Argument Decomposition in Abstract Dialectical Frameworks

Johannes E. Bendler[1][0009−0002−3568−5145] and Tjitze Rienstra[2][0000−0001−9971−3067]

[1] Department of Computer Science, Faculty of Sciences, Vrije Universiteit Amsterdam, The Netherlands
j.e.bendler@vu.nl
[2] Department of Advanced Computing Sciences (DACS), Faculty of Science and Engineering, Maastricht University, The Netherlands
t.rienstra@maastrichtuniversity.nl

Abstract. Abstract Dialectical Frameworks (ADFs) represent a powerful generalization for various argumentation frameworks and non-monotonic reasoning formalisms, wherein propositional acceptance conditions determine the acceptability of statements. This research introduces three decomposition operations addressing logical negation, conjunction, and disjunction, which systematically deconstruct complex acceptance conditions into simpler components while preserving the framework's extensions under standard semantics. For each operation, rigorous proofs are provided, demonstrating that the transformed ADF produces identical extensions under admissible, preferred, complete, and grounded semantics for all statements of the original framework. Through iterative application of these decomposition operations, any ADF can be transformed into a maximal decomposed ADF with minimal acceptance conditions, thereby enhancing explainability and modularity. Furthermore, these maximal decomposed ADFs exhibit the properties of Bipolar ADFs, thus potentially yielding computational advantages and better translatability into other formalisms such as Boolean Networks. This decomposition approach establishes a formal foundation for structural optimization, as well as improved explainability and transparency in argumentative reasoning systems.

Keywords: Argumentation Theory · Knowledge Representation · Non-monotonic reasoning · Explainable AI · Abstract Dialectical Frameworks (ADFs).

1 Introduction

In artificial intelligence and knowledge representation, argumentation theory provides a robust framework for reasoning with incomplete or inconsistent information. Dung's Abstract Argumentation Frameworks (AFs) [19], are foundational in this field, where arguments are represented as abstract entities connected by binary attack relations between them. While laying the groundwork for many subsequent argumentation systems, Dung's AFs are often criticized due to their limitation in modeling real-world argumentation scenarios. This lead to the development of several additions and extensions of the framework, such as support relations [17] [29], preferences and priorities [2] [3], or weighted [27] and joined attacks [28] (see [12] for an overview).

To further address the limitation and to unify many of the aforementioned additions, Brewka et al. introduced Abstract Dialectical Frameworks (ADFs), which present a generalization of Dung's AFs [13] [11]. In ADFs, arguments (now called statements) are assigned a propositional acceptance condition that determines when the statement is accepted, according to their dependencies on other statements. Furthermore, ADFs evaluate these conditions using a three-valued logic, assigning true (**t**), false (**f**) or undecided (**u**). The authors describe ADFs not as a

tool for knowledge representation, but rather as "argumentation middleware" that can be used to translate between many of the previously mentioned formalisms.

Argumentation often presents a dynamic process, where arguments and relations might change as new information arises. Especially for AFs, significant research has explored the mechanics and effects of these dynamics, such as adding new arguments [16], techniques to enforce the acceptance of specific arguments [9] [8], and the application of belief revision operators to argumentation frameworks [18] [6]. However, there exists another category of structural modification that receives considerably less attention: Transformations that preserve the resulting sets of acceptable statements while also altering the structure of the framework. These modifications are not redundant; they can lead to improved computational performance, as well as contribute to better readability and understandability of the argumentation framework.

This research addresses this gap by focusing on a specific result-preserving operation in ADFs: The *decomposition of arguments* and their acceptance conditions. The introduced decomposition operators split complex acceptance conditions into a set of simpler ones, which are then assigned to newly introduced auxiliary statements. Specifically, this research aims to answer the questions on how to define such operators for different types of acceptance conditions and how to ensure the preservation of extensions from the original to the decomposed ADF. In particular, admissible, preferred, complete, and grounded semantics are examined in this regard.

The paper is organized as follows: Section 2 reviews existing literature on ADFs and structural change in argumentation frameworks. Section 3 establishes the foundational concepts of ADFs like properties, interpretations, and semantics. Section 4 introduces the decomposition operations, providing formal definitions. Subsequently, Section 5 discusses the properties, results, and implications of the decomposition operators and compares them with existing approaches to decomposition. Section 6 summarizes the key findings and provides an outlook for further research.

2 Previous Work

In their paper introducing ADFs, Brewka et al. already show the possibility to translate between Dung-style AFs and ADFs. Subsequent research explored how ADFs can partially or fully capture other frameworks and approaches to argumentation and non-monotonic reasoning. Brewka et al. also showed how ADFs can represent normal logic programs, where nodes of the framework correspond to the atoms in the program [13]. In a similar manner, Hannes Strass expands this translation, by showing that ADFs can faithfully be translated into normal logic programs, and by proving that the semantics are preserved [30]. Alexander Bochman shows the close conceptual ties between ADFs and causal reasoning [10]. He argues that accepting a statement based on its acceptance condition is the same as the acceptance condition causing the statement to be true. Heyninck et al. uncover a striking similarity between ADFs and Boolean Networks (BNs) [23], which are frequently used in biological contexts (see for example [31]), by highlighting their similar use of boolean formulas to establish relationships. While BNs can be fully captured within an ADF, the reverse is not directly true, as relations between statements in ADFs do not have to be purely positive (supporting) or negative (attacking), as is the case in BNs. However, the subclass of Bipolar ADFs (BADFs) (see [13]) captures this relationship type fully and is thus equivalent.

Brewka et al., as well as Stefan Ellmauthaler, show that solving many of the problems in ADFs, like verifying whether a specific set of statements is a valid extension, is computationally more complex than in Dung-style frameworks and at least NP-hard [13] [11] [20]. The subclass of BADFs exhibits favorable complexity while still offering more modeling capabilities [7]. For example, verifying admissible semantics for general ADFs is coNP-complete, but for BADFs it becomes computable in polynomial time.

As mentioned in the introduction, structural change in abstract argument frameworks has always been an active research area. Beyond the ones presented previously, more elaborate operations on argumentation frameworks in general and ADFs specifically have been developed, especially regarding the separation of parts of the framework. In the domain of Dung-style AFs, Baroni et al. apply the notion of *strongly connected components (SCC)* from graph theory to AFs, in order to determine the behavior of the components [5]. Liao et al. pursue a similar strategy to splitting AFs, focusing on the dynamic aspects of the framework, like changing arguments or relations [25]. They divide the AF into three parts: the unaffected, the affected, and the conditioning part, of which the latter two can be abstracted away into a simpler *Conditional Argumentation Framework*. In the realm of ADFs, Thomas Linsbichler developed a notion of splitting an ADF, such that components can be evaluated separately [26]. He presents *directional splitting*, in which the ADF is partitioned such that links are only directed from one partition into the other. Gaggl and Strass adapted the concept of SCCs to ADFs [22]. Their approach, also termed *decomposition* affects the ADF as a whole, allowing a novel way of computing interpretations for the common semantics.

While these methods focus on partitioning an entire framework, our work on decomposition is closer to Baroni et al.'s research on *the Input/Output behavior of Argumentation Frameworks* [4]. In it, they introduce local partitions of an AF into so-called modules, which are defined by their respective subset of statements, their relations to statements outside of the set, and the truth labelings for the latter. By determining the acceptability of the statements inside, modules can be described by their relation to outgoing statements and can be treated like a single "virtual node", regardless of their internal structure (similar to [5]). Baroni et al. relate this abstraction process to the act of summarization, in which a complex part is summarized and therefore replaced by a simpler and more synthetic representation. The reverse operation can be equated to an explanation, in which the details and inner workings of a statement are laid out. Our process of decomposition is comparable to such a reverse operation.

3 Preliminaries

In this section the basics concepts and notations on ADFs are defined as introduced by Brewka et al. in [11].

