor [1] Alexander Steen [1]

- The session `AA_Fixed`: the library of Fuenmayor and Steen, audited and repaired (slide 35)
- Today's theories import it as it is. They use seven of its definitions and three of its results: next slide
- **Modular**: L2 uses only the attack graph and the conclusion map. A structured formalism could replace this layer without touching L2

What layer L1 provides

```
defends^𝒜 ≡ λatt S a. ∀b. 𝒜 b ⟶ att b a ⟶ (∃z. 𝒜 z ∧ S z ∧ att z b)
conflictfreeExt^𝒜 ≡ λatt S. ∀a b. (𝒜 a ∧ 𝒜 b) ⟶ (S a ∧ S b) ⟶ ¬(att a b)
admissibleExt^𝒜   ≡ λatt S. conflictfreeExt^𝒜 att S ∧ (∀a. 𝒜 a ⟶ S a ⟶ defends^𝒜 att S a)
completeExt^𝒜     ≡ λatt S. admissibleExt^𝒜 att S ∧ (∀a. 𝒜 a ⟶ defends^𝒜 att S a ⟶ S a)
preferredExt^𝒜    ≡ λatt S. maximal^𝒜 (completeExt^𝒜 att) S id
idealSet^𝒜        ≡ λatt S. admissibleExt^𝒜 att S ∧ (∀E. preferredExt^𝒜 att E ⟶ S ⊆^𝒜 E)

minimal Prop Obj E ≡ Prop Obj ∧ (∀X. Prop X ∧ E(X) ⊆ E(Obj) ⟶ E(X) ≈ E(Obj))
```

- These seven definitions of `AA_Fixed` are all the semantics that today's theories use. A framework is a universe $\mathcal{A}$ and an attack relation `att`, passed one after the other
- Three results of the library are used as well: maximal admissible sets exist (by Zorn's lemma), they are the preferred extensions, and subsets of a preferred extension are conflict-free
- **Notation from here on**: the universe is called `U`, and the predicates are written in prefix form, `preferredExt U att E` for $\texttt{preferredExt}^{\texttt{U}}\ \texttt{att}\ \texttt{E}$. The theories write `extensions.preferredExt`, with the name of the library theory; the code boxes drop this qualifier, and most type annotations

The LogiKEy layers: L2, the domain theory

L3	**Applications** the running example and five small graphs: `ID_Examples`
L2	**Domain theory** agents, coalitions, inference, explanations, disagreement: `ID_Frameworks`, `ID_Inference`, `ID_Explanations`, `ID_Disagreement`
L1	**Object logic** abstract argumentation with Dung's semantics: session `AA_Fixed`
L0	**Meta-logic** classical higher-order logic: Isabelle/HOL, with Sledgehammer and Nitpick

- The theory of parts 2 and 3, in four theories. Each imports the one before it, and the first imports L1
- **Definitions and abbreviations only**, thirty of them, and no axioms: every notion unfolds to the predicates of L1, and from there to plain HOL
- The rest of this part covers the first two theories, part 5 the other two

```
session Implicit_Disagreement = AA_Fixed +
  theories
    ID_Frameworks     (* agents, coalitions, deletion *)
    ID_Inference      (* three inference relations *)
    ID_Explanations   (* explanations *)
    ID_Disagreement   (* disagreement, locus *)
    ID_Examples       (* the finite graphs: L3 *)
```

The ROOT entry, without its options; the comments are added here.

One type for graphs, agents, coalitions and deletions

- **The design decision**: the ambient graph, an agent, a coalition and a framework after a deletion are all predicates of type `'a Set`, over **one** type of arguments
- They can be compared and combined in one theory. The attacks are always **induced** by the ambient relation

```
definition induced_att :: "'a Rel ⇒ 'a Set ⇒ 'a Rel" where
  "induced_att att A = (λa b. A a ∧ A b ∧ att a b)"
definition is_agent :: "'a Set ⇒ 'a Rel ⇒ 'a Set ⇒ 'a Rel ⇒ bool" where
  "is_agent U att A r ⟷ A ⊆ U ∧ r = induced_att att A"

abbreviation "coalition A B ≡ A ∪ B"
abbreviation "coalition_att att A B ≡ induced_att att (coalition A B)"
abbreviation "remove_argument U a ≡ U ∩ (λx. x ≠ a)"
abbreviation "remove_att U att a ≡ induced_att att (remove_argument U a)"
```

