Exploring Intervention-Solution Spaces (ISS): Change and Choice in Abstract Argumentation

Shawn Bowers[1], Bertram Ludäscher[2,*]

[1]Department of Computer Science, Gonzaga University, USA
[2]School of Information Sciences, University of Illinois, Urbana-Champaign, USA

Abstract

An argumentation framework (AF) can admit multiple extensions (solutions) and may itself be modified, e.g., by removing or suspending attacks. We propose to study these two kinds of variation, choice among solutions and change by suspension, as dimensions of a single finite lattice called the *intervention-solution space* (ISS). ISS provides a unifying framework within which several existing lines of work, including incomplete AFs, enforcement, stability analysis, grounding extensions, and counterfactual reasoning, can be understood as queries over a common structure. We define ISS concepts that capture how suspensions and solutions interact, determine their minimal witnesses, classify attacks by type and analyze their effect on solutions, and formalize a notion of behavioral equivalence and sound quotient that determines when two ISS states are interchangeable under further suspensions. Two algorithms and an open-source prototype support exploration of the ISS.

Keywords

abstract argumentation, argumentation dynamics, incomplete argumentation frameworks, enforcement, explanation

1. Introduction

Consider an argumentation framework (AF) with attack edges A→B→C←→D (Figure 1a). The grounded semantics accepts A, defeats B, and leaves C and D undecided (Figure 1b). The preferred extensions $S_1 = \{A, C\}$ and $S_2 = \{A, D\}$ (Figure 1c,d) resolve undecided regions in two ways. We call this a *choice* in the *X-dimension* (for e*X*tension): one AF with several defensible solutions "side by side". Whether an argument belongs to some or all extensions is well studied [1, 2].

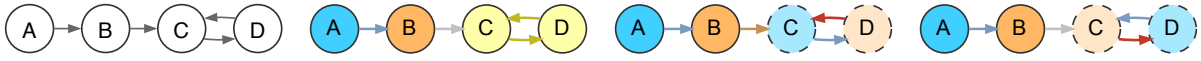
Our prior work [3] explains a chosen solution S_i through a minimal set of *critical attacks* whose *suspension* makes S_i the grounded extension of the restricted framework [4]: e.g., suspending D→C yields the grounded solution $\{A, C\}$, while suspending C→D yields $\{A, D\}$. Each such suspension modifies the framework to make a chosen solution grounded. This is *change* in the *Y-dimension*: one climbs up an axis while suspensions accumulate. Recent work on *grounding extensions* [5] generalizes the analysis of critical attacks to other semantics and analyzes its complexity. The two suspensions in Figure 1c,d do not tell the whole story: the AF admits $2^4 = 16$ possible suspension states (Table 1), and the full set forms a *suspension lattice* (Figure 2).

In this paper, we incorporate both the solution *choice* and attack *suspension* (an intervention) as dimensions into a single structure: the *intervention-solution space* (ISS). In ISS, each state of the lattice carries a set of solutions (X-dimension). The lattice determines which states are one suspension apart (Y-dimension), how many suspensions separate two states, and what happens along paths (sequences of suspensions).

Several existing lines of work can be expressed as queries over the two dimensions of an ISS.[1] For example, in incomplete AFs [8], some attacks are uncertain, and each way of resolving them yields a different AF (a Y-state) with its own extensions (X-dimension). Stability [9, 7] asks whether an

SAFA'26: Sixth International Workshop on Systems and Algorithms for Formal Argumentation, September, 2026, Barcelona, Spain
*Corresponding author.
bowers@gonzaga.edu (S. Bowers); ludaesch@illinois.edu (B. Ludäscher)

[1]The idea of exploring framework variants has a long history, e.g., HYPO [6] generated hypothetical variations; but this "seems to have attracted less attention" than other ideas [7].



(a) input $F = (A, R)$ (b) grounded labeling (c) preferred extension $S_1 = \{A, C\}$ (d) preferred extension $S_2 = \{A, D\}$

Figure 1: AF example: (a) input AF; (b) 3-valued grounded labeling: IN blue, OUT orange, UNDEC yellow; (c), (d): the preferred extensions, each resolving the undecided region $\{C, D\}$ by suspending either D→C or C→D. Dashed borders (and lighter fill) mark arguments whose status is resolved by solution *choice*. Suspended attacks are shown as red arrows and remaining arrow colors in (b)–(d) denote attack types (Sec. 4.2).

#	State D	$F - D$	$\mathcal{S}_D = \mathrm{pr}(F-D)$	$\mathrm{gr}(F-D)$
0	{}	A→B→C←→D	{A, C}, {A, D}	I O U U
1	{A→B}	A B→C←→D	{A, B, D}	I I O I
2	{B→C}	A→B C←→D	{A, C}, {A, D}	I O U U
3	{C→D}	A→B→C← D	{A, D}	I O O I
4	{D→C}	A→B→C →D	{A, C}	I O I O
5	{A→B, B→C}	A B C←→D	{A, B, C}, {A, B, D}	I I U U
6	{A→B, C→D}	A B→C← D	{A, B, D}	I I O I
7	{A→B, D→C}	A B→C →D	{A, B, D}	I I O I
8	{B→C, C→D}	A→B C← D	{A, D}	I O O I
9	{B→C, D→C}	A→B C →D	{A, C}	I O I O
10	{D→C, C→D}	A→B→C D	{A, C, D}	I O I I
11	{A→B, B→C, C→D}	A B C← D	{A, B, D}	I I O I
12	{A→B, D→C, C→D}	A B→C D	{A, B, D}	I I O I
13	{A→B, B→C, D→C}	A B C →D	{A, B, C}	I I I O
14	{B→C, D→C, C→D}	A→B C D	{A, C, D}	I O I I
15	{A→B, B→C, D→C, C→D}	A B C D	{A, B, C, D}	I I I I

Table 1: ISS of the running example $F = \mathsf{A} \to \mathsf{B} \to \mathsf{C} \longleftrightarrow \mathsf{D}$. All 16 suspension states, ordered by number of suspensions. *State* D lists the suspended attacks; $F - D$ shows the resulting AF, with those attacks omitted; $\mathcal{S}_D$ and $\mathrm{gr}(F-D)$ give its preferred solutions and grounded labeling (order: A, B, C, D; I/O/U abbreviate IN/OUT/UNDEC). Moving *down the rows* is Y (*change*: one AF variant per row); reading *across* the $\mathcal{S}_D$ column is X (*choice*: the solutions, side by side). $\mathcal{S}_D$ colors correspond to node colors in the suspension lattice of Figure 2.

argument's acceptance status (an X-property) persists across all ways of resolving uncertainty (a Y-region). Relevance [9, 7] asks which uncertain attacks to resolve (which Y-moves to make) to stabilize the argument's status. Enforcement [10, 11] seeks the smallest structural change (a Y-move) that makes a desired set of arguments an extension (an X-target). Grounding extensions [5] finds minimal sets of attacks to suspend (a Y-path) that make a chosen solution the grounded extension, collapsing X to a single point. Counterfactual reasoning for AFs [12, 13] asks what would change if an argument's status were different, which can involve removing attacks (a change in the Y-dimension) and observing the effect on other arguments' acceptance (X-dimension).

Additionally, new AF-related concepts can be revealed by studying how the dimensions interact across the ISS. The following examples are further developed in Sec. 3:

- Suspending one further attack (here, e.g., B→C, after A→B) can *increase* the number of preferred solutions. So resolving ambiguity is not the only thing suspensions do: they can also *create new ambiguity*.
- Two states with identical solutions (e.g., rows 0 and 2 in Table 1) diverge after one and the same additional suspension, i.e., "*semantically equivalent here*" does not imply "*equivalent later*" (after further suspensions).
- The skeptical acceptance of an argument (C in Figure 2) can hold, fail, and hold again along a single suspension path, i.e., *acceptance may come and go*.