Definition 1. *An* abstract dialectical framework *(in short, ADF) is a tuple* $D = (S, L, C)$ *where*

- S *is a set of statements*
- $L \subseteq S \times S$ *is a set of links*
- $C = \{C_s\}_{s \in S}$ *is a set of total functions* $C_s : 2^{par(s)} \mapsto \{\boldsymbol{t}, \boldsymbol{f}\}$*, one for each statement* s*, where* $par(s)$ *represents the set of parents of* s*.* C_s *is called the acceptance condition of* s*.*

It is common practice to represent the acceptance condition C as a collection $C = \{\varphi_s\}_{s \in S}$, where φ_s corresponds to a propositional formula for a statement s. Furthermore, Brewka et al. suggest omitting the set of links L from the defining tuple, as links between statements are given implicitly through the acceptance conditions [13] [11]. In this research, this reduced tuple, as well as the propositional format for acceptance conditions are used. The range of possible propositional formulas will be restricted to the set of formulas using the logic connectives of negation ($\neg$), conjunction ($\wedge$), and disjunction ($\vee$), as any propositional formula can be expressed through these three connectives.

Definition 2. *A* three-valued interpretation v *of an ADF* $D = (S, C)$ *is a mapping* $v : S \mapsto \{\boldsymbol{t}, \boldsymbol{f}, \boldsymbol{u}\}$.

- $v(s)$ denotes the truth value of $s \in S$.
- $v(C_s)$ denotes the truth value of the acceptance condition C_s. Propositional formulas are evaluated according to Kleene's strong three-valued logic [24, p. 334].
- $v{\downarrow_B}$ denotes the restriction of v to B, with $B \subseteq S$, that is, $v{\downarrow_B}$ only includes mappings for arguments in B.

Definition 3. *The three truth values* $\{\boldsymbol{t}, \boldsymbol{f}, \boldsymbol{u}\}$ *are partially ordered by their* information content $\leq_i$*, such that* $\boldsymbol{u} \leq_i \boldsymbol{t}$ *and* $\boldsymbol{u} \leq_i \boldsymbol{f}$*. Thus the pair* $(\{\boldsymbol{t}, \boldsymbol{f}, \boldsymbol{u}\}, \leq_i)$ *forms a complete meet-semilattice with the meet operation* $\sqcap$*. This meet can be read as* consensus *and assigns* $\boldsymbol{t} \sqcap \boldsymbol{t} = \boldsymbol{t}$, $\boldsymbol{f} \sqcap \boldsymbol{f} = \boldsymbol{f}$ *and* $\boldsymbol{u}$ *otherwise.*

The information ordering $\leq_i$ *extends to interpretations* v_1, v_2 *of* S *in that* $v_1 \leq_i v_2$ *iff* $\forall s \in S : v_1(s) \leq_i v_2(s)$.

Definition 4. *For a three-valued interpretation* v*, an interpretation* w extends v *iff* $v \leq_i w$*. This means that all statements mapped to* $\boldsymbol{t}$ / $\boldsymbol{f}$ *by* v *are mapped to* $\boldsymbol{t}$ / $\boldsymbol{f}$ *by* w*. Futhermore, if all statements mapped to* $\boldsymbol{u}$ *by* v *are mapped to* $\boldsymbol{t}$ / $\boldsymbol{f}$ *by* w*, then* w *is a* two-valued extension*. The set of all two-valued extensions of* v *is denoted by* $[v]_2$.

To determine the set of acceptable arguments of an ADF under different semantics, Brewka et al. introduce the Γ-operator, which allows for a fixed-point-based characterization of the semantics. When applied to a statement, the operator takes every possible valuation of the acceptance condition into account, given that some of the parents values are still unknown. Through the relationship between an interpretation v and its derived $\Gamma_D(v)$, the semantics are defined, determining or justifying a coherent set of arguments.

Definition 5. *The operator* Γ_D *over a three-valued interpretation* v *of an ADF* $D = (S, C)$ *revises an interpretation* v *by the mapping:*

$$s \mapsto \bigsqcap \{w(C_s) \mid w \in [v]_2\}$$

- $\Gamma_D(v)$ denotes the revised three-valued interpretation of v.
- $\Gamma_D(v)(s)$ denotes the revised truth value of the statement s after applying the operator Γ_D to v.

Definition 6. *A three-valued interpretation* v *of an ADF* $D = (S, C)$ *is:*

- admissible *iff* $v \leq_i \Gamma_D(v)$*, denoted by* $v \in ad(D)$.
- preferred *iff* v *is* $\leq_i$*-maximal admissible, denoted by* $v \in pr(D)$.
- complete *iff* $v = \Gamma_D(v)$*, denoted by* $v \in co(D)$.
- grounded *iff* v *is* $\leq_i$*-minimal complete, denoted by* $v \in gr(D)$.

These semantics correspond to their counterparts in Dung-style AFs, such that they yield the same results in both formalisms. They capture different approaches to determining the set of accepted statements, in that admissible semantics represent a credulous approach, while grounded semantics are more skeptical.

While stable semantics represents another important approach in abstract argumentation theory, we exclude it from this research due to its fundamentally different computational paradigm. Unlike the aforementioned semantics, stable semantics operate on two-valued interpretations through framework reduction rather than three-valued interpretations with the Γ-operator.

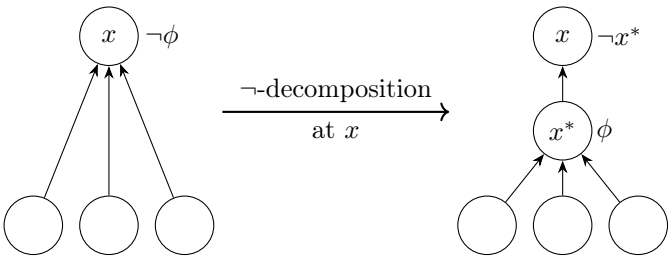
4 Decomposition

In this chapter, we will formally define decomposition operations that can be applied to specific statements in an ADF.

4.1 Decomposition Operations

The operation of *negation decomposition* can be applied to any statement with an acceptance condition of the form $C_s = \neg\phi$, where ϕ represents some propositional formula. This operation separates the negation connective ($\neg$) from the rest of the formula ϕ by introducing a new auxiliary statement x^* that adopts ϕ as its acceptance condition.

Definition 7. *Let $D = (S, C)$ be an ADF. Let $x \in S$ with $C_x = \neg\phi$, where ϕ is some propositional statement. Then the* ¬-decomposition at x *of D is the ADF $D' = (S', C')$, where $S' = S \cup \{x^*\}$ and C' is defined by $C'_y = C_y$ for all $y \in S \setminus \{x, x^*\}$, $C'_x = \neg x^*$ and $C'_{x^*} = \phi$.*

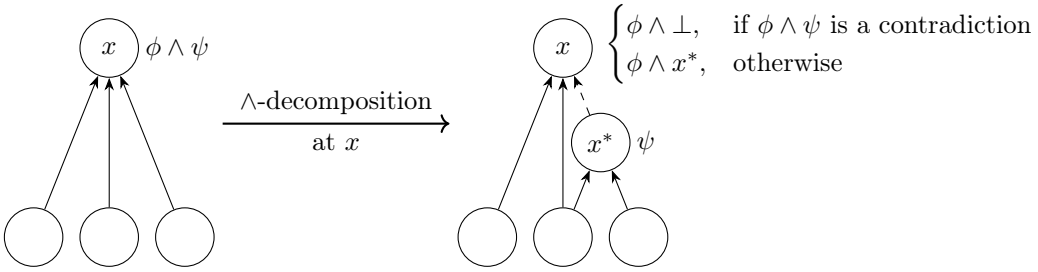


Applying this negation decomposition operations yields various benefits by simplifying the acceptance condition within the framework. It allows the analysis and processing of the relationships between statements, regardless of the attacking characteristic. Thus it effectively separates the content of an argument from its positive or negative relation with other arguments. Furthermore, removing the leading negation of a more complex propositional formula ϕ enables the further application of other decomposition operations, that previously couldn't be applied to this statement.