Induced or ambient attacks?

```
lemma preferred_induced [simp]:
  "preferredExt U (induced_att att U) E = preferredExt U att E"
  unfolding extensions.preferredExt_def maximal_rel_def by simp

lemma coalition_is_agent:
  "is_agent U att A r ⟹ is_agent U att B s ⟹ is_agent U att (coalition A B) (coalition_att att A B)"
lemma coalition_commutes:   "coalition A B = coalition B A"
lemma coalition_associates: "coalition (coalition A B) C = coalition A (coalition B C)"
lemma deletion_is_agent:    "is_agent U att (remove_argument U a) (remove_att U att a)"
lemma deletion_removes_incident_edges: "¬ remove_att U att a a b ∧ ¬ remove_att U att a b a"
```

- An agent can be evaluated with its own induced relation, or with the ambient relation restricted to its universe. The library guards every quantifier with the universe, so the result is **the same**
- `preferred_induced` has four siblings, for defence and for conflict-free, admissible and complete sets. They are simp rules: the induced relation disappears from every goal
- **Consequence**: all later notions are stated with the ambient `att`. An agent is just its universe
- Coalitions and deletions are agents again; forming coalitions is commutative and associative

Conclusions without a logic

```
definition proximal_explanation :: "'a Set ⇒ ('a ⇒ 'c) ⇒ 'c ⇒ 'a ⇒ bool" where
  "proximal_explanation U Con p a ⟷ U a ∧ Con a = p"
```

The conclusion map is a parameter

- `Con :: 'a ⇒ 'c` assigns conclusions of an **arbitrary type** `'c` to the arguments
- Only **equality** of conclusions is used. There is no consequence relation: an argument for a conjunction is not an argument for its conjuncts
- Several arguments may have the same conclusion, so a proximal explanation is a relation, not a function

The guard `U a` is essential

- An extension is a predicate on the whole type, and the library says nothing about its values **outside** the universe
- Without the guard, an argument that the agent does not have could witness the agent's conclusion
- This is the price of an explicit universe, as in the first lecture: sets are compared with $\approx^{U}$, members are guarded

In the case study the conclusion map is `λa. a = A1`, of type `bio_argument ⇒ bool`: a Boolean label for the issue P(Abort).

The inference relations in HOL

```
definition credulous_conclusion :: "'a Set ⇒ 'a Rel ⇒ ('a ⇒ 'c) ⇒ 'c ⇒ bool" where
  "credulous_conclusion U att Con p ⟷
    (∃E. preferredExt U att E ∧ (∃a. E a ∧ proximal_explanation U Con p a))"

definition shared_conclusion where
  "shared_conclusion U att Con p ⟷
    (∀E. preferredExt U att E ⟶ (∃a. E a ∧ proximal_explanation U Con p a))"

definition shared_argument where
  "shared_argument U att Con p ⟷
    (∃a. proximal_explanation U Con p a ∧ (∀E. preferredExt U att E ⟶ E a))"

abbreviation agrees_that where "agrees_that ≡ shared_conclusion"
abbreviation agrees_because where "agrees_because ≡ shared_argument"
```

- **The quantifier order carries the whole distinction**: ∀E. … ∃a allows a different witness in each extension, ∃a. … ∀E demands one witness for all of them
- ∀E is a HOL quantifier over predicates. The definitions match the paper literally, with no enumeration of the extensions and no finiteness assumption
- “Agrees that” and “agrees because” add names, not semantics: they are abbreviations

The agreement hierarchy, proved

```
lemma shared_argument_imp_shared_conclusion:
  "shared_argument U att Con p ⟹ shared_conclusion U att Con p"
  unfolding shared_argument_def shared_conclusion_def by blast

lemma preferred_extensions_exist: "∃E. preferredExt U att E"
  using preferredExist maxAdmissibleComplete unfolding extensions.preferredExt_def by blast

lemma shared_conclusion_imp_credulous:
  "shared_conclusion U att Con p ⟹ credulous_conclusion U att Con p"
  using preferred_extensions_exist[of U att]
  unfolding shared_conclusion_def credulous_conclusion_def by blast

lemma empty_framework_no_conclusion:
  "¬ credulous_conclusion (λ_. False) att Con p ∧ ¬ shared_conclusion (λ_. False) att Con p ∧
   ¬ shared_argument (λ_. False) att Con p"
```

- $\vdash_a \Longrightarrow \vdash_c$ is pure quantifier logic: unfold the two definitions, call `blast`
- $\vdash_c \Longrightarrow \vdash_\cup$ needs that a preferred extension **exists**. Otherwise "every preferred extension has an argument for p" would hold vacuously
- Existence comes from the library, for every universe, finite or not: Zorn's lemma. The empty framework has the preferred extension $\emptyset$, and derives nothing