This paper is an initial exploration of the ISS. We develop the framework, identify new concepts, analyze

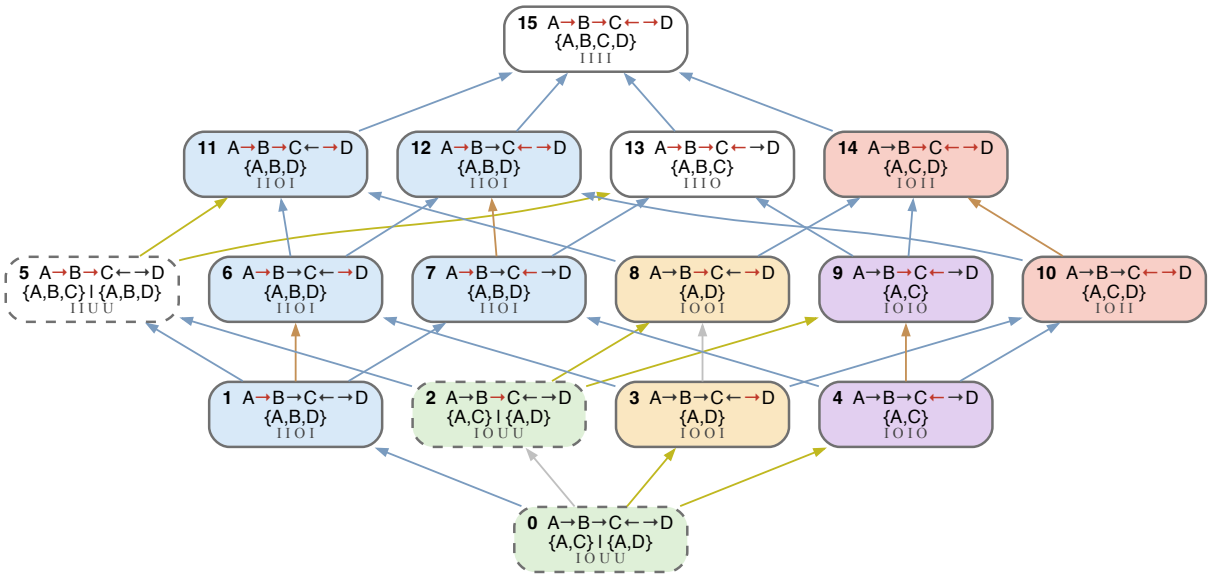


Figure 2: The suspension lattice (Hasse diagram). Each node shows its row number in Table 1, the AF in force (suspended attacks red), its preferred solutions, and its grounded labeling. Node 0 (bottom, Table 1's first row) is the intact AF; node 15 (top) has all attacks suspended. *Y-navigation* (change) follows the edges upward, one suspension per edge (colored by attack type); *X-navigation* (choice) happens inside a node, between the solutions listed. Equal-solution nodes share a fill color: together they form the non-trivial *Plateaus*, connected by *solution-preserving* steps (white nodes are singleton Plateaus). Dashed borders mark states whose grounded labeling contains UNDEC arguments. Nodes 0 and 2 form a *Shadow pair*: one Plateau, yet suspending A→B sends the two states to different solution sets (nodes 1 vs. 5), so merging them is unsound (Prop. 1).

the structure of the space, and validate it on examples from the literature. Our main contributions are:

- *A two-dimensional ISS perspective* (Sec. 2). We formalize the ISS as a suspension lattice with solution sets.
- *New ISS concepts and their minimal specimens* (Sec. 3). We formally define six ISS concepts, illustrate each with a running example, and determine the smallest AFs witnessing each through an exhaustive search.
- *Behavioral equivalence* (Sec. 4.1). A natural idea is to merge states with identical solutions. We show that this is unsound in general and define a *sound quotient* based on behavioral equivalence that determines when states are interchangeable under further suspensions.
- *Attack types have structural consequences* (Secs. 4.2–4.3). The grounded labeling classifies each attack by the acceptance status of its arguments [14, 3], yielding different attack types with distinct effects on the ISS. We show that certain types when suspended are always solution-preserving, that reaching a target grounded extension requires only a specific type of suspension, and identify a structural condition under which solution-preserving steps can be safely merged.
- *Algorithms and implementation* (Sec. 5). We present two algorithms for materializing the ISS and finding minimal specimens, implemented in an open-source prototype [15].

2. Argumentation Frameworks and the ISS

We begin with standard AFs, then introduce the ISS and its connections to existing approaches.

2.1. Preliminaries

We assume finite, abstract AFs $F = (A, R)$ with arguments A and attacks $R \subseteq A \times A$ [1]. A set $S \subseteq A$ is *conflict-free* if no two members attack each other; S *defends* a if every attacker of a is attacked by some member of S; S is *admissible* if it is conflict-free and defends all its members; and S is *complete*

if it is admissible and contains every argument it defends. The *preferred* extensions $\mathrm{pr}(F)$ are the $\subseteq$-maximal admissible sets; the *grounded* extension $\mathrm{gr}(F)$ is the least complete extension. Given an extension, a *labeling* uniquely assigns each argument IN, OUT, or UNDEC (I, O, U for short) [2]. An argument is *credulously accepted* if it belongs to *some* preferred extension, and *skeptically accepted* if it belongs to *all*. Under a semantics σ, the extensions in $\sigma(F)$ are its *solutions*. Although the X-dimension can be instantiated with any fixed semantics, unless otherwise stated, we take the preferred extensions $\mathrm{pr}(F)$ as the *solutions* of F, making the choice dimension non-trivial. The grounded extension $\mathrm{gr}(F)$ of F and its labeling provide a unique baseline.

2.2. Intervention-Solution Spaces

For $R' \subseteq R$, the *restriction* of F by R' is $F - R' = (A, R \setminus R')$ [5, 16]. We read R' as *suspended*: the attacks are ignored, not destroyed (the original AF sits at the bottom of the suspension lattice). Attacks in $R \setminus R'$ are called *active*. The lattice is read *directionally*, where moving upward implies additional suspensions, and a *future* state $D' \supseteq D$ is one with further interventions applied. This matches the enforcement reading (interventions compound) and the suspension of critical attacks in [3, 4]: the sets R' with $\mathrm{gr}(F-R') \in \sigma(F)$ are exactly the σ-*critical* sets of [5].

Definition 1 (Intervention-Solution Space)**.** *Given $F = (A, R)$, its* suspension lattice *is the Boolean lattice $(2^R, \subseteq)$ and its elements D (sets of suspended attacks) are* suspension states*. Lattice edges $D \to D \cup \{e\}$, directed toward larger D, are called* cuts*. The* solution set *of D is $\mathcal{S}_D = \mathrm{pr}(F-D)$. $\mathrm{ISS}(F)$ is the suspension lattice together with the* solution map *$D \mapsto \mathcal{S}_D$.*

Each ISS point is a pair (state D, solution $S \in \mathcal{S}_D$). Moving in *X* means picking a different solution of the *same* AF and moving in *Y* means suspending one more attack. The axes differ: in Y, states are connected by single suspensions, giving distance and paths, whereas X is an unordered set. Let $\mathcal{D} \subseteq 2^R$ denote a nonempty domain of suspension states in an ISS. Depending on the application, $\mathcal{D}$ may be the full ISS, the future states reachable from a chosen state D, or the sublattice induced by a selected set of suspendable attacks. Queries over $\mathcal{D}$ naturally decompose over the two dimensions:

	$\exists$ **solution** (*credulous*)	$\forall$ **solutions** (*skeptical*)
$\exists$ **state** (*possible*)	*possibly credulous*	*possibly skeptical*
$\forall$ **states** (*necessary*)	*necessarily credulous*	*necessarily skeptical*

The columns (X-dimension) distinguish credulous from skeptical acceptance. The rows (Y-dimension) quantify over $\mathcal{D}$: a property is *robust* in $\mathcal{D}$ if it holds at every state and *achievable* in $\mathcal{D}$ if it holds at some state. When $\mathcal{D}$ consists of the future states of a state D, these correspond to guarantees and opportunities under further suspension. When $\mathcal{D}$ is induced by an IAF, they correspond to necessary and possible acceptance over its completions [8]. Together, the four cells represent basic queries about an argument's status across the selected state domain. Beyond these point queries, the lattice structure also supports path queries along suspensions: for instance, which states share a particular solution, which suspension paths connect them, and what happens to solutions along a path.