The second operation, termed *conjunction decomposition*, applies to any statement with an acceptance condition of the form $C_s = \phi \wedge \psi$. It preserves one conjunctive component (ϕ) within the original statement, while extracting the remaining component (ψ) into a newly introduced auxiliary statement x^*. This effectively decouples the conjunctive parts of the original condition, but a parent of the original statement may appear simultaneously in both ϕ and ψ, thereby becoming a parent to both the original and auxiliary statement. Formally:

Definition 8. *Let $D = (S, C)$ be an ADF. Let $x \in S$ with $C_x = \phi \wedge \psi$, where ϕ and ψ are some propositional statements. Then the* ∧-decomposition at x *of D is the ADF $D' = (S', C')$, where $S' = S \cup \{x^*\}$ and C' is defined by:*

- $C'_y = C_y$ *for all* $y \in S \setminus \{x, x^*\}$
- $C'_x = \begin{cases} \phi \wedge \bot, & \text{if } \phi \wedge \psi \text{ is a contradiction} \\ \phi \wedge x^*, & \text{otherwise} \end{cases}$
- $C'_{x^*} = \psi$



The revised C'_x is conditioned on whether or not $\phi \wedge \psi$ present a logical contradiction (for example if $\psi = \neg\phi$). This conditioning in the definition of the new C'_x is necessary, in order to preserve the semantics in all cases. A naive breakdown into $\phi \wedge x^*$ in all cases would allow for a undecided (**u**) outcome, where it should be false. Section 5 and Appendix A elaborate on this conditioning.

The presented decomposition operation only removes one sub-formula (ψ) from the acceptance condition. A decomposition removing both parts ϕ and ψ into a new auxiliary statement each, is called a *full decomposition*. This is not introduced as a separate concept or definition, as it can be achieved by applying the conjunction decomposition twice on the same statement, once extracting ϕ to x^* and once extracting ψ to x^{**}. The order of execution does not alter the structure and can thus be done arbitrarily. This full decomposition operation can therefore be applied to conjunctions of any length such as $\phi \wedge \psi \wedge \mu$. A definition of full conjunction decomposition, as well as the proof showing that its equal to applying decomposition twice can be found in Appendix B.

The third operation is called *disjunction decomposition* and applies to any statement with an acceptance condition of the form $C_s = \phi \vee \psi$. It functions similarly to conjunction decomposition, but is condition on $\phi \vee \psi$ being a logical tautology, instead of a contradiction.

Definition 9. *Let $D = (S, C)$ be an ADF. Let $x \in S$ with $C_x = \phi \vee \psi$, where ϕ and ψ are some propositional statements. Then the* ∨-decomposition *at x of D is the ADF $D' = (S', C')$, where $S' = S \cup \{x^*\}$ and C' is defined by:*

- $C'_y = C_y$ *for all* $y \in S \setminus \{x, x^*\}$
- $C'_x = \begin{cases} \phi \vee \top, & \text{if } \phi \wedge \psi \text{ is a tautology} \\ \phi \vee x^*, & \text{otherwise} \end{cases}$
- $C'_{x^*} = \psi$

4.2 Preservation of Semantics

Given these definitions, it needs to be shown that the operations of decomposition don't change the resulting extensions of the ADF. This is done by proving that the set of admissible, preferred, complete and grounded interpretation stays the same between the original and decomposed framework. As it is the case that the decomposed framework includes additional auxiliary statements, which are not accounted for in interpretation of the original framework, only the value assignments of statements $s \in S$ of the original framework are considered.

Theorem 1. *Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its $\circ$-decomposition at x for $\circ \in \{\neg, \wedge, \vee\}$. For any semantics $\sigma \in \{ad, pr, co, gr\}$: $\sigma(D) = \sigma(D')\downarrow_S$.*

The restriction of a set $\sigma(D')\downarrow_S$ to S denotes that all interpretations in this set are restricted to statements in S, as introduced in Definition 2. Because the proofs for all three decomposition

operations and all relevant semantics are out of scope of this paper, they can be found in Appendix A. In this appendix, several lemmas are introduced with the aim of reducing repeating steps in the proofs and improving the structure.

4.3 Maximal Decomposed ADFs

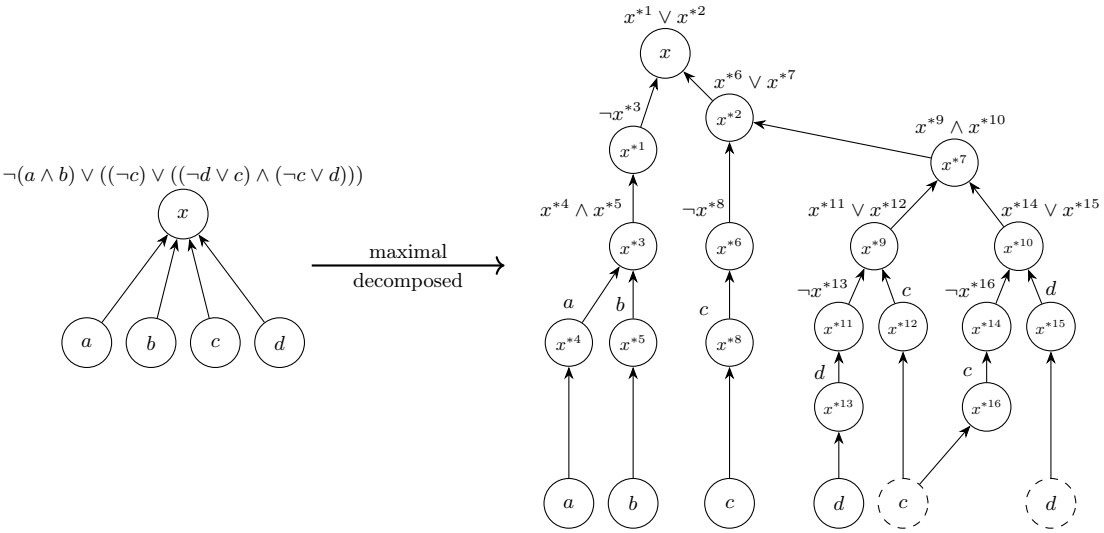
In propositional logic, well-formed formulas (WFFs) are constructed through a recursive process: Atomic propositions serve as foundational elements, while complex formulas are built by applying logical connectives to simpler formulas. Herbert B. Enderton, in his book "A mathematical introduction to logic" [21], defines equations to build such WFFs, the main three being:

$$\mathcal{E}_{\neg}(\alpha) = (\neg\alpha)$$
$$\mathcal{E}_{\wedge}(\alpha, \beta) = (\alpha \wedge \beta)$$
$$\mathcal{E}_{\vee}(\alpha, \beta) = (\alpha \vee \beta)$$

It becomes apparent that the decomposition operations presented in our research precisely reverse this construction process. Each operation systematically deconstructs a propositional acceptance condition by isolating the logical connective from its components. One may, for example, characterize *negation decomposition* as a reverse WFF equation of the form $\mathcal{E}_{\neg}^{-1}(\neg\alpha) = (\alpha)$ or *full conjunction decomposition* as $\mathcal{E}_{\wedge}^{-1}(\alpha \wedge \beta) = (\alpha, \beta)$. Through iterative application of these decomposition operations, any arbitrary acceptance condition can be reduced until only atomic components remain, each represented by a distinct statement in the framework. This process of repeated decomposition leads to the concept of a *maximal decomposed ADF*.

Definition 10. *Let $D = (S, C)$ be an ADF. Then the ADF $D' = (S', C')$ is* maximal decomposed *if for all $s \in S$ the acceptance condition C_s satisfies one of the following conditions:*

- $C_s = a$ *where* $a \in S$
- $C_s = \neg a$ *where* $a \in S' \setminus S$
- $C_s = a \wedge b$ *where* $a, b \in S' \setminus S$
- $C_s = a \vee b$ *where* $a, b \in S' \setminus S$



As every WFF can be decomposed with the presented operations into atom, and as every acceptance condition in an ADF is a WFF, it follows that every acceptance condition can be decomposed into atoms and every ADF can be transformed into a *maximal decomposed ADF*.

5 Discussion

The decomposition operations presented in the preceding chapter necessitate the introduction of new auxiliary statements into the framework. Consequently, any interpretation of the original framework must now account for the truth value assignment of these newly introduced statements. The auxiliary statement is supposed to mirror the behavior of the sub-formula it replaces, in order to preserver the interpretations of the original framework. However, it is not enough to just assign the value of the formula (like $v'(x^*) = v'(\psi)$ for example), as this can lead to inconsistencies when the original formula presents a logical contradiction or tautology. Therefore, as it follows from the proofs in Appendix A, the assignment has to be $v'(x^*) = \bigsqcap\{w(C'_{x^*}) \mid w \in [v']_2\}$.