Reusing a layer is only as safe as the layer

What an audit of the original library found

- Two Zorn-style bridge lemmas, behind the existence of preferred extensions, were admitted without proof. As stated they are **false**: they mix inclusion relative to the universe with global inclusion
- Skeptical acceptance ranged only over labellings that accept something: a self-attacking argument counted as skeptically accepted

What the session `AA_Fixed` does

- It proves Zorn's lemma relative to a universe (`ZornLemma_rel`, `ZornLemma2_rel`), so that the existence of preferred extensions rests on checked proofs
- It repairs the definition of skeptical acceptance
- It keeps the counterexamples as **regression theorems**

```
lemma audit_bridge2_counterexample:   (* finite prefixes of the naturals, over an empty universe *)
  "allChainsHaveUB_rel (λ(_::nat). False) audit_prefixes ∧ ¬ allChainsHaveUB audit_prefixes"
lemma self_attack_not_accepted:
  "¬ sJ (labellings.completeLab (λ_. True)) (λ(_::unit) _. True) () ∧
   ¬ cJ (labellings.completeLab (λ_. True)) (λ(_::unit) _. True) ()"
```

Source: `AA_Fixed/regression-checks.thy`. The first lecture showed the library after these repairs.

05

The embedding: explanations and disagreement

Explanations in HOL

```
definition explanation_support :: "'a Set ⇒ 'a Rel ⇒ ('a ⇒ 'c) ⇒ 'c ⇒ 'a Set ⇒ bool" where
  "explanation_support U att Con p E ⟷
    E ⊆ U ∧ (∃a. E a ∧ Con a = p) ∧                                    (* (i) *)
    conflictfreeExt U att E ∧ (∀a. E a ⟶ defends_rel U att E a)"        (* (ii) *)

definition explanation where
  "explanation U att Con p E ⟷ minimal (explanation_support U att Con p) E id"   (* (iii) *)

definition shared_explanation where
  "shared_explanation U att Con p ⟷
    (∃E. explanation U att Con p E ∧ (∀P. preferredExt U att P ⟶ E ⊆ P))"
abbreviation strongly_agrees_because where "strongly_agrees_because ≡ shared_explanation"

lemma explanation_admissible:
  assumes expl: "explanation U att Con p E" shows "admissibleExt U att E"
theorem agreement_hierarchy:
  assumes strong: "shared_explanation U att Con p"
  shows "shared_argument U att Con p ∧ shared_conclusion U att Con p ∧ credulous_conclusion U att Con p"
```

- The three clauses of the definition, in order. Clause (iii) is the generic `minimal` of the library, with the property `explanation_support` … and the projection `id`
- Defence is evaluated in the **whole** framework, not inside E: every explanation is admissible

Why the conflict-freeness clause

one argument

that attacks itself

- A self-attacking argument defends itself: it attacks its only attacker
- With “defends all its arguments” alone, the singleton would be a minimal self-defending set, hence an **explanation of its own conclusion**
- The first version of the definition had no conflict-freeness clause. Three lemmas say **exactly what the clause changes**

```
lemma self_attacking_explanation_without_cf:
  "explanation_nocf (λ_. True) (λ(_::unit) _. True) (λ_. ()) () (λ_. True)"
lemma self_attacking_no_explanation:
  "¬ explanation (λ_. True) (λ(_::unit) _. True) (λ_. ()) () (λ_. True)"

lemma explanation_cf_invariant:
  assumes cf: "conflictfreeExt U att C" and sub: "E ⊆ C"
  shows "explanation U att Con p E ⟷ explanation_nocf U att Con p E"
```

- The framework of the first two lemmas: the type `unit` has one inhabitant `()`, and the relation `λ_ _. True` makes it attack itself
- Inside a conflict-free set, in particular inside a preferred extension, the clause is invisible. Shared explanations and the locus only look there, so they do not depend on it

Shared explanations are ideal semantics

- An **ideal set** is an admissible set that is contained in every preferred extension. The **ideal extension** is the greatest ideal set[8]
- The inference relation $\vdash_{\mathcal{E}}$ was defined through explanations. It turns out to be a known one: **skeptical acceptance under ideal semantics**
- The statement is about ideal sets. It gives acceptance in the ideal extension, because the ideal sets have a greatest element

In the running example

The only subsets of both preferred extensions are $\emptyset$ and $\{a_1\}$, and $\{a_1\}$ is not admissible. The only ideal set is $\emptyset$: no ideal set argues for P(Abort), and no explanation is shared.

```
theorem shared_explanation_iff_idealset:
  assumes fin: "finite {a. U a}"
  shows "shared_explanation U att Con p
         ⟷ (∃E. idealSet U att E ∧ (∃a. U a ∧ E a ∧ Con a = p))"
```

- What the paper contributes is therefore a **characterization**: agreement under ideal semantics is agreement on an explanation

[8] Dung et al., *Computing ideal sceptical argumentation*, Artificial Intelligence 2007.