2.3. Connections to Existing Approaches

As noted in the introduction, several existing lines of work can be understood as different queries over the ISS. We briefly describe each connection below.

Incomplete AFs [8, 17]. In an incomplete AF (IAF) some arguments and attacks are assumed to be *uncertain*, i.e., it is not known whether they are present. Each way of resolving the uncertainty is a *completion*, turning the IAF into a concrete AF. For an IAF in which only attacks $R_u \subseteq R$ are uncertain, the completions are a subset of the ISS states, one for each way of resolving the uncertainty. Thus, R_u induces a sublattice $(2^{R_u}, \subseteq)$ contained within the full ISS. Over this sublattice, totality, determinism, and functionality ask whether a goal is never UNDEC, always has the same label, or both, across its solutions and states [18]. The ISS adds structure beyond the completion set, e.g., distances between

completions, how solutions change along paths, and whether completions with identical solutions remain equivalent under further changes.

Stability and Relevance [9, 7]. Stability asks whether an argument's acceptance status persists across all completions of an IAF, corresponding to the "necessarily" row of the ISS quantifier matrix. Relevance asks which of the uncertain attacks in R_u stabilize the argument's status. Acceptance is not monotone along ISS suspensions in general (Def. 5), so the acceptance status at the bottom and top of a suspension region does not determine stability throughout that region. The attack types of Sec. 4.3 identify which suspensions preserve grounded-label monotonicity and which may reverse decided labels.

Enforcement [10, 11]. Enforcement asks how to modify an AF so that a desired set of arguments becomes an extension while minimizing the number of changes. With ISS, enforcement can be viewed as a shortest-path query, i.e., find the fewest Y-moves (suspensions) that achieve an X-target (the desired extension). Because an ISS captures the full space of such modifications, enforcement queries become path distance computations in the lattice. ISS as defined covers attack *removal*. Additions can be handled by embedding F in a larger lattice that includes hypothetical attacks.

Grounding extensions [5, 3, 4]. The grounding extension problem is a special case of enforcement. A set $D \subseteq R$ is σ-critical if $\mathrm{gr}(F{-}D) \in \sigma(F)$, and minimally σ-critical if no proper subset of D is σ-critical. The problem of finding minimally σ-critical sets therefore amounts in an ISS to finding inclusion-minimal suspension states D satisfying $\mathrm{gr}(F{-}D) \in \sigma(F)$.

Counterfactual and semifactual reasoning [12, 19, 13, 20]. Counterfactual reasoning asks how changing the status of one or more arguments affects a goal argument. Modification-based approaches change the AF and then inspect its solutions [12]. A more recent approach permits interventions on several labels while keeping other selected labels fixed and checks whether the goal's label changes in some or all complete labelings after the intervention [20]. In ISS terms, these are analogous to Y-moves followed by X-queries. By contrast, another approach keeps F fixed and seeks a labeling that changes the goal while changing as few other labels as possible [19]. The corresponding semifactuals preserve the goal's status while maximizing the number of other arguments whose labels change. Both queries are expressed entirely within the X-dimension.

The rest of the paper explores the ISS itself: new concepts that emerge from the interaction of its two dimensions (Sec. 3), the structure of its state space (Sec. 4), and algorithms for materializing and searching it (Sec. 5).

3. ISS Concepts and Minimal Specimens

We define several ISS concepts that capture how suspensions and solutions interact. For each, we also determine the smallest AF in which it occurs (its *minimal specimen*, Table 2), as a measure of how much structure each concept requires. We separate minimal specimens into those that allow self-attacks and those that do not. In the following, we assume an AF $F = (A, R)$, suspension states $D, D_i \subseteq R$, an attack $e \in R$, and write $n_D = |\mathcal{S}_D|$ for the number of solutions at state D. The *impact* of a cut is the number of arguments whose grounded label changes, i.e., $\mathrm{imp}(D, e) = |\{a \in A \mid$ the grounded label of a differs between $F{-}D$ and $F{-}(D \cup \{e\})\}|$.

Definition 2 (Domino)**.** *A cut $D \to D \cup \{e\}$ is a* Domino *if $\mathrm{imp}(D, e) = |A|$, i.e., every argument changes its grounded label.*

Domino is the simplest ISS concept with the smallest minimal witnesses. A minimal self-loop-free specimen is a↔b (Table 2): suspending either attack flips both labels. With self-attacks, a → a suffices: suspending the single attack changes a from UNDEC to IN.

Definition 3 (Plateau)**.** *A* Plateau *is a maximal set of states connected by cuts that preserve the solution set. A Plateau is* non-trivial *if it contains at least two states.*

Concept	Self-loop-free minimal witness	With self-attacks
Domino	$(2,2)$: the choice a↔b	$(1,1)$: a→a
Non-triv. Plateau	$(3,2)$: the chain a→b→c	$(2,2)$: a→a, b→a
Shadow	$(3,2)$: the chain a→b→c	$(2,2)$: a→a, b→a
Chameleon	$(3,2)$: the chain a→b→c	$(2,3)$: a→a, a↔b
Hydra	$(3,3)$: a→b↔c	$(2,3)$: a→a, a↔b
Phoenix	$(4,6)$: a attacks each of {b, c, d}; b→c→d→b	$(2,3)$: a→a, a↔b

Table 2: Minimal specimens: one smallest AF per cell, ordered by $(|A|, |R|)$ (arguments, attacks), up to isomorphism. The middle column excludes self-attacks; the right column allows them. Machine-verified by Algorithm 2. "Non-trivial Plateau" means a solution-preserving one-suspension step exists.

In Table 1, states sharing a fill color in Figure 2 form non-trivial Plateaus, e.g., states 0 and 2 (both with solutions {A, C}, {A, D}). A minimal self-loop-free specimen is the reinstatement chain a→b→c (Table 2): suspending b→c preserves the solution {a, c}, so the two states form a non-trivial Plateau.

Definition 4 (Shadow Pair). *A* Shadow pair *is a pair of states $D_1 \neq D_2$ with $\mathcal{S}_{D_1} = \mathcal{S}_{D_2}$ such that for some $e \notin D_1 \cup D_2$, $\mathcal{S}_{D_1 \cup \{e\}} \neq \mathcal{S}_{D_2 \cup \{e\}}$, i.e., equal solutions now, but different futures.*

In Table 1, rows 0 and 2 form a Shadow pair: both carry solutions {A, C}, {A, D} (one Plateau), yet suspending A→B yields one solution from row 0 (row 1) but two from row 2 (row 5). A minimal self-loop-free specimen is again the chain (Table 2): states ∅ and {b→c} share the solution {a, c}, but suspending a→b yields different solutions from each.

Definition 5 (Chameleon). *A* Chameleon *is an argument whose skeptical acceptance holds at D_1, fails at some $D_2 \supsetneq D_1$, and holds again at some $D_3 \supsetneq D_2$.*

In Table 1, argument C is skeptically accepted at row 4, lost at row 7 (suspending A→B frees B to attack C), and regained at row 13 (suspending B→C removes the attack). A minimal self-loop-free specimen is again the chain (Table 2).

Definition 6 (Hydra). *A cut $D \to D \cup \{e\}$ is a* Hydra *if $n_{D \cup \{e\}} > n_D$, i.e., suspending the attack increases the number of solutions.*

In Table 1, while suspending A→B leaves a single solution {A, B, D} (row 1), further suspending B→C yields two solutions (row 5). In this example, the second suspension increases ambiguity. When disallowing self-attacks, a minimal Hydra specimen is a→b↔c (Table 2): the AF has one preferred extension {a, c}, and suspending a→b frees b, splitting it into two preferred extensions {a, b} and {a, c}.