Another notable special case of decomposition occurs when the statement being decomposed has itself as a parent, creating a reflexive dependency. As per the definition of decomposition, the auxiliary statements inherit the parents of the original statement according to their respective formula component. In the reflexive case, this process effectively transforms the self-relation into connections between the original and auxiliary statement, thus also representing a tool for eliminating such reflexive relations from a given statement. This does not mean the circular reasoning chain is interrupted, but just that it is extracted into multiple separate statements.

The presented decomposition operations naturally suggest an inverse operation for statement *composition*, wherein multiple simpler statements are consolidated, having their acceptance conditions appropriately connected. The direct inverse of the presented operations present specific structural requirements for the structure of the statements. For example, a definition of a reverse operation for conjunction decomposition might be:

Definition. *Let $D = (S, C)$ be an ADF. Let $x, x^* \in S$ with $C_x = \phi \wedge x^*$ and $C_{x^*} = \psi$, where ϕ and ψ are some propositional statements. Then the* conjunction composition *at x of D is the ADF $D' = (S', C')$, where $S' = S \setminus \{x^*\}$ and C' is defined by $C'_y = C_y$ if $y \neq x$ and $C'_x = \phi \wedge \psi$.*

Decomposition exhibits striking similarities to Baroni et al.'s work "On the Input/Output behavior of argumentation frameworks" [4], particularly the concept of an argumentation multipoles; modular component defined by their interface of input and output relations. This parallels decomposition operations for ADFs, as each original statement can be conceptualized as such a multipole, consisting of a, potentially more complex, internal structure. This principle also applies to the *transparency* of semantics, where the replacement of a multipole with an abstract alternative does not alter the remainder of the framework.

The specific structure of maximal decomposed frameworks aligns particularly well with the concept of dialogue theory or discussion games, wherein a proponent and opponent exchange arguments in order to reach some conclusion. The relationship between such dialogical approaches and abstract argumentation theory has been established multiple times in the literature, predominantly in the context of expressing semantics such as grounded and preferred through discussion-based approaches [15] [14]. Each step of the discussion constitutes an explanation for an original statement, which is not immediately accepted by the opponent. To explain and justify that statement, decomposition operations introduce progressive reasoning-sub-steps, ultimately resolving into fundamental axioms or empirical facts.

A practical illustration of how decomposition enhances explainability can be given in the legal domain with models such as ANGELIC [1], where leaf nodes capture facts and evidence, while

intermediate statements express legal concepts and rules that combine and relate these facts. The terminal statements then correspond to legal outcomes derived from the joint acceptance of relevant conditions. By applying decomposition in this context, a complex condition combining the aforementioned legal elements can be decomposed to show how each piece of evidence influences the applicability and result of each related legal concept or law. Through decomposition the legal reasoning process becomes more explainable by making individual relations more explicit, while preserving the outcome of the legal case.

Another structural characteristic of a maximal decomposed ADF is that the form of its acceptance conditions classifies it as a Bipolar ADF. Within such bipolar frameworks, any parent statement is either directly supporting or attacking. Specifically, only the negation decomposition forms an attacking relation between statements. Since the presented decomposition operations are applicable to any arbitrary ADF, and every ADF is maximal decomposable, it follows that any ADF can be systematically transformed into a BADF through our decomposition operations. Moreover, the resulting maximal decomposed BADF exhibits a highly specific structure, where each statement has at most two (auxiliary) parents. Consequently, this also yields an acceptance condition in the normal form (either conjunctive or disjunctive) for every condition. A fully decomposed ADF therefore represents a specialized subclass of BADFs that also preserves all semantic characteristics of the original framework, with respect to the original statements. Even without maximally decomposing the ADF, applying the operations can deconstruct any acceptance condition until it is in a normal form.

The conclusion that any ADF can be transformed into a BADF aligns with the findings of Stefan Ellmauthaler [20]. In his research, Ellmauthaler similarly employs auxiliary statements to achieve this transformation, but rather than decomposing logic connectives within the acceptance conditions, he focuses exclusively on formulas with dependent variables, inhibiting both attacking and supporting characteristics. His algorithm iteratively resolves dependencies by replacing any condition C_s that relies on such a variable α with a disjunction of two new auxiliary statements, which represent the cases where α is true and false, respectively. Although Ellmauthaler's approach shares structural similarities with decomposition by extracting components that cause non-bipolarity into separate statements, it increases the complexity of some acceptance conditions through the additional conjunction with the dependent variables α and $\neg\alpha$. Conversely, our decomposition operations focus on simplifying propositional formulas and decoupling components from the logical connectives, with the removal of dependent variables occurring merely as a byproduct. In his work, Ellmauthaler further demonstrates that BADFs exhibit superior computational efficiency compared to non-bipolar ADFs. Consequently, the decomposition operations presented in this research not only enhance transparency and explainability in argumentation, but also yield computational advantages through transformation into maximal decomposed frameworks.

6 Conclusion and Future Work

This research addresses structural modification of Abstract Dialectical Frameworks through decomposition operations, while preserving the interpretations and semantics of the framework. Three distinct operations are introduced: Negation, conjunction, and disjunction decomposition. Each operation systematically separates a sub-formula of the acceptance condition from the other components and assigns it to a newly introduced auxiliary statements. This set of decomposition operations covers any arbitrary acceptance condition and is thus applicable to any ADF. Subsequently, the preservation of admissible, preferred, complete, and grounded interpretations from the original to the decomposed framework were shown. This consistency ensures that despite structural modifications, the reasoning outcomes of these frameworks are preserved.

Furthermore, the concept of maximal decomposed ADFs was introduced, wherein all acceptance conditions are reduced to either atomic propositional statements or simple logical connections between auxiliary statements. The fully decomposed structure represents BADFs, which exhibit improved computational efficiency and a one-to-one correspondence to Boolean Networks. Furthermore, the decomposition operations were compared to previous approaches to modularize ADFs, such as Baroni et al.'s concept of multipoles [4] or Ellmauthaler's resolution of dependent variables [20].

Future work could concentrate on several promising directions, such as focusing on the reverse operations to develop more explicit composition operations with less restrictive requirements. Additionally, extending the preservation result to other semantics like stable, semi-stable, or ideal semantics can show how decomposition preserves more than just the common semantics. Exploring the connections to multipoles and modularization, as well as similar approaches, can be enhance understanding of modular argumentation structures and dynamical replacement of subframeworks. Furthermore, the connection between (maximal) decomposed ADFs and dialogue theory could be formalized in a dialogue game based on decomposition operations. This might involve defining permissible moves, termination conditions, or finding winning strategies, but might also be used for retrieving first principles from given statements and conditions. Such dialogue mechanics can also provide step-by-step explainability, not only for single statements or conditions, but for ADFs as a whole. Lastly, the principles of decomposition could be applied to certain modified ADFs, such as weighted or probabilistic argumentation structures, potentially allowing for the deconstruction of more sophisticated and nuanced argumentation structures that incorporate degrees of uncertainty or preference.