How the proof goes

From a shared explanation to an ideal set

- A shared explanation $\mathcal{E}$ defends its members and contains an argument for p
- It lies in a preferred extension, which exists, so it is conflict-free
- Hence $\mathcal{E}$ is an ideal set. **No finiteness** is needed

From an ideal set to a shared explanation

- An ideal set I with an argument for p, cut down to the universe, satisfies clauses (i) and (ii)
- A minimal subset is extracted by **induction on the cardinality**: an explanation inside I

```
lemma minimal_support_extract:
  assumes fin: "finite {a. U a}" and SU: "S ⊆ U" and supp: "explanation_support U att Con p S"
  shows "∃E. explanation U att Con p E ∧ E ⊆ S"

lemma shared_explanation_iff_idealset_nitpick: "finite {a. U a} ⟶ (shared_explanation U att Con p ⟷ …)"
  nitpick[card 'a = 1-4, card 'c = 1-3, timeout = 120, expect = none]  oops
```

- Finiteness enters only through `minimal_support_extract`
- The Nitpick query is a bounded regression check: up to four arguments and three conclusions it must find **no** countermodel, or the build fails

Disagreement in HOL

```
definition "disagrees_that U att Con p ⟷ credulous_conclusion U att Con p ∧ ¬ shared_conclusion U att Con p"
definition "proximal_disagreement U att Con p ⟷ shared_conclusion U att Con p ∧ ¬ shared_argument U att Con p"
definition "distal_disagreement U att Con p ⟷ shared_conclusion U att Con p ∧ ¬ shared_explanation U att Con p"
```

The matrix of part 3, with HOL in the cells. S, A, H stand for `shared_conclusion`, `shared_argument`, `shared_explanation`; the arguments `att Con p` are left out.

		Agreeing that	
		Potential: `S A ∧ S B`	**Actual:** `S (A ∪ B)`
Disagreeing because	**Distal:** `¬ H (A ∪ B)`	`potential_distal_disagreement` `S A ∧ S B ∧ ¬ H (A ∪ B)`	`distal_implicit_disagreement` `distal_disagreement (coalition A B)`
	Proximal: `¬ A (A ∪ B)`	`potential_proximal_disagreement` `S A ∧ S B ∧ ¬ A (A ∪ B)`	`proximal_implicit_disagreement` `proximal_disagreement (coalition A B)`

- **Disagreement adds no semantics**: each notion is a Boolean combination of the inference predicates, evaluated on one universe or another
- The actual notions are abbreviations; the potential ones take the two agents as separate parameters

Weak spots, antagonists and the locus in HOL

```
definition weak_spot where
  "weak_spot U att Con p a ⟷
    shared_conclusion U att Con p ∧
    disagrees_that (remove_argument U a) (remove_att U att a) Con p"

definition antagonist where
  "antagonist U att Con p a ⟷
    U a ∧ distal_disagreement U att Con p ∧
    shared_explanation (remove_argument U a) (remove_att U att a) Con p"

definition accepted_explanation where
  "accepted_explanation U att Con p E ⟷
    explanation U att Con p E ∧ (∃P. preferredExt U att P ∧ E ⊆ P)"

definition locus where
  "locus U att Con p x ⟷
    (∃E. accepted_explanation U att Con p E ∧ E x) ∧
    ¬ (∀E. accepted_explanation U att Con p E ⟶ E x)"
```

- The paper's definitions, transcribed. For a coalition, pass `coalition A B` as the universe
- The locus is "in some accepted explanation, not in all": union minus intersection, as two quantifiers

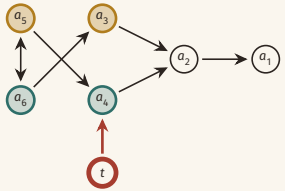
General facts, proved

```
lemma proximal_imp_distal: "proximal_disagreement U att Con p ⟹ distal_disagreement U att Con p"
lemma ordinary_and_distal_disjoint: "¬ (disagrees_that U att Con p ∧ distal_disagreement U att Con p)"

lemma weak_spot_is_present: assumes "weak_spot U att Con p a" shows "U a"
lemma removing_the_only_concluder_is_not_a_weak_spot:
  assumes "∀b. U b ⟶ Con b = p ⟶ b = a" shows "¬ weak_spot U att Con p a"
lemma antagonists_are_not_weak_spots:
  assumes ant: "antagonist U att Con p a" shows "¬ weak_spot U att Con p a"

lemma preferred_contains_explanation:
  assumes fin: "finite {a. U a}" and sc: "shared_conclusion U att Con p" and P: "preferredExt U att P"
  shows "∃E. explanation U att Con p E ∧ E ⊆ P"
lemma distal_disagreement_locus_nonempty:
  assumes fin: "finite {a. U a}" and dd: "distal_disagreement U att Con p"
  shows "∃x. locus U att Con p x"
```