Definition 7 (Phoenix). *A* Phoenix *is a solution E with $E \in \mathcal{S}_{D_1}$, $E \notin \mathcal{S}_{D_2}$, $E \in \mathcal{S}_{D_3}$ for some chain $D_1 \subsetneq D_2 \subsetneq D_3$, i.e., a solution that disappears and reappears.*

While a Chameleon tracks an argument's acceptance, a Phoenix tracks an entire solution. A minimal self-loop-free specimen is a attacking each of {b, c, d} with b→c→d→b (Table 2). The solution {a} is a preferred extension of the original AF, but when a→b is suspended, {a} is no longer maximal (the larger {a, b} is preferred instead). When all of a's attacks are suspended, {a} reappears as a preferred extension. In this example, the grounded labelings at the two endpoint states differ. In the original AF, a is IN and b, c, d are OUT. When all of a's attacks are suspended, a is IN but b, c, d are now UNDEC. However, this change of labeling is not required in general for a Phoenix. Consider the self-loop-free AF $R = \{a \to b, a \to c, b \to a, c \to b, c \to d, d \to a\}$. For $D_1 = \varnothing$, $D_2 = \{c \to b\}$, and $D_3 = \{c \to b, b \to a\}$, the empty set is the sole preferred extension at D_1 and D_3. At D_2, the sole preferred extension is $\{b, c\}$. In this case, every argument in the grounded labeling is UNDEC at all three states. Thus a Phoenix can arise solely from changes in preferred maximality while the grounded labeling remains unchanged.

The specimens in Table 2 are exhaustive: 287 non-isomorphic AFs were searched for $|A| \leq 3$ (with and without self-attacks) and for $|A| = 4$ with $|R| \leq 7$ in the self-loop-free case. The reinstatement chain a→b→c is a minimal witness of three concepts (Plateau, Shadow, and Chameleon). With self-attacks allowed, a single AF (a→a, a↔b) witnesses Hydra, Phoenix, and Chameleon, suggesting that the self-loop-free regime is more informative [21]. Prior work has observed such effects individually (e.g., expansion or restriction of extension sets [22], persistence of acceptance under change [23]), but as one-step or axiomatic facts. In an ISS, they become path and adjacency properties of the lattice.

4. Exploring the Structure of ISS

The suspension lattice grows exponentially with the number of attacks, so queries over the structure can be prohibitively expensive unless states can be combined or pruned. Multiple notions of equivalence have been defined based on whether two frameworks remain interchangeable under future modifications [24, 25]. Here we adapt this idea to the ISS setting, asking when two suspension states are interchangeable for queries about current solutions and what further suspensions can produce. We formalize a notion of behavioral equivalence that requires two states to have the same solutions now and matching behavior under all further suspensions. We then introduce an attack-type classification based on the grounded labeling [3] that identifies which suspensions are solution-preserving, which states can be safely merged, and which types of suspensions are necessary for certain ISS concepts to arise.

4.1. State Equivalence under Suspensions

We consider notions of equivalence between ISS states, starting with the most basic notion of whether two states have the same solutions.

Definition 8 (Solution Equivalence)**.** *States D_1 and D_2 are* solution-equivalent *if $\mathcal{S}_{D_1} = \mathcal{S}_{D_2}$.*

Every X-dimension query (e.g., credulous and skeptical acceptance, extension existence) gives the same result on solution-equivalent states.

The solution-equivalence class of a state D is the set $[D]_{\text{sol}} = \{D' \mid \mathcal{S}_{D'} = \mathcal{S}_D\}$. Such a class can be disconnected in the lattice, with states in different regions sharing the same solutions but no solution-preserving path between them. Plateaus (Def. 3) refine solution-equivalence classes by requiring connectivity, so that every pair of states in a Plateau is linked by solution-preserving cuts. Plateaus therefore identify maximal connected regions in which suspensions leave the current solution set unchanged. In the running example (Table 1), the lattice of Figure 2 contains five non-trivial Plateaus (visible as the colored groups). Treating each such region as a single unit reduces the 16 states to 8 units.

Connectivity, however, does not make states within a Plateau interchangeable under further suspensions. In Figure 2, states 6 and 7 share the same Plateau (solution $\{\mathsf{A}, \mathsf{B}, \mathsf{D}\}$), but suspending B→C leads to different solutions (state 11 vs. state 13). This is a Shadow pair (Def. 4), leading to the following negative result.

Proposition 1 (Solution Equivalence Is Not Suspension-Invariant)**.** *Solution equivalence, even within a single Plateau, does not guarantee equivalent solution sets after suspending the same attacks.*

Proof. For a→b→c, states $D_1 = \varnothing$ and $D_2 = \{\mathsf{b}{\to}\mathsf{c}\}$ are solution-equivalent, both with solution $\{\mathsf{a}, \mathsf{c}\}$. The states are joined by the solution-preserving cut b→c, so they lie in the same Plateau. Suspending a→b in D_1 yields solution $\{\mathsf{a}, \mathsf{b}\}$, while in D_2 it yields $\{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$. □

This result parallels the insight behind *strong equivalence* for AFs [24, 25], where two frameworks with the same extensions can produce different extensions after adding arguments or attacks. In an ISS, the analogous modifications are different sets of attack suspensions from an AF.

Given Prop. 1, we seek an equivalence that preserves the solution behavior available under further suspensions. We therefore use a definition based on *bisimulation*, where two states are equivalent if they can match each other's reachable futures.

Definition 9 (Behavioral Equivalence)**.** *States D_1 and D_2 are* behaviorally equivalent *if (i) $\mathcal{S}_{D_1} = \mathcal{S}_{D_2}$; and (ii) for every future state D_1' reachable from D_1, there exists a future state D_2' reachable from D_2 such that D_1' and D_2' are behaviorally equivalent, and vice versa. Behavioral equivalence is the largest relation satisfying (i) and (ii).*

Reachability is reflexive (i.e., permits zero additional suspensions), so every state is reachable from itself. Because the ISS lattice is finite, this largest relation can equivalently be obtained by beginning with all pairs of states that satisfy condition (i), then repeatedly removing any pair of states for which a future of either state has no matching future of the other state among the pairs currently retained. When no further pair can be removed, the remaining pairs constitute behavioral equivalence. Pairs of the form (D, D) are never removed, since every future of a state can be matched by that same future. Therefore every state is behaviorally equivalent to itself.

In Figure 2, states 11 and 12 form an initial candidate pair because they have the same solution. When the pair is tested under condition (ii), future 11 is matched by future 12 using the candidate pair $(11, 12)$, while their only additional future, state 15, matches itself. The reverse direction is symmetric, so the pair is never removed. State 6 forms an initial candidate pair with each of states 11 and 12 because all three have the same solution. Consider pair $(6, 11)$. Future 6 is matched by future 11 using the candidate pair itself. Future 11 matches itself, future 12 is matched by 11 using the surviving pair $(12, 11)$, and future 15 matches itself. In the reverse direction, the futures 11 and 15 of state 11 are also reachable from state 6 and match themselves. Thus $(6, 11)$ is never removed. The pair $(6, 12)$ survives by the analogous argument. This comparison also shows that a matching state does not have to be reached by suspending the same attack. Suspending D→C from state 6 leads to state 12, whereas the same suspension from state 11 leads to state 15. Under condition (ii), however, state 12 is matched by state 11 itself rather than by state 15. The set $\{6, 11, 12\}$ is the only non-trivial behavioral-equivalence class in Figure 2.

By contrast, state 7 is not behaviorally equivalent to any member of this class. Although it satisfies condition (i) initially, it can reach state 13, whose solution $\{\mathsf{A}, \mathsf{B}, \mathsf{C}\}$ is not the solution of any future reachable from states 6, 11, or 12. Because state 13 has no possible match, the corresponding candidate pairs are removed under condition (ii).

As a bisimilarity over reachable futures, behavioral equivalence is an equivalence relation. From behaviorally equivalent states, matching future states may be reached by suspending different attacks or by following suspension paths of different lengths.

The behavioral-equivalence class of a state D is the set $[D]_{\text{beq}} = \{D' \mid D' \text{ is behaviorally equivalent to } D\}$. Grouping states into such classes suggests a natural reduction. Collapse each behavioral-equivalence class $[D]_{\text{beq}}$ into a single abstract state, with transitions inherited from the lattice.