References

1. Latifa Al-Abdulkarim, Katie Atkinson, and Trevor Bench-Capon. A methodology for designing systems to reason with legal cases using Abstract Dialectical Frameworks. *Artificial Intelligence and Law*, 24(1):1–49, March 2016.
2. Leila Amgoud and Claudette Cayrol. A Reasoning Model Based on the Production of Acceptable Arguments. *Annals of Mathematics and Artificial Intelligence*, 34(1/3):197–215, 2002.
3. Leila Amgoud and Srdjan Vesic. A new approach for preference-based argumentation frameworks. *Annals of Mathematics and Artificial Intelligence*, 63(2):149–183, October 2011.
4. Pietro Baroni, Guido Boella, Federico Cerutti, Massimiliano Giacomin, Leendert Van Der Torre, and Serena Villata. On the Input/Output behavior of argumentation frameworks. *Artificial Intelligence*, 217:144–197, December 2014.
5. Pietro Baroni, Massimiliano Giacomin, and Giovanni Guida. SCC-recursiveness: a general schema for argumentation semantics. *Artificial Intelligence*, 168(1-2):162–210, October 2005.
6. Ringo Baumann and Gerhard Brewka. AGM meets abstract argumentation: expansion and revision for dung frameworks. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 2734–2740, Buenos Aires, Argentina, 2015. AAAI Press.
7. Ringo Baumann and Maximilian Heinrich. Bipolar Abstract Dialectical Frameworks Are Covered by Kleene's Three-valued Logic. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 3123–3131, Macau, SAR China, August 2023. International Joint Conferences on Artificial Intelligence Organization.
8. Baumann Ringo. What Does it Take to Enforce an Argument? Minimal Change in abstract Argumentation. In *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2012.
9. Baumann Ringo and Brewka Gerhard. Expanding Argumentation Frameworks: Enforcing and Monotonicity Results. In *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2010.
10. Bochman Alexander. Abstract Dialectical Argumentation Among Close Relatives. In *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2016.

11. Gerd Brewka, Stefan Ellmauthaler, Hannes Strass, Johannes Peter Wallner, and Stefan Woltran. Abstract Dialectical Frameworks Revisited. In *International Joint Conference on Artificial Intelligence*, pages 803–809. AAAI Press, 2013. Accepted: 2022-08-04T11:59:06Z.
12. Gerhard Brewka, Sylwia Polberg, and Stefan Woltran. Generalizations of Dung Frameworks and Their Role in Formal Argumentation. *IEEE Intelligent Systems*, 29(1):30–38, January 2014.
13. Gerhard Brewka and Stefan Woltran. Abstract dialectical frameworks. In *Proceedings of the Twelfth International Conference on Principles of Knowledge Representation and Reasoning*, KR'10, pages 102–111, Toronto, Ontario, Canada, May 2010. AAAI Press.
14. Martin W. A. Caminada, Wolfgang Dvořák, and Srdjan Vesic. Preferred semantics as socratic discussion. *Journal of Logic and Computation*, 26(4):1257–1292, August 2016.
15. Caminada Martin and Podlaszewski Mikołaj. Grounded Semantics as Persuasion Dialogue. In *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2012.
16. C. Cayrol, F. Dupin De Saint-Cyr, and M. Lagasquie-Schiex. Change in Abstract Argumentation Frameworks: Adding an Argument. *Journal of Artificial Intelligence Research*, 38:49–84, May 2010.
17. C. Cayrol and M. C. Lagasquie-Schiex. On the Acceptability of Arguments in Bipolar Argumentation Frameworks. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, and Lluís Godo, editors, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, volume 3571, pages 378–389. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. Series Title: Lecture Notes in Computer Science.
18. Sylvie Coste-Marquis, Sébastien Konieczny, Jean-Guy Mailly, and Pierre Marquis. A Translation-Based Approach for Revision of Argumentation Frameworks. In Eduardo Fermé and João Leite, editors, *Logics in Artificial Intelligence*, volume 8761, pages 397–411. Springer International Publishing, Cham, 2014. Series Title: Lecture Notes in Computer Science.
19. Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, September 1995.
20. Stefan Ellmauthaler. *Abstract Dialectical Frameworks : properties, complexity, and implementation.* Thesis, Technische Universität Wien, 2012. Accepted: 2020-06-30T15:58:01Z.
21. Herbert B. Enderton. *A mathematical introduction to logic.* Harcourt/Academic Press, San Diego, 2. ed.; transf. to digital print edition, 2010.
22. Gaggl Sarah Alice and Strass Hannes. Decomposing Abstract Dialectical Frameworks. In *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2014.
23. Jesse Heyninck, Matthias Knorr, and João Leite. Abstract Dialectical Frameworks are Boolean Networks. In Carmine Dodaro, Gopal Gupta, and Maria Vanina Martinez, editors, *Logic Programming and Nonmonotonic Reasoning*, volume 15245, pages 98–111. Springer Nature Switzerland, Cham, 2025. Series Title: Lecture Notes in Computer Science.
24. Stephen Cole Kleene. *Introduction to Metamathematics.* P. Noordhoff N.V., Groningen, 1952.
25. Beishui Liao, Li Jin, and Robert C. Koons. Dynamics of argumentation systems: A division-based method. *Artificial Intelligence*, 175(11):1790–1814, July 2011.
26. Linsbichler Thomas. Splitting Abstract Dialectical Frameworks. In *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2014.
27. Diego C. Martínez, Alejandro J. García, and Guillermo R. Simari. An abstract argumentation framework with varied-strength attacks. In *Proceedings of the Eleventh International Conference on Principles of Knowledge Representation and Reasoning*, KR'08, pages 135–143, Sydney, Australia, September 2008. AAAI Press.
28. Søren Holbech Nielsen and Simon Parsons. A Generalization of Dung's Abstract Framework for Argumentation: Arguing with Sets of Attacking Arguments. In Nicolas Maudet, Simon Parsons, and Iyad Rahwan, editors, *Argumentation in Multi-Agent Systems*, volume 4766, pages 54–73. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. Series Title: Lecture Notes in Computer Science.
29. Nir Oren and Timothy J. Norman. Semantics for Evidence-Based Argumentation. In *Proceedings of the 2008 conference on Computational Models of Argument: Proceedings of COMMA 2008*, pages 276–284, NLD, June 2008. IOS Press.

30. Hannes Strass. Approximating operators and semantics for abstract dialectical frameworks. *Artificial Intelligence*, 205:39–70, December 2013.
31. Rui-Sheng Wang, Assieh Saadatpour, and Réka Albert. Boolean modeling in systems biology: an overview of methodology and applications. *Physical Biology*, 9(5):055001, October 2012.

Appendix A: Proofs

Negation Decomposition

Lemma 1. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its *¬-decomposition at* x. Let v be a three-valued interpretation of D. Then $v \in ad(D) \implies \exists v' \in ad(D') : v'{\downarrow_S} = v$.

Proof. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its *¬-decomposition at* x. Let v be a three-valued interpretation of D and suppose this v is admissible in D. We construct an interpretation v' for D' such that $\forall s \in S : v'(s) = v(s)$ and $v'(x^*) = \bigsqcap\{w(C'_{x^*}) \mid w \in [v']_2\}$, thus fulfilling the restriction that $v'{\downarrow_S} = v$.

Now we show that this v' is admissible in D':

$$
\begin{aligned}
v \in ad(D) &\implies \forall s \in S : v(s) \leq_i \Gamma_D(v)(s) && (1)\\
&\implies \forall s \in S : v(s) \leq_i \bigsqcap\{w(C_s) \mid w \in [v]_2\} && (2)\\
&\implies \forall s \in S : v'(s) \leq_i \bigsqcap\{w(C_s) \mid w \in [v']_2\} && (3)\\
&\implies \forall s \in S : v'(s) \leq_i \bigsqcap\{w(C'_s) \mid w \in [v']_2\} && (4)\\
&\implies \forall s' \in S' : v'(s') \leq_i \bigsqcap\{w(C'_{s'}) \mid w \in [v']_2\} && (5)\\
&\implies \forall s' \in S' : v'(s') \leq_i \Gamma_{D'}(v')(s') && (6)\\
&\implies v' \in ad(D') && (7)
\end{aligned}
$$

(1) Follows from Definition 6.
(2) Follows from Definition 5.
(3) It follows from the construction of v' that $\forall s \in S : v'(s) = v(s)$.
(4) By Definition 7, for statements $s \in S \setminus \{x\}$, the acceptance condition remains unchanged in D', so $w(C_s) = w(C'_s)$ and therefore $\bigsqcap\{w(C_s) \mid w \in [v']_2\} = \bigsqcap\{w(C'_s) \mid w \in [v']_2\}$.