- The first two are propositional consequences of the hierarchy: unfold, `blast`
- A weak spot is an argument of the framework, and the only argument for p is never one. The proofs of the second and third weak-spot facts were found by **Sledgehammer** (`smt`, `metis`)
- The last two are short **Isar** proofs. They reuse the extraction of a minimal support, which is why they assume a finite universe

Deletion versus a fresh attacker

- In a debate nobody deletes an argument. One **adds** an objection, as with viability against a_4
- Sakama's counterfactual construction: add a fresh, unattacked attacker t of a
- **The two are equivalent**: t is in every complete extension, a in none, and the rest behaves exactly as after deleting a



a fresh attacker t of a_4

```
definition fresh_attacker_att :: "'a Rel ⇒ 'a Set ⇒ 'a ⇒ 'a ⇒ 'a Rel" where
  "fresh_attacker_att att U a t = (λx y. induced_att att U x y ∨ (x = t ∧ y = a))"

lemma fresh_attacker_complete:
  assumes t: "¬ U t" and a: "U a"
  shows "completeExt (U ∪ {|t|}) (fresh_attacker_att att U a t) E ⟷
         E t ∧ ¬ E a ∧ completeExt (remove_argument U a) att E"
```

- An Isar proof of about a hundred lines. **Not yet checked**: the corollary for preferred extensions, whose proof is a draft inside a comment

06

The case study, lemma by lemma

The LogiKEy layers: L3, the applications

L3 **Applications**
the running example and five small graphs: `ID_Examples`

L2 **Domain theory**
agents, coalitions, inference, explanations, disagreement: `ID_Frameworks`, `ID_Inference`, `ID_Explanations`, `ID_Disagreement`

L1 **Object logic**
abstract argumentation with Dung's semantics: session `AA_Fixed`

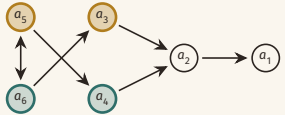
L0 **Meta-logic**
classical higher-order logic: Isabelle/HOL, with Sledgehammer and Nitpick

- Finite graphs as instances of L2: a type of arguments, an attack relation, a conclusion map
- Proofs by exhaustion: the quantifiers over arguments unfold into cases, then `simp` and `auto` decide. `metis` and Sledgehammer do the rest
- **Every question becomes a lemma** of `ID_Examples`:

```
Q1  bio_preferred   bio_shared_argument
    bio_no_shared_explanation
Q2  bio_accepted_explanations   bio_locus
Q3  bio_A3_weak_spot   bio_A4_weak_spot
    bio_A1_not_weak_spot   bio_A2_antagonist
    bio_A5_antagonist   bio_A6_antagonist
Q4  bio_potential_and_actual_distal
    bio_distal_but_not_proximal
```

Encoding the running example

- A **datatype** with six constructors, an attack relation that lists the seven edges, a Boolean label for P(Abort)
- Julia and Emma are two predicates. Their coalition is the whole type: its universe is $\lambda\_.\ \texttt{True}$




```
datatype (plugins only: code) bio_argument = A1 | A2 | A3 | A4 | A5 | A6
definition bio_att :: "bio_argument Rel" where
  "bio_att a b ⟷ (a = A2 ∧ b = A1) ∨ (a = A3 ∧ b = A2) ∨ (a = A4 ∧ b = A2) ∨ (a = A5 ∧ b = A4) ∨
                 (a = A6 ∧ b = A3) ∨ (a = A5 ∧ b = A6) ∨ (a = A6 ∧ b = A5)"
abbreviation bio_con where "bio_con a ≡ a = A1"
abbreviation bio_left where "bio_left ≡ {|A1,A3,A5|}"
abbreviation bio_right where "bio_right ≡ {|A1,A4,A6|}"
abbreviation bio_left_agent where "bio_left_agent ≡ bio_left ∪ {|A2|}"          (* Julia *)
abbreviation bio_right_agent where "bio_right_agent ≡ bio_right ∪ {|A2|}"       (* Emma *)

lemma bio_all: "(∀x::bio_argument. P x) ⟷ P A1 ∧ P A2 ∧ P A3 ∧ P A4 ∧ P A5 ∧ P A6"
  by (metis bio_argument.exhaust)
lemma bio_ex:  "(∃x::bio_argument. P x) ⟷ P A1 ∨ P A2 ∨ P A3 ∨ P A4 ∨ P A5 ∨ P A6"
lemma bio_coalition_all: "coalition bio_left_agent bio_right_agent = (λ_. True)"
```

`bio_all`, `bio_ex`: quantifiers become six cases, as with `all_Arg` and `ex_Arg` in the first lecture.