Definition 10 (Sound Quotient)**.** *The* sound quotient *of an ISS is the reduced transition structure whose states are behavioral-equivalence classes $[D]_{\text{beq}}$, with a transition from $[D_1]_{\text{beq}}$ to $[D_2]_{\text{beq}}$ whenever some state in $[D_1]_{\text{beq}}$ has a cut leading to a state in $[D_2]_{\text{beq}}$.*

Because Def. 9 defines behavioral equivalence as the largest relation satisfying conditions (i) and (ii), the sound quotient is the coarsest quotient whose classes contain only solution-equivalent states with matching reachable futures. It is computed by partition refinement over the lattice (Sec. 5). Whether structural conditions on the AFs $F-D_1$ and $F-D_2$ can determine behavioral equivalence without full partition refinement, analogous to the kernel characterizations of strong equivalence [24], is an open question. The shielded-cut condition of Sec. 4.2 provides a partial answer.

4.2. Attack Types and Behavioral Equivalence

Behavioral equivalence is defined globally over the lattice. We now introduce a local classification of attacks based on the grounded labeling [3]. The classification identifies solution-preserving suspensions and sufficient conditions for behavioral equivalence. In the following, we write $L_D(x)\to L_D(y)$ to denote the label pair of an attack $x\to y$ in $F - D$, e.g., I→O means $L_D(x) = \text{I}$ and $L_D(y) = \text{O}$.

Definition 11 (Attack Type [3]). *Given suspension state D and the grounded labeling L_D of $F - D$, the* attack type $\tau(x{\to}y)$ *of an attack $x{\to}y$ in $F - D$ is determined by the labels of x and y.*

$$\tau(x{\to}y) = \begin{cases} \text{successful } \textit{(S-attack)} & \textit{if } \mathrm{I}{\to}\mathrm{O} \\ \text{failed } \textit{(F-attack)} & \textit{if } \mathrm{O}{\to}\mathrm{I} \\ \text{undecided } \textit{(U-attack)} & \textit{if } \mathrm{U}{\to}\mathrm{U} \\ \text{moot } \textit{(M-attack)} & \textit{if } \mathrm{O}{\to}\mathrm{O},\ \mathrm{O}{\to}\mathrm{U},\ \mathrm{U}{\to}\mathrm{O} \end{cases}$$

An S-attack is one where x successfully defeats y. An F-attack has failed because x is itself defeated. A U-attack is undecided since it is not yet determined whether x defeats y. An M-attack is irrelevant to the labeling because the target's status is already determined by another attacker: an OUT target has at least one IN attacker, while an UNDEC target has at least one UNDEC attacker [2]. The remaining three label pairs (I→I, I→U, U→I) cannot occur in a valid grounded labeling, since an IN attacker forces its target OUT and an UNDEC attacker prevents its target from being IN [2]. The attack types are determined directly by the grounded labeling, which is computable in linear time from $F - D$ [26]. In the running example, attack types are shown as edge colors in Figure 1(b)–(d): blue for S-attacks, orange for F-attacks, gray for M-attacks, and yellow for U-attacks. In Figure 2, the lattice-edge colors denote the attack type of the suspended attack at each state.

The attack types raise a natural question: what is the impact of each type on ISS solutions? We first consider the case when attacks have no effect, i.e., preserve a state's solutions.

Definition 12 (Solution-Preserving). *Given semantics σ and suspension state D, an attack $x{\to}y$ in $F - D$ is* solution-preserving *at D relative to σ if $\sigma(F{-}D) = \sigma(F{-}(D \cup \{x{\to}y\}))$.*

Proposition 2 (Attack-Type Solution Preservation). *F-attacks and M-attacks preserve the grounded labeling and the set of preferred extensions at every suspension state D.*

Proof (sketch). Assume labels refer to the grounded labeling. Removing an F-attack or M-attack leaves the labeling unchanged: an attack from an OUT source does not determine its target's label, while an OUT target retains an IN attacker distinct from the removed attack's source. Every F-attack or M-attack involves at least one OUT argument. Every such argument is attacked by an argument labeled IN in the grounded labeling. This IN attacker belongs to every preferred extension because every preferred extension contains the grounded extension. Conflict-freeness therefore prevents the OUT argument from belonging to any preferred extension. If the removed attack has an OUT source, that source is already defeated in every preferred extension. If it has an OUT target, that target remains defeated by another IN attacker. Thus removing the attack neither removes necessary defense nor permits the OUT argument of the attack to enter a preferred extension, so the preferred extensions remain unchanged. □

Although suspending an attack may trigger a cascade of changes in the grounded labeling of an AF, S- and U-attack cuts provide useful persistence guarantees.

Lemma 3 (Grounded-Label Persistence). *Let $e = x{\to}y$ be an active attack at D, let $D' = D \cup \{e\}$, and consider the grounded labelings of $F - D$ and $F - D'$, respectively.*

(i) If e is a U-attack at D, every argument labeled IN or OUT at D retains its label at D'. Thus only arguments labeled UNDEC at D may change.

(ii) If e is an S-attack at D, its source x remains IN at D'.

Proof. For a finite AF, the grounded labeling can be constructed incrementally from unattacked IN arguments by repeatedly assigning OUT to arguments attacked by an IN argument and IN to arguments whose attackers are all OUT [2]. Every decided label therefore has a finite, well-founded justification rooted in unattacked arguments. For (i), every attack used in such a justification is an S- or F-attack at D. Since $x{\to}y$ is a U-attack at D, it occurs in no justification of a decided label. Suspending $x{\to}y$ therefore leaves these justifications intact, so every argument labeled IN or OUT at D retains its label

	Grounded Preserved		Preferred Preserved	
Type	Singleton	Non-singleton	Singleton	Non-singleton
F	Always	Always	Always	Always
M	Always	Always	Always	Always
S	Never	Sometimes	Sometimes	Sometimes
U	Never	Always	Sometimes	Sometimes

Table 3: Preservation properties by attack type and multiplicity. All cases marked "Sometimes" have machine-verified minimal witnesses (see [15]).

at D'. For (ii), a justification for the IN label of x cannot include $x \to y$. If it did, the IN label of x would be used to establish y as OUT, which would then have to contribute, possibly through other arguments, to establishing x as IN. Thus the label of x would depend on itself, creating a cycle and contradicting the well-foundedness of the justification. Suspending $x \to y$ therefore leaves a justification for x intact, so x remains IN at D'. □

S-attacks and U-attacks are sometimes, but not always, solution-preserving. An attack is *singleton* if its source is the target's only attacker having the source's grounded label. Otherwise, it is *non-singleton.* Thus, a singleton S-attack has the target's only IN attacker as its source, while a singleton U-attack has the target's only UNDEC attacker as its source. When suspended, singleton S- and U-attacks always change the grounded labeling.

Proposition 4 (Attack-Type Grounded Change)**.** *Singleton S- and U-attack cuts always change the grounded labeling.*

Proof (sketch). For a singleton S-attack $x \to y$, x is the only IN attacker of y. If the labeling remained unchanged after suspending $x \to y$, then y would remain OUT despite having no IN attacker, a contradiction. For a singleton U-attack, x is the target's only UNDEC attacker, and every other attacker is OUT. By Lemma 3(i), those attackers remain OUT. Removing $x \to y$ therefore makes y IN. □

The preceding proposition shows that suspending a singleton U-attack always changes the grounded labeling. The non-singleton case has the opposite effect.