If $s = x$, then we show $\bigsqcap\{w(C_x) \mid w \in [v']_2\} = \bigsqcap\{w(C'_x) \mid w \in [v']_2\}$ by three cases:

1. $\forall w \in [v']_2 : w(C'_{x^*}) = \mathbf{t}$, then $v'(x^*) = \mathbf{t}$ (by construction of v'). From this it follows that $\forall w \in [v']_2 : w(C'_{x^*}) = w(\phi) = \mathbf{t}$ and thus $w(\neg\phi) = \mathbf{f}$. Similarly, it follows that $\forall w \in [v']_2 : w(x^*) = \mathbf{t}$ and thus $w(\neg x^*) = \mathbf{f}$. Therefore $\forall w \in [v']_2 : w(\neg\phi) = w(\neg x^*) = \mathbf{f}$. From $C_x = \neg\phi$ and $C'_x = \neg x^*$ it follows that $\bigsqcap\{w(C_x) \mid w \in [v']_2\} = \bigsqcap\{w(C'_x) \mid w \in [v']_2\} = \mathbf{f}$.

2. $\forall w \in [v']_2 : w(C'_{x^*}) = \mathbf{f}$, then $v'(x^*) = \mathbf{f}$ (by construction of v'). From this it follows that $\forall w \in [v']_2 : w(C'_{x^*}) = w(\phi) = \mathbf{f}$ and thus $w(\neg\phi) = \mathbf{t}$. Similarly, it follows that $\forall w \in [v']_2 : w(x^*) = \mathbf{f}$ and thus $w(\neg x^*) = \mathbf{t}$. Therefore $\forall w \in [v']_2 : w(\neg\phi) = w(\neg x^*) = \mathbf{t}$. From $C_x = \neg\phi$ and $C'_x = \neg x^*$ it follows that $\bigsqcap\{w(C_x) \mid w \in [v']_2\} = \bigsqcap\{w(C'_x) \mid w \in [v']_2\} = \mathbf{t}$.

3. $\exists w \in [v']_2 : w(C'_{x^*}) = \mathbf{t}$ and $\exists w \in [v']_2 : w(C'_{x^*}) = \mathbf{f}$, then $v'(x^*) = \mathbf{u}$ (by construction of v'). From $C'_{x^*} = \phi$ it follows that $\exists w \in [v']_2 : w(\phi) = \mathbf{t}$ and $\exists w \in [v']_2 : w(\phi) = \mathbf{f}$. Similarly, $\exists w \in [v']_2 : w(x^*) = \mathbf{t}$ and $\exists w \in [v']_2 : w(x^*) = \mathbf{f}$. From $C_x = \neg\phi$ and $C'_x = \neg x^*$ it follows that $\bigsqcap\{w(C_x) \mid w \in [v']_2\} = \bigsqcap\{w(C'_x) \mid w \in [v']_2\} = \mathbf{u}$.

Hence in all cases it holds that $\bigsqcap\{w(C_x) \mid w \in [v']_2\} = \bigsqcap\{w(C'_x) \mid w \in [v']_2\}$.

(5) Let $s' \in S'$. If $s' \in S$, then it follows from (4) that $v'(s') \leq_i \bigsqcap\{w(C'_{s'}) \mid w \in [v']_2\}$. If $s' \notin S$, then $s' = x^*$. It follows from the construction of v' that $v'(x^*) = \bigsqcap\{w(C'_{x^*}) \mid w \in [v']_2\}$.

(6) Follows from Definition 5.

(7) Follows from Definition 6.

Lemma 2. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its $\neg$-*decomposition at* x. Let v be a three-valued interpretation of D'. Then $v \in ad(D') \implies v{\downarrow_S} \in ad(D)$.

Proof. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its $\neg$-*decomposition at* x. Let v be a three-valued interpretation of D'. Suppose v is admissible in D'. We aim to show that $v \in ad(D') \implies v{\downarrow_S} \in ad(D)$.

$$
\begin{aligned}
v \in ad(D') &\implies \forall s' \in S' : v(s') \leq_i \Gamma_{D'}(v)(s') && (8)\\
&\implies \forall s' \in S' : v(s') \leq_i \bigsqcap\{w(C'_{s'}) \mid w \in [v]_2\} && (9)\\
&\implies \forall s \in S : v(s) \leq_i \bigsqcap\{w(C'_s) \mid w \in [v]_2\} && (10)\\
&\implies \forall s \in S : v(s) \leq_i \bigsqcap\{w(C_s) \mid w \in [v]_2\} && (11)\\
&\implies \forall s \in S : v{\downarrow_S}(s) \leq_i \bigsqcap\{w(C_s) \mid w \in [v{\downarrow_S}]_2\} && (12)\\
&\implies \forall s \in S : v{\downarrow_S}(s) \leq_i \Gamma_D(v{\downarrow_S})(s) && (13)\\
&\implies v{\downarrow_S} \in ad(D) && (14)
\end{aligned}
$$

(8) Follows from Definition 6.

(9) Follows from Definition 5.

(10) Follows from $S \subset S'$.

(11) By Definition 7, for statements $s \in S \setminus \{x\}$, the acceptance condition remains unchanged in D', so $w(C_s) = w(C'_s)$ and therefore $\bigsqcap\{w(C_s) \mid w \in [v]_2\} = \bigsqcap\{w(C'_s) \mid w \in [v]_2\}$.

If $s = x$, then we show $v(x) \leq_i \bigsqcap\{w(C_x) \mid w \in [v]_2\}$ by three cases:

1. $v(x) = \mathbf{t}$, then $\bigsqcap\{w(C'_x) \mid w \in [v]_2\} = \mathbf{t}$. Therefore $\forall w \in [v]_2 : w(C'_x) = w(\neg x^*) = \mathbf{t}$ and $\forall w \in [v]_2 : w(x^*) = \mathbf{f}$ and thus $v(x^*) = \mathbf{f}$. From (10) it follows that $\bigsqcap\{w(C'_{x^*}) \mid w \in [v]_2\} = \mathbf{f}$ and thus $\forall w \in [v]_2 : w(C'_{x^*}) = w(\phi) = \mathbf{f}$. Hence $\forall w \in [v]_2 : w(\neg\phi) = w(C_x) = \mathbf{t}$ and $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{t}$.

2. $v(x) = \mathbf{f}$, then $\bigsqcap\{w(C'_x) \mid w \in [v]_2\} = \mathbf{f}$. Therefore $\forall w \in [v]_2 : w(C'_x) = w(\neg x^*) = \mathbf{f}$ and $\forall w \in [v]_2 : w(x^*) = \mathbf{t}$ and thus $v(x^*) = \mathbf{t}$. From (10) it follows that $\bigsqcap\{w(C'_{x^*}) \mid w \in [v]_2\} = \mathbf{t}$ and thus $\forall w \in [v]_2 : w(C'_{x^*}) = w(\phi) = \mathbf{t}$. Hence $\forall w \in [v]_2 : w(\neg\phi) = w(C_x) = \mathbf{f}$ and $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{f}$.

3. $v(x) = \mathbf{u}$, then it follows from Definition 4 that $v(x) \leq_i \bigsqcap\{w(C_x) \mid w \in [v]_2\}$.

Hence in all cases it holds that $v(x) \leq_i \bigsqcap\{w(C_x) \mid w \in [v]_2\}$.

(12) By Definition 2 it holds that $\forall s \in S : v{\downarrow_S}(s) = v(s)$.

(13) Follows from Definition 5.

(14) Follows from Definition 6.

Lemma 3. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its $\neg$-*decomposition at* x. Let v be a three-valued interpretation of D. Then $v \in co(D) \implies \exists v' \in co(D') : v'{\downarrow_S} = v$.

Proof. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its $\neg$-*decomposition at* x. Let v be a three-valued interpretation of D and suppose v is complete in D. We construct an interpretation v' such that $\forall s \in S : v'(s) = v(s)$ and $v'(x^*) = \bigsqcap\{w(C'_{x^*}) \mid w \in [v']_2\}$, thus fulfilling the restriction $v'{\downarrow_S} = v$.