Computing the preferred extensions by proof

```
lemma bio_complete:
  "completeExt (λ_. True) bio_att E ⟷ (E ≈ (λ_. False)) ∨ E ≈ bio_left ∨ E ≈ bio_right"
  unfolding extensions.completeExt_def extensions.admissibleExt_def
    extensions.conflictfreeExt_def defends_rel_def bio_att_def
  by (auto simp: bio_all bio_ex; blast)

lemma bio_left_preferred: "preferredExt (λ_. True) bio_att bio_left"
  unfolding extensions.preferredExt_def maximal_rel_def bio_complete id_def
  by (auto simp: bio_all)

lemma bio_preferred:
  "preferredExt (λ_. True) bio_att E ⟷ E ≈ bio_left ∨ E ≈ bio_right"
```

- `bio_complete` speaks about **every** set E: of the 64 sets of arguments, exactly three are complete. Unfold the definitions down to plain HOL, split into cases, let the automation finish
- The characterization is then used as a **rewrite rule**: in the proof of `bio_left_preferred` the quantifier over complete extensions is replaced by three cases
- **Proved, not searched**. In the first lecture Nitpick enumerated the extensions. Here every later lemma needs these facts as premises, so they must be theorems

Q1 and Q4 as lemmas

```
lemma bio_shared_argument: "shared_argument (λ_. True) bio_att bio_con True"
  unfolding shared_argument_def proximal_explanation_def
  by (auto simp: bio_preferred bio_all bio_ex)
lemma bio_no_shared_explanation: "¬ shared_explanation (λ_. True) bio_att bio_con True"

lemma bio_distal_but_not_proximal:
  "distal_disagreement (λ_. True) bio_att bio_con True ∧
    ¬ proximal_disagreement (λ_. True) bio_att bio_con True"

lemma bio_agents_share_conclusion:
  "shared_conclusion bio_left_agent bio_att bio_con True"
  "shared_conclusion bio_right_agent bio_att bio_con True"
lemma bio_potential_and_actual_distal:
  "potential_distal_disagreement bio_left_agent bio_right_agent bio_att bio_con True ∧
    distal_implicit_disagreement bio_left_agent bio_right_agent bio_att bio_con True"
```

- The conclusion `True` of `bio_con` is the issue P(Abort)
- **Q1**: the argument is shared, the explanation is not. Proof of the second lemma, in Isar: a shared explanation would lie in both preferred extensions, so it could contain neither A3 nor A4; but it contains A1, which one of them must defend against A2
- **Q4**: the distal implicit disagreement is both potential and actual

Q2: the locus as a lemma

```
lemma bio_left_support_forces:
  assumes "explanation_support (λ_. True) bio_att bio_con True X" and "X ⊆ bio_left"
  shows "X = bio_left"
  using assms unfolding explanation_support_def defends_rel_def bio_att_def
  by (auto simp: fun_eq_iff bio_all bio_ex)

lemma bio_accepted_explanations:
  "accepted_explanation (λ_. True) bio_att bio_con True E ⟷ E = bio_left ∨ E = bio_right"

lemma bio_locus:
  "locus (λ_. True) bio_att bio_con True x ⟷ x = A3 ∨ x = A4 ∨ x = A5 ∨ x = A6"
  unfolding locus_def bio_accepted_explanations by (cases x) auto

lemma bio_coalition_locus:
  "locus (coalition bio_left_agent bio_right_agent) bio_att bio_con True =
    (λx. x = A3 ∨ x = A4 ∨ x = A5 ∨ x = A6)"
```

- Inside `bio_left`, a set that satisfies clauses (i) and (ii) needs A1, then A3, then A5: it is all of `bio_left`. So the two preferred extensions are the only accepted explanations
- With that characterization the locus is computed by a case distinction on the argument
- **Q2**: the locus of Julia + Emma on P(Abort) is $\{a_3, a_4, a_5, a_6\}$, as an equation between predicates