Proposition 5 (Non-Singleton U-Attack Preservation)**.** *Suspending a non-singleton U-attack preserves the grounded labeling.*

Proof. Let $G = \mathrm{gr}(F - D)$, $x \to y$ be a non-singleton U-attack at D, and $D' = D \cup \{x \to y\}$. The argument y has another UNDEC attacker w. Neither x nor w belongs to G, and w is not attacked by G. Suspending $x \to y$ therefore removes no attack from G, and y remains undefended by G. Thus G remains a complete extension of $F - D'$, so $\mathrm{gr}(F - D') \subseteq G$. Conversely, Lemma 3(i) preserves every argument in G, giving $G \subseteq \mathrm{gr}(F - D')$. The grounded extensions are therefore equal. Since $x \notin G$, suspending $x \to y$ changes no attack originating in G, so the OUT labels are also unchanged. Thus the grounded labelings at D and D' are identical. □

Non-singleton S-attacks can either preserve or change the grounded labeling and preferred solutions. The Hydra specimen in Table 2 witnesses a change to both. By Prop. 5, non-singleton U-attacks always preserve the grounded labeling, although they can either preserve or change the preferred solutions by altering the conflict structure within the undecided region. Table 3 summarizes these cases.

Solution preservation alone does not guarantee behavioral equivalence because solution-preserving steps can still affect the outcome of subsequent cuts (Prop. 1). The following structural condition identifies when a solution-preserving step does preserve all future behavior.

Definition 13 (Shielded Cut)**.** *Given suspension state D and an attack $x \to z$ in $F - D$, the cut $D \to D \cup \{x \to z\}$ is* shielded *if x is unattacked in $F - D$ and z has another attacker $y \neq x$ in $F - D$ that is unattacked in $F - D$.*

In Figure 2 at state 6 both B and D are unattacked and both attack C. The cut suspending B→C is shielded (with D as the other unattacked attacker of C), and symmetrically the cut suspending D→C is shielded. A shielded cut necessarily suspends an S-attack since both x and the shielding attacker y are unattacked (hence IN).

Proposition 6 (Shielded Cuts Preserve Futures)**.** *If the cut $D \to D \cup \{e\}$ is shielded, then D and $D \cup \{e\}$ are behaviorally equivalent.*

Proof (sketch). Let $e = x \to z$. Since the cut suspending e is shielded, there is another attack $f = y \to z$ such that both x and y are unattacked in $F - D$. They remain unattacked, and hence IN, in every future state. Therefore, whenever at least one of e and f remains active, z is OUT and belongs to no admissible set.

To apply Def. 9 (behavioral equivalence), consider a candidate relation for behavioral equivalence that includes all identity pairs. For every future state D' of D in which neither e nor f is suspended, include every ordered pair formed from D', $D' \cup \{e\}$, and $D' \cup \{f\}$.

Identity pairs satisfy condition (i) immediately. In each of the three states, every preferred extension contains x and y and excludes z. The different choices of active attacks therefore neither affect conflict-freeness nor remove a defense against attacks from z, since whichever of e and f remains active has its source in every preferred extension. Therefore, the admissible sets containing x and y, and hence the preferred extensions, are the same in all three states. This means every other pair in the candidate relation also satisfies condition (i).

Suspending any other active attack g sends the above states, respectively, to $D' \cup \{g\}$, $D' \cup \{e, g\}$, and $D' \cup \{f, g\}$, which are paired by the construction above. Suspending e or f either leads to a state already paired by the candidate relation or to the common state $D' \cup \{e, f\}$, whose identity pair is included. Hence the candidate relation is closed under further suspensions and satisfies condition (ii). It therefore satisfies both conditions of Def. 9 and is contained in behavioral equivalence. Taking $D' = D$, states D and $D \cup \{e\}$ are thus behaviorally equivalent. □

In the running example, applying the proposition to the two shielded cuts from state 6 certifies that states 6, 11, and 12 belong to the same behavioral-equivalence class, without running partition refinement.

Shielded cuts provide a sufficient condition for safe merging in the sound quotient (Sec. 4.1). A shielded cut identifies two "mirror" S-attacks from distinct unattacked arguments to the same target. Since both sources are permanently IN and attack the same target, the associated states belong to the same equivalence class in the sound quotient. Although F-attacks and M-attacks are solution-preserving (Prop. 2), solution preservation alone does not certify safe merging. A future S-attack cut can make the suspended attack relevant and cause the states' futures to diverge (Prop. 1).

The shielded condition can be checked in linear time and provides a local certificate for particular safe merges without requiring global partition refinement. Which other solution-preserving steps preserve all future behavior remains an open question.

4.3. Attack Types and ISS Dynamics

Beyond behavioral equivalence, the attack-type classification constrains the dynamics of the ISS. It determines which suspensions can change the grounded labeling monotonically, which paths lead to grounding extensions, and which ISS concepts require which types to arise.

A cut is *grounded-monotonic* if the decided region of the grounded labeling can only expand: IN and OUT arguments keep their status, and UNDEC arguments may become decided. All but S-attack cuts have this property:

Proposition 7 (Grounded-Monotonic Attack Types)**.** *U-attack, F-attack, and M-attack cuts are grounded-monotonic in the suspension lattice.*

Proof (sketch). F- and M-attack cuts preserve the grounded labeling entirely by Prop. 2. U-attack cuts preserve every previously decided label by Lemma 3(i). Hence all three types are grounded-monotonic. □

Observation 8 (S-Attacks and Grounded-Monotonicity)**.** *Singleton S-attack cuts are not grounded-monotonic. Non-singleton S-attack cuts may also violate grounded-monotonicity depending on the surrounding structure.*

In the reinstatement chain, cutting a $\rightarrow$ b (S-attack) frees b, which attacks c, converting c from IN to OUT. Because S-attacks can reverse decided labels, an argument's acceptance status (credulous, skeptical, or grounded) can toggle along a suspension path, as witnessed by the Chameleon concept (Def. 5).

U-attack monotonicity has a direct consequence for grounding extensions (Sec. 2). Recall that a set $D \subseteq R$ is pr-critical if $\mathrm{gr}(F{-}D) \in \mathrm{pr}(F)$ [5]. Under preferred semantics, the attack types narrow the search for minimally pr-critical states.

Proposition 9 (pr-Critical Reachability)**.** *Let $S \in \mathrm{pr}(F)$ and let $D \subseteq R$ be $\subseteq$-minimal with $\mathrm{gr}(F{-}D) = S$. Every ordering of the attacks in D yields a path consisting only of U-attack cuts from the origin to the* pr*-critical state D.*

Proof. By Lemma 1 of [5], D is minimally pr-critical. Proposition 4 of [5] then gives, for every attack $x{\rightarrow}y \in D$, that $y \in S$ and S attacks x, thus $x \notin S$. Fix an intermediate suspension state $P \subseteq D$. Since every attack in D originates outside S, suspending P removes none of the attacks from S and cannot create a conflict within S. Thus S remains admissible in $F{-}P$. Every admissible set is contained in a preferred extension, and every preferred extension contains the grounded extension [2]. Therefore, under the grounded labeling at P, no member of S is OUT and no argument attacked by S is IN. For every remaining attack $x{\rightarrow}y \in D \setminus P$, we therefore have that x is not IN and y is not OUT, so the attack is not an S-attack. Since P was arbitrary, every cut along any ordering of D is an F-, M-, or U-attack and is therefore grounded-monotonic by Prop. 7. Hence every label decided at P persists through the remaining cuts.

It remains to rule out F- and M-attacks. Suppose that some remaining attack $x{\rightarrow}y \in D \setminus P$ is an F- or M-attack at P. Because y cannot be OUT, $x{\rightarrow}y$ can have either type only if x is OUT. Continue from P to $D' = D \setminus \{x{\rightarrow}y\}$ without suspending $x{\rightarrow}y$. All intervening cuts are grounded-monotonic, so x remains OUT. At D', y still cannot be OUT by the preceding argument, and thus $x{\rightarrow}y$ remains an F- or M-attack. Suspending it therefore preserves the grounded labeling by Prop. 2, giving $\mathrm{gr}(F{-}D') = \mathrm{gr}(F{-}D) = S$. Since $D' \subsetneq D$, this contradicts the $\subseteq$-minimality of D. Therefore every remaining attack at every intermediate state is a U-attack, and every ordering of D yields the claimed U-only path. □

The search for minimally pr-critical sets can be restricted to the U-reachable region of the ISS: the states reachable from the origin (i.e., the original AF) along paths consisting only of U-attack cuts. Not every state in this region is pr-critical. A U-attack cut can make the source and target of a suspended attack IN, which means the resulting grounded extension is not conflict-free in the original AF.