Now we show that this v' is complete in D':

$$
\begin{aligned}
v \in co(D) &\implies \forall s \in S : v(s) = \Gamma_D(v)(s) && (15)\\
&\implies \forall s \in S : v(s) = \bigsqcap\{w(C_s) \mid w \in [v]_2\} && (16)\\
&\implies \forall s \in S : v'(s) = \bigsqcap\{w(C_s) \mid w \in [v']_2\} && (17)\\
&\implies \forall s \in S : v'(s) = \bigsqcap\{w(C'_s) \mid w \in [v']_2\} && (18)\\
&\implies \forall s' \in S' : v'(s') = \bigsqcap\{w(C'_{s'}) \mid w \in [v']_2\} && (19)\\
&\implies \forall s' \in S' : v'(s') = \Gamma_{D'}(v')(s') && (20)\\
&\implies v' \in co(D') && (21)
\end{aligned}
$$

By the same reasoning as (1)-(7).

Lemma 4. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its $\neg$-*decomposition at* x. Let v be a three-valued interpretation of D'. Then $v \in co(D') \implies v{\downarrow_S} \in co(D)$.

Proof. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its $\neg$-*decomposition at* x. Let v be a three-valued interpretation of D' and suppose v is complete in D'. We aim to show that $v \in co(D') \implies v{\downarrow_S} \in co(D)$.

$$
\begin{aligned}
v \in co(D') &\implies \forall s' \in S' : v(s') = \Gamma_{D'}(v)(s') && (22)\\
&\implies \forall s' \in S' : v(s') = \bigsqcap\{w(C'_{s'}) \mid w \in [v]_2\} && (23)\\
&\implies \forall s \in S : v(s) = \bigsqcap\{w(C'_s) \mid w \in [v]_2\} && (24)\\
&\implies \forall s \in S : v(s) = \bigsqcap\{w(C_s) \mid w \in [v]_2\} && (25)\\
&\implies \forall s \in S : v{\downarrow_S}(s) = \bigsqcap\{w(C_s) \mid w \in [v{\downarrow_S}]_2\} && (26)\\
&\implies \forall s \in S : v{\downarrow_S}(s) = \Gamma_D(v{\downarrow_S})(s) && (27)\\
&\implies v{\downarrow_S} \in co(D) && (28)
\end{aligned}
$$

(22) Follows from Definition 6.
(23) Follows from Definition 5.
(24) Follows from $S \subset S'$.
(25) By Definition 7, for statements $s \in S \setminus \{x\}$, the acceptance condition remains unchanged in D', so $w(C_s) = w(C'_s)$ and therefore $\bigsqcap\{w(C_s) \mid w \in [v]_2\} = \bigsqcap\{w(C'_s) \mid w \in [v]_2\}$.

If $s = x$, then we show $v(x) = \bigsqcap\{w(C_x) \mid w \in [v]_2\}$ by three cases:

1. $v(x) = \mathbf{t}$, then $\bigsqcap\{w(C'_x) \mid w \in [v]_2\} = \mathbf{t}$. Therefore $\forall w \in [v]_2 : w(C'_x) = w(\neg x^*) = \mathbf{t}$ and $\forall w \in [v]_2 : w(x^*) = \mathbf{f}$ and thus $v(x^*) = \mathbf{f}$. From (24) it follows that $\bigsqcap\{w(C'_{x^*}) \mid w \in [v]_2\} = \mathbf{f}$ and $\forall w \in [v]_2 : w(C'_{x^*}) = w(\phi) = \mathbf{f}$. Hence $\forall w \in [v]_2 : w(\neg\phi) = w(C_x) = \mathbf{t}$ and $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{t}$.

2. $v(x) = \mathbf{f}$, then $\bigsqcap\{w(C'_x) \mid w \in [v]_2\} = \mathbf{f}$. Therefore $\forall w \in [v]_2 : w(C'_x) = w(\neg x^*) = \mathbf{f}$ and $\forall w \in [v]_2 : w(x^*) = \mathbf{t}$ and thus $v(x^*) = \mathbf{t}$. From (24) it follows that $\bigsqcap\{w(C'_{x^*}) \mid w \in [v]_2\} = \mathbf{t}$ and $\forall w \in [v]_2 : w(C'_{x^*}) = w(\phi) = \mathbf{t}$. Hence $\forall w \in [v]_2 : w(\neg\phi) = w(C_x) = \mathbf{f}$ and $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{f}$.

3. $v(x) = \mathbf{u}$, then $\bigsqcap\{w(C'_x) \mid w \in [v]_2\} = \mathbf{u}$. Therefore $\exists w \in [v]_2 : w(C'_x) = w(\neg x^*) = \mathbf{t}$ and $\exists w \in [v]_2 : w(C'_x) = w(\neg x^*) = \mathbf{f}$ and thus $v(x^*) = \mathbf{u}$. From (24) it follows that $\bigsqcap\{w(C'_{x^*}) \mid w \in [v]_2\} = \mathbf{u}$ and thus it follows that $\exists w \in [v]_2 : w(C'_{x^*}) = w(\phi) = \mathbf{t}$ and $\exists w \in [v]_2 : w(C'_{x^*}) = w(\phi) = \mathbf{f}$. From $C_x = \neg\phi$ it thus follows that $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{u}$.

Hence in all cases it holds that $v(x) = \bigsqcap\{w(C_x) \mid w \in [v]_2\}$.

(26) By Definition 2 it holds that $\forall s \in S : v\downarrow_S(s) = v(s)$.

(27) Follows from Definition 5.

(28) Follows from Definition 6.

Theorem. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its $\neg$-decomposition at x. For any semantics $\sigma \in \{ad, pr, co, gr\}$: $\sigma(D) = \sigma(D')\downarrow_S$.

Proof. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its *$\neg$-decomposition at x*. We aim to show that

1. $ad(D) = ad(D')\downarrow_S$. This follows from Lemma 1 and Lemma 2.
2. $pr(D) = pr(D')\downarrow_S$. By Definition 6, the preferred interpretations are $\leq_i$-maximal admissible interpretations. We establish the equality by proving set inclusion in both directions.
   (a) For $pr(D) \subseteq pr(D')\downarrow_S$: Let $v \in pr(D)$, then v is $\leq_i$-maximal in $ad(D)$. From Lemma 1 it follows that $\exists v' \in ad(D') : v'\downarrow_S = v$. Suppose that v' is not $\leq_i$-maximal in $ad(D')$. Then there is a $v'' \in pr(D')$ such that $v' <_i v''$. Two cases:
      i. $v'\downarrow_S <_i v''\downarrow_S$, then since $v'' \in pr(D')$, it follows from Lemma 2 that $v''\downarrow_S \in ad(D)$, which contradicts $v'\downarrow_S = v$ being preferred in D. So this case is impossible.
      ii. $v'\downarrow_S = v''\downarrow_S$, then $v'(x^*) <_i v''(x^*)$. Since $v'\downarrow_S = v$ it follows that $v''\downarrow_S = v$. Therefore $\exists v'' \in pr(D') : v''\downarrow_S = v$ and thus $v \in pr(D')\downarrow_S$.

   (b) For $pr(D')\downarrow_S \subseteq pr(D)$: Let $v \in pr(D')$, then v is $\leq_i$-maximal in $ad(D')$. From Lemma 2 it follows that $v\downarrow_S \in ad(D)$. Suppose that $v\downarrow_S$ is not $\leq_i$-maximal in $ad(D)$. Then there is a $v' \in pr(D)$ such that $v\downarrow_S <_i v'$.
   From Lemma 1 it follows that $\exists v'' \in ad(D') : v''\downarrow_S = v'$. Since $v\downarrow_S <_i v'$, we have $v\downarrow_S <_i v''\downarrow_S$. From this it follows that $v <_i v''$ and thus contradicts that v is $\leq_i$-maximal in $ad(D')$. Therefore $v\downarrow_S$ is $\leq_i$-maximal in $ad(D)$ and thus $v\downarrow_S \in pr(D)$.