Q3: weak spots and antagonists as lemmas

```
definition bio_minus :: "bio_argument ⇒ bio_argument Set" where
  "bio_minus a = remove_argument (λ_. True) a"
lemma bio_minus_mem [simp]: "bio_minus a x ⟷ x ≠ a"

lemma bio_minus3_complete:
  "completeExt (bio_minus A3) bio_att E ⟷
     (¬ E A1 ∧ ¬ E A2 ∧ ¬ E A4 ∧ ¬ E A5 ∧ ¬ E A6) ∨
     (¬ E A1 ∧ E A2 ∧ ¬ E A4 ∧ E A5 ∧ ¬ E A6) ∨
     (E A1 ∧ ¬ E A2 ∧ E A4 ∧ ¬ E A5 ∧ E A6)"
lemma bio_minus3_left_preferred:  "preferredExt (bio_minus A3) bio_att {|A2,A5|}"
lemma bio_minus3_right_preferred: "preferredExt (bio_minus A3) bio_att {|A1,A4,A6|}"

lemma bio_A3_weak_spot: "weak_spot (λ_. True) bio_att bio_con True A3"          (* and bio_A4_weak_spot *)
lemma bio_A1_not_weak_spot: "¬ weak_spot (λ_. True) bio_att bio_con True A1"
  by (rule removing_the_only_concluder_is_not_a_weak_spot) simp
lemma bio_A2_antagonist: "antagonist (λ_. True) bio_att bio_con True A2"        (* also for A5 and A6 *)
```

- Without A3: the same datatype, a smaller universe. The complete extensions are described on the five remaining arguments; about A3, outside the universe, nothing is said
- `bio_minus` is a definition, not an abbreviation: an opaque constant keeps the rewriting first-order
- **Q3**: weak spots a_3, a_4; not a_1, by the general lemma; antagonists a_2, a_5, a_6

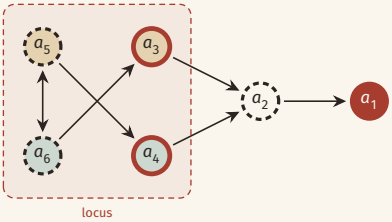
The other graphs of `ID_Examples`

Graph	What it shows	Theorem
Two mutually attacking arguments, one conclusion	$\vdash_c$ without $\vdash_a$: proximal implicit disagreement, no disagreement that	`proximal_disagreement_is_not_ordinary_disagreement`
One self-attacking argument	what the conflict-freeness clause excludes	`self_attacking_no_explanation`
One attack, two one-argument agents	an attack that only the coalition sees; the attacked conclusion is lost	`coalition_reveals_cross_edge`, `one_way_coalition_keeps_attacker`
The coalition-only example (six arguments)	distal implicit disagreement, actual but not potential	`co_distal_disagreement_only_in_coalition`
Three conflicting defenders (five arguments)	a locus element that is neither weak spot nor antagonist	`tri_locus_element_neither`
Gatekeeper and wrecker (eight arguments)	a weak spot outside the locus	`gate_weak_spot_outside_locus`

- The smallest graphs need no datatype: the two arguments are the two values of `bool`, the single argument is the value of `unit`
- Each definition is tested where it should hold and where it should fail

Back to Julia and Emma: the four questions answered

- **Q1** *On what do they agree?*
  They agree that, and they agree because: a_1 is shared. They do not strongly agree because: the explanations $\{a_1, a_3, a_5\}$ and $\{a_1, a_4, a_6\}$ differ
- **Q2** *Where is the disagreement?*
  In the locus $\{a_3, a_4, a_5, a_6\}$: the replies to a_2 and the stances behind them
- **Q3** *What breaks the coalition, what would make it robust?*
  Weak spots a_3, a_4: lose one and the coalition fractures, as at the committee hearing. Antagonists a_2, a_5, a_6: lose one and a shared explanation appears
- **Q4** *Actual, or only potential?*
  A distal implicit disagreement, potential and actual. Not a proximal one

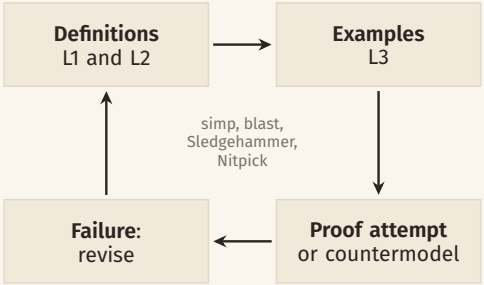


Every answer is a theorem of `ID_Examples`, checked by Isabelle/HOL.