The monotonicity and solution-preservation results constrain which ISS concepts can arise from which cut types.

Proposition 10 (Domino Needs a U-Cut)**.** *A Domino (a cut with impact $|A|$) must be a U-attack cut.*

Proof (sketch). The result follows by eliminating the other three attack types. F- and M-attack cuts preserve the grounded labeling by Prop. 2. For an S-attack $x{\rightarrow}y$, the source x remains IN after the cut by Lemma 3(ii). Thus at least one argument retains its label, so the cut's impact is at most $|A| - 1$. Therefore a cut with impact $|A|$ must be a U-attack cut. □

As an immediate corollary, a Domino can occur only at a state where every argument is labeled UNDEC. By the preceding proposition, a Domino must be a U-attack cut, and Lemma 3(i) ensures that every previously decided label survives such a cut. Since a Domino changes all $|A|$ labels, there can be no previously decided argument.

A Chameleon (Def. 5) asks whether an argument's skeptical acceptance can toggle along a suspension path. A stronger form asks whether the grounded label itself can toggle.

Algorithm 1: ATLAS (sweeping the ISS and annotating its concepts)

Input: AF $F = (A, R)$
Output: annotated ISS with concept flags and sound quotient
// Phase 1: compute ISS states
foreach $D \subseteq R$ **do**
  $\mathcal{S}_D \leftarrow \mathrm{pr}(F - D)$; $L_D \leftarrow$ grounded labeling of $F - D$;
// Phase 2: build and annotate lattice edges
foreach *cut* $D \to D \cup \{e\}$ **do**
  add edge $D \to D \cup \{e\}$ to the ISS;
  classify $\tau(e)$ at state D;
  record $\mathrm{imp}(D, e)$; flag *Domino* if $\mathrm{imp}(D, e) = |A|$; flag *Hydra* if $n_{D\cup\{e\}} > n_D$;
// Phase 3: detect concepts and compute quotient
Plateaus: connected components of solution-preserving cuts;
Shadow: for each group of solution-equivalent states, check if a cut produces different solutions;
Chameleon: for each argument, sweep up and down the lattice to find a toggling chain;
Phoenix: for each solution, sweep up and down the lattice to check for reappearance;
Quotient: refine solution-equivalence classes by the sets of classes reachable from each state;

Proposition 11 (Grounded Chameleon Needs an S-Attack)**.** *Any suspension path along which an argument's grounded label changes and later returns to a previous value must include at least one S-attack cut.*

Proof (sketch). F-attack and M-attack cuts do not change labels (Prop. 2). U-attack cuts are grounded-monotonic (Prop. 7), so labels can only move from UNDEC to IN or OUT along paths without S-attack cuts. Toggling requires a reversal, which is impossible without at least one S-attack cut. □

The general Chameleon (skeptical acceptance) does not require S-attack cuts. Skeptical acceptance can toggle via U-attack cuts alone because a non-singleton U-attack cut can change the preferred extensions without changing the grounded labeling. For example, consider the AF on $\{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}\}$ with $R = \{\mathsf{a}{\to}\mathsf{b}, \mathsf{a}{\to}\mathsf{c}, \mathsf{a}{\to}\mathsf{d}, \mathsf{b}{\to}\mathsf{a}, \mathsf{b}{\to}\mathsf{c}, \mathsf{c}{\to}\mathsf{b}, \mathsf{d}{\to}\mathsf{b}\}$. Let $D_1 = \varnothing$, $D_2 = \{\mathsf{a}{\to}\mathsf{b}\}$, and $D_3 = \{\mathsf{a}{\to}\mathsf{b}, \mathsf{a}{\to}\mathsf{c}\}$. At all three states, every argument is UNDEC under the grounded labeling. Thus, each of the two successive cuts suspends a U-attack. The sole preferred extensions of D_1, D_2, and D_3 are $\{\mathsf{a}\}$, $\varnothing$, and $\{\mathsf{a}, \mathsf{c}\}$, respectively. Therefore, along the path $D_1 \to D_2 \to D_3$, skeptical acceptance of a toggles: it holds at D_1, fails at D_2, and holds again at D_3.

5. Algorithms and Prototype

We present two algorithms for exploring the ISS and finding minimal witnesses of its concepts. The open-source prototype [15] (Python and Clingo) computes the ISS, detects all concepts, classifies attack types, computes the sound quotient, and provides an interactive web-based explorer. All examples and tables in this paper are machine-verified using the prototype. Because the ISS grows exponentially with the number of attacks ($2^{|R|}$ states), full materialization is currently practical only for AFs with a modest number of attacks without further optimizations. Many practical ISS queries may not require full materialization, and incremental recomputation of extensions across neighboring states [27] may help scale the approach to larger AFs. The rest of this section describes the two algorithms: ATLAS (Algorithm 1) for materializing the ISS, and SYNTH (Algorithm 2) for finding minimal specimens.

As shown in Algorithm 1, ATLAS materializes the ISS of an AF and annotates it with the concepts of Sec. 3, the sound quotient of Sec. 4.1, and the attack types of Sec. 4.2. Let $N = 2^{|R|}$ be the number of states and $E = |R| \cdot 2^{|R|-1}$ the number of lattice edges. Materializing the full ISS requires enumerating the preferred extensions and computing the grounded labeling once at each of the N suspension states. The number of preferred extensions at a state may be exponential in $|A|$.

Thus, the total materialization cost depends on N and on the work needed to enumerate and store the preferred extensions at each state. The following post-processing bounds assume that solutions

Algorithm 2: SYNTH (finding a minimal specimen of a concept)

Input: concept P; search bounds n, m
Output: smallest AF exhibiting P, in $(|A|, |R|)$ order, up to isomorphism

```
foreach digraph F on at most n nodes and at most m edges, in ascending (|A|, |R|) order, up to
    isomorphism (canonical forms by permutation minimization) do
    run Algorithm 1 on F;
    if P occurs then return F;
```

and grounded labels have been stored, that each distinct solution set is assigned a unique identifier for constant-time equality comparisons, and that credulous and skeptical acceptance have been precomputed for each argument at each state.

Hydra detection takes $\mathcal{O}(E)$ time because it compares the number of solutions across each lattice edge. Domino detection takes $\mathcal{O}(|A| \cdot E)$ time because it compares the grounded label of every argument across each edge. Plateaus are connected components of the solution-preserving edges and can therefore be computed in $\mathcal{O}(N+E)$ time. For a fixed argument, Chameleon detection uses upward and downward lattice sweeps taking $\mathcal{O}(N + E)$ time, so checking all arguments takes $\mathcal{O}(|A| \cdot (N + E))$. Phoenix detection uses the same sweeps for each distinct solution. If K distinct solutions occur in the ISS, its cost is $\mathcal{O}(K \cdot (N + E))$. The current Shadow-pair detector groups solution-equivalent states, then checks each pair within a group to determine whether suspending any attack active in both states produces different solution sets. In the worst case, all N states belong to one group, yielding $\mathcal{O}(N^2)$ pairs with up to $|R|$ attacks checked per pair, for $\mathcal{O}(|R| \cdot N^2)$ time.

The sound quotient is computed by first grouping states by solution equivalence and then repeatedly subdividing the groups until all states within each group can reach the same set of groups. Its cost depends on the representation of reachability and the number of refinement iterations, rather than being a single linear pass. Algorithm 2 (SYNTH) finds minimal specimens by enumerating AFs until one exhibits the target concept, yielding the specimens in Table 2. For this table, SYNTH is run separately over self-loop-free AFs and over unrestricted AFs. In each case, AFs are searched first by increasing number of arguments ($|A| = 1, 2, \ldots$) and, for each $|A|$, by increasing number of attacks ($|R| = 0, 1, \ldots$).