07

Lessons and outlook

The experimental loop



What the loop caught in this development

- **Three faulty examples** in earlier drafts: attacks outside an agent's set, a coalition example that did not support its claim, one attack written as a mutual one. Each surfaced as a proof that would not close
- **A definition**: explanations without conflict-freeness
- **A theorem**: shared explanations are ideal semantics
- **The layer below**: two false lemmas in the library

- LogiKEy is **experimental by design**: definitions, examples and lemmas are developed together, and the prover tells you early which of them is wrong

Lessons for LogiKEy development

Modelling

- Transcribe the definitions **literally**. Let abbreviations carry the terminology of the paper
- Keep the framework as explicit parameters: one definition then serves the graph, the agents, the coalitions and the deletions
- With an explicit universe, **guard every witness**
- Test each definition on the smallest types first: `unit`, `bool`

Proving

- Exhaustion lemmas turn quantifiers over a finite datatype into cases
- Characterize the complete extensions **once**, then rewrite with that lemma
- Name derived universes with opaque definitions
- On larger graphs, write the attack relation by cases on the attacker, and prove small facts with `blast` before calling `auto`
- Sledgehammer for the general lemmas; Nitpick with `expect = none` as a regression check

In numbers: five theories, about 2 230 lines, 55 definitions and abbreviations, over 160 lemmas and theorems, about 50 of them with structured Isar proofs. On top of `AA_Fixed` the session is checked in under a minute.

Outlook

- **Structured argumentation at L1**: with a formalism such as ASPIC+[9] the arguments have premises and rules. The L2 theories apply unchanged to the generated graph, and the analysis could then locate the premises on which the parties differ, not only the arguments
- **Roles of arguments in explanations**: who defends, who is defended. Useful for strategic and dynamic settings, where a rival coalition chooses what to attack
- **Support and implicit agreement**: bipolar frameworks, and the dual phenomenon of parties who disagree on a conclusion while sharing reasons
- **More than two agents**: coalitions of coalitions already exist, since the operation is associative. The potential notions need a definition for n agents
- **Other semantics**: the inference relations are stated for preferred extensions. Making the semantics a parameter is a small change in HOL
- **Infinite frameworks**: the definitions need no finiteness, two of the theorems do

[9] Prakken, *An abstract framework for argumentation with structured arguments*, Argument & Computation 2010.

Summary

1. **Implicit disagreement** is agreement that together with disagreement because: the same conclusion, for different reasons
2. **Three levels of agreement**, shared conclusion, shared argument, shared explanation, which differ in what stays fixed across the preferred extensions
3. **Agents, coalitions and deletions are universes**: predicates over one type of arguments, with the induced attacks. This is what the explicit universe of the library is for
4. **Disagreement adds no semantics**: proximal and distal, potential and actual are Boolean combinations of the inference predicates
5. **Locating**: weak spots and antagonists by counterfactual deletion, the locus by comparing explanations. The three notions are independent
6. **The prover did substantive work**: a correspondence with ideal semantics, a repaired definition, repaired examples, an audited library

Sources: the Isabelle session `Implicit_Disagreement` (theories `ID_Frameworks`, `ID_Inference`, `ID_Explanations`, `ID_Disagreement`, `ID_Examples`) and its parent session `AA_Fixed`.

Backup: the theories at a glance

Theory	Lines	Contents	Part
`ID_Frameworks`	349	induced attacks, agents, coalitions, deletion; invariance under induced attacks; existence of preferred extensions; the fresh attacker	4
`ID_Inference`	68	proximal explanations; the three inference relations; two steps of the hierarchy	4
`ID_Explanations`	288	explanations, with and without conflict-freeness; shared explanations; the agreement hierarchy; ideal semantics	5
`ID_Disagreement`	195	the disagreement predicates; weak spots, antagonists, locus; general facts	5
`ID_Examples`	1335	small tests; the running example; the coalition-only example; three defenders; the gatekeeper	6

- More than half of the development is the examples: computing with a proof assistant is done by proving
- The general theory, the first four theories, is 900 lines

Backup: building and reading the session

```
# build the session and its document (Linux, macOS)
./build.sh
# on Windows, in the shell that comes with Isabelle
isabelle env bash build.sh
# read the theories interactively
isabelle jedit -d . -d AA_Fixed -l AA_Fixed ID_Examples.thy
```

- Developed and checked with **Isabelle2025-2**
- The first build also builds the parent session `AA_Fixed`, which takes most of the time: about nine minutes on a laptop, then under half a minute for the five theories
- The PDF document of the session needs a LaTeX installation. Without one the script still checks every proof
- On Windows, Nitpick has only the Java solver SAT4J. The two model-generation queries for ideal extensions in `AA_Fixed` take about a minute each and carry an explicit `timeout=300`
- Reading order: `ID_Frameworks`, `ID_Inference`, `ID_Explanations`, `ID_Disagreement`, then the sections of `ID_Examples` in the order of slide 50