The prototype has been applied to several examples from the literature, including F_1 from [5] (5 arguments, 6 attacks), the critical-attacks example from [3] (8 arguments, 11 attacks), and the Pierson v. Post AF from [4, 28] (10 arguments, 9 attacks). All ISS concepts of Sec. 3 except Phoenix appear in these examples, which are available in the prototype [15].

6. Conclusion and Outlook

We have proposed intervention-solution spaces (ISS) of AFs as an object of study, combining change (attack suspensions) and choice (among preferred extensions) as two coordinates of one finite structure. The ISS connects incomplete AFs [8], enforcement [10, 11], stability and relevance [9, 7], grounding extensions [5], and counterfactual reasoning [12]. We have introduced new concepts for characterizing AF behavior under modification, an attack-type classification with structural consequences, and a sound quotient, complementing prior results on strong equivalence for AFs [24, 25]. An open-source prototype with an interactive web-based explorer [15] implements both algorithms and all examples.

Future work includes scaling ISS construction and analysis through compact representations and computational reuse. Structuring the X-dimension would support queries such as finding small changes that reach an alternative solution. Another area is applying ISS to AF repair [29], where restoring consistency reduces to finding suitable suspensions, and studying which results carry over under semantics other than preferred. More generally, expressing existing AF problems as ISS lattice queries may suggest new analyses.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT and Claude to: draft specific content, paraphrase, reword portions of the text, and check grammar and spelling. After each tool's use, the authors reviewed, edited, and revised all content as needed and take full responsibility for the publication's content.

References

[1] P. M. Dung, On the Acceptability of Arguments and Its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games, Artificial Intelligence 77 (1995) 321–357.

[2] P. Baroni, M. Caminada, M. Giacomin, Abstract Argumentation Frameworks and Their Semantics, in: P. Baroni, D. Gabbay, M. Giacomin, L. van der Torre (Eds.), Handbook of Formal Argumentation, volume 1, College Publications, 2018, pp. 159–236.

[3] B. Ludäscher, Y. Xia, S. Bowers, Choices and their Provenance: Explaining Stable Solutions of Abstract Argumentation Frameworks, in: Proc. 17th Intl. Workshop on Theory and Practice of Provenance (TaPP), ACM, 2025, pp. 62–70.

[4] Y. Xia, H. Zheng, S. Bowers, B. Ludäscher, AF-XRAY: Visual Explanation and Resolution of Ambiguity in Legal Argumentation Frameworks, in: Proc. 20th Intl. Conf. on Artificial Intelligence and Law (ICAIL), 2025, pp. 495–497.

[5] L. Bengel, S. Bowers, B. Ludäscher, M. Thimm, On Grounding Extensions in Abstract Argumentation, in: Computational Models of Argument (COMMA 2026), IOS Press, 2026. To appear.

[6] K. D. Ashley, E. L. Rissland, A Case-Based Approach to Modeling Legal Expertise, IEEE Expert 3 (1988) 70–77.

[7] D. Odekerken, F. Bex, H. Prakken, Precedent-Based Reasoning with Incomplete Information for Human-in-the-Loop Decision Support, Artificial Intelligence and Law 34 (2026) 107–152.

[8] D. Baumeister, M. Järvisalo, D. Neugebauer, A. Niskanen, J. Rothe, Acceptance in Incomplete Argumentation Frameworks, Artificial Intelligence 295 (2021) 103470.

[9] D. Odekerken, A. Borg, F. Bex, Justification, Stability and Relevance in Incomplete Argumentation Frameworks, Argument & Computation 15 (2024) 251–308.

[10] R. Baumann, G. Brewka, Expanding Argumentation Frameworks: Enforcing and Monotonicity Results, in: Computational Models of Argument (COMMA 2010), IOS Press, 2010, pp. 75–86.

[11] A. Niskanen, J. P. Wallner, M. Järvisalo, Optimal Status Enforcement in Abstract Argumentation, in: Proc. 25th Intl. Joint Conf. on Artificial Intelligence (IJCAI), 2016, pp. 1216–1222.

[12] C. Sakama, Counterfactual Reasoning in Argumentation Frameworks, in: Computational Models of Argument (COMMA 2014), volume 266 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2014, pp. 385–396.

[13] M. Shao, S. Liu, B. Liao, A Causal Approach to Contrastive Explanation in Abstract Argumentation, in: Logics for New-Generation Artificial Intelligence (LNGAI 2025), College Publications, 2025.

[14] S. Bowers, Y. Xia, B. Ludäscher, The Skeptic's Argumentation Game or: Well-Founded Explanations for Mere Mortals, in: Proc. 5th Intl. Workshop on Systems and Algorithms for Formal Argumentation (SAFA), volume 3757 of *CEUR Workshop Proceedings*, 2024, pp. 104–118.

[15] S. Bowers, B. Ludäscher, ISS: Exploring Intervention-Solution Spaces (research prototype), https://github.com/idaks/ISS, 2026. Prototype ISS implementation, example suite, and web explorer (https://idaks.github.io/ISS/iss-explorer.html).

[16] R. Baumann, S. Doutre, J.-G. Mailly, J. P. Wallner, Enforcement in Formal Argumentation, in: D. Gabbay, M. Giacomin, G. R. Simari, M. Thimm (Eds.), Handbook of Formal Argumentation, volume 2, College Publications, 2021.

[17] J.-G. Mailly, Yes, No, Maybe, I Don't Know: Complexity and Application of Abstract Argumentation with Incomplete Knowledge, Argument & Computation 13 (2022) 291–324.

[18] G. Alfano, S. Greco, F. Parisi, I. Trubitsyna, Totality, Determinism and Functionality Problems in

(Incomplete) Abstract Argumentation Frameworks, Journal of Logic and Computation 36 (2026) exag025.
[19] G. Alfano, S. Greco, F. Parisi, I. Trubitsyna, Counterfactual and Semifactual Explanations in Abstract Argumentation: Formal Foundations, Complexity and Computation, in: Proc. 21st Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2024), 2024, pp. 14–26.
[20] S. Liu, M. Shao, B. Liao, Beyond But-for Test: Counterfactual Explanation in Abstract Argumentation via Actual Causality, in: Computational Models of Argument (COMMA 2026), IOS Press, 2026. To appear.
[21] R. Baumann, S. Woltran, The Role of Self-Attacking Arguments in Characterizations of Equivalence Notions, Journal of Logic and Computation 26 (2016) 1293–1313.
[22] C. Cayrol, F. Dupin de Saint-Cyr, M.-C. Lagasquie-Schiex, Change in Abstract Argumentation Frameworks: Adding an Argument, Journal of Artificial Intelligence Research 38 (2010) 49–84.
[23] T. Rienstra, C. Sakama, L. v. d. Torre, Persistence and Monotony Properties of Argumentation Semantics, in: Theory and Applications of Formal Argumentation (TAFA 2015), volume 9524 of *LNCS*, Springer, 2015, pp. 211–225.
[24] E. Oikarinen, S. Woltran, Characterizing Strong Equivalence for Argumentation Frameworks, Artificial Intelligence 175 (2011) 1985–2009.
[25] R. Baumann, W. Dvořák, T. Linsbichler, S. Woltran, A General Notion of Equivalence for Abstract Argumentation, Artificial Intelligence 275 (2019) 379–410.
[26] S. Nofal, K. Atkinson, P. E. Dunne, Computing Grounded Extensions of Abstract Argumentation Frameworks, The Computer Journal 64 (2021) 54–63.
[27] G. Alfano, S. Greco, F. Parisi, Efficient Computation of Extensions for Dynamic Abstract Argumentation Frameworks: An Incremental Approach, in: Proc. 26th Intl. Joint Conf. on Artificial Intelligence (IJCAI), 2017, pp. 49–55.
[28] T. J. M. Bench-Capon, Representation of Case Law as an Argumentation Framework, in: Legal Knowledge and Information Systems (JURIX 2002), IOS Press, 2002, pp. 103–112.
[29] M. Ulbricht, R. Baumann, If Nothing Is Accepted – Repairing Argumentation Frameworks, Journal of Artificial Intelligence Research 66 (2019) 1099–1145.