   We have proven the inclusion in both directions, hence $pr(D) = pr(D')\downarrow_S$.
3. $co(D) = co(D')\downarrow_S$. This follows from Lemma 3 and Lemma 4.
4. $gr(D) = gr(D')\downarrow_S$. By Definition 6, the grounded interpretation is unique and the $\leq_i$-minimal interpretation. Let $v = gr(D)$ be the grounded interpretation of D. Then $v \in co(D)$ and v is the $\leq_i$-least element in $co(D)$. From Lemma 3, there exists a $v' \in co(D')$ such that $v'\downarrow_S = v$. Suppose that v' is not the grounded interpretation of D'. Then there exists a $w = gr(D')$ such that $w <_i v'$. Two cases:
   (a) $w\downarrow_S <_i v'\downarrow_S$, then it follows from Lemma 4 that $w\downarrow_S \in co(D)$. This contradicts that $v'\downarrow_S = v$ is the $\leq_i$-minimal interpretation. So this case is impossible.
   (b) $w\downarrow_S = v'\downarrow_S$, then $w(x^*) <_i v'(x^*)$. Since $v'\downarrow_S = v$ it follows that $w\downarrow_S = v$. Therefore $gr(D')\downarrow_S = w\downarrow_S = v = gr(D)$.

Conjunction Decomposition

Lemma 5. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its $\wedge$-*decomposition at* x. Let v be a three-valued interpretation of D and let v' be a three-valued interpretation of D' such that $\forall s \in S : v'(s) = v(s)$ and $v'(x^*) = \bigsqcap\{w(C'_{x^*}) \mid w \in [v']_2\}$. Then $\forall s \in S : \bigsqcap\{w(C_s) \mid w \in [v']_2\} = \bigsqcap\{w(C'_s) \mid w \in [v']_2\}$.

Proof. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its $\wedge$-*decomposition at* x. Let v be a three-valued interpretation of D and let v' be a three-valued interpretation of D' such that $\forall s \in S : v'(s) = v(s)$ and $v'(x^*) = \bigsqcap\{w(C'_{x^*}) \mid w \in [v']_2\}$. Let $s \in S$. We aim to show that $\bigsqcap\{w(C_s) \mid w \in [v']_2\} = \bigsqcap\{w(C'_s) \mid w \in [v']_2\}$.

If $s \neq x$, then by Definition 8, the acceptance condition remains unchanged in D'. Hence $\forall w \in [v']_2 : w(C_s) = w(C'_s)$ and therefore $\bigsqcap\{w(C_s) \mid w \in [v']_2\} = \bigsqcap\{w(C'_s) \mid w \in [v']_2\}$.

If $s = x$, we prove the equality by two cases:

1. $C_x = \phi \wedge \psi$ is a contradiction, then $C'_x = \phi \wedge \bot$. From this it follows that $\forall w \in [v']_2 : w(C_x) = w(\phi \wedge \psi) = \mathbf{f}$ and $w(C'_x) = w(\phi \wedge \bot) = \mathbf{f}$. Therefore $\bigsqcap\{w(C_x) \mid w \in [v']_2\} = \bigsqcap\{w(C'_x) \mid w \in [v']_2\} = \mathbf{f}$.

2. $C_x = \phi \wedge \psi$ is not a contradiction, then $C'_x = \phi \wedge x^*$. We differentiate three cases:
   (a) $\forall w \in [v']_2 : w(C'_{x^*}) = w(\psi) = \mathbf{t}$, then $v(x^*) = \mathbf{t}$ (by construction of v') and therefore $\forall w \in [v']_2 : w(x^*) = \mathbf{t}$. Hence $\forall w \in [v']_2 : w(x^*) = w(\psi) = \mathbf{t}$ and thus $w(\phi \wedge \psi) = w(\phi \wedge x^*)$. From $C_x = \phi \wedge \psi$ and $C'_x = \phi \wedge x^*$ it follows that $\bigsqcap\{w(C_x) \mid w \in [v']_2\} = \bigsqcap\{w(C'_x) \mid w \in [v']_2\}$.

   (b) $\forall w \in [v']_2 : w(C'_{x^*}) = w(\psi) = \mathbf{f}$, then $v(x^*) = \mathbf{f}$ (by construction of v') and therefore $\forall w \in [v']_2 : w(x^*) = \mathbf{f}$. Hence $\forall w \in [v']_2 : w(x^*) = w(\psi) = \mathbf{f}$ and thus $w(\phi \wedge \psi) = w(\phi \wedge x^*)$. From $C_x = \phi \wedge \psi$ and $C'_x = \phi \wedge x^*$ it follows that $\bigsqcap\{w(C_x) \mid w \in [v']_2\} = \bigsqcap\{w(C'_x) \mid w \in [v']_2\} = \mathbf{f}$.

   (c) $\exists w \in [v']_2 : w(C'_{x^*}) = w(\psi) = \mathbf{t}$ and $\exists w \in [v']_2 : w(C'_{x^*}) = w(\psi) = \mathbf{f}$, then $v'(x^*) = \mathbf{u}$ (by construction of v'). The valuation of $C_x = \phi \wedge \psi$ and $C'_x = \phi \wedge x^*$ now depends on the value of ϕ.
      i. $\forall w \in [v']_2 : w(\phi) = \mathbf{t}$, then $\exists w \in [v']_2 : w(\phi \wedge \psi) = \mathbf{t}$ and $\exists w \in [v']_2 : w(\phi \wedge \psi) = \mathbf{f}$. Similarly, $\exists w \in [v']_2 : w(\phi \wedge x^*) = \mathbf{t}$ and $\exists w \in [v']_2 : w(\phi \wedge x^*) = \mathbf{f}$. From this it follows that $\bigsqcap\{w(C_x) \mid w \in [v']_2\} = \bigsqcap\{w(C'_x) \mid w \in [v']_2\} = \mathbf{u}$.

      ii. $\forall w \in [v']_2 : w(\phi) = \mathbf{f}$, then $\forall w \in [v']_2 : w(\phi \wedge \psi) = \mathbf{f}$ and $w(\phi \wedge x^*) = \mathbf{f}$. From this it follows that $\bigsqcap\{w(C_x) \mid w \in [v']_2\} = \bigsqcap\{w(C'_x) \mid w \in [v']_2\} = \mathbf{f}$.

      iii. $\exists w \in [v']_2 : w(\phi) = \mathbf{t}$ and $\exists w \in [v']_2 : w(\phi) = \mathbf{f}$, then we have to examine three more cases:
         A. $\forall w \in [v']_2 : w(\phi \wedge \psi) = \mathbf{t}$. This case is *impossible* as there always exists a w, such that $w(\phi) = \mathbf{f}$ or a w such that $w(\psi) = \mathbf{f}$.

         B. $\forall w \in [v']_2 : w(\phi \wedge \psi) = \mathbf{f}$, then even though both ϕ and ψ sometimes valuate to $\mathbf{t}$ and sometimes to $\mathbf{f}$, their conjunction always valuates to $\mathbf{f}$. This implies that $\phi \wedge \psi$ is a logical contradiction, and thus it contradicts the case that $\phi \wedge \psi$ is <u>not</u> a contradiction.

C. $\exists w \in [v']_2 : w(\phi \wedge \psi) = \mathbf{t}$ and $\exists w \in [v']_2 : w(\phi \wedge \psi) = \mathbf{f}$, then $\bigsqcap\{w(C_x) \mid w \in [v']_2\} = \mathbf{u}$. From $v'(x^*) = \mathbf{u}$ it follows that $\exists w \in [v']_2 : w(\phi \wedge x^*) = \mathbf{t}$ and $\exists w \in [v']_2 : w(\phi \wedge x^*) = \mathbf{f}$. Therefore $\bigsqcap\{w(C_x) \mid w \in [v']_2\} = \mathbf{u} = \bigsqcap\{w(C'_x) \mid w \in [v']_2\}$.

Hence in all cases it holds that $\bigsqcap\{w(C_x) \mid w \in [v']_2\} = \bigsqcap\{w(C'_s) \mid w \in [v']_2\}$.

Lemma 6. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its $\wedge$-*decomposition at* x. Let v be a three-valued interpretation of D. Then $v \in ad(D) \implies \exists v' \in ad(D') : v'{\downarrow_S} = v$.

Proof. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its $\wedge$-*decomposition at* x. Let v be a three-valued interpretation of D. Suppose v is admissible in D. We construct an interpretation v' for D' such that $\forall s \in S : v'(s) = v(s)$ and $v'(x^*) = \bigsqcap\{w(C'_{x^*}) \mid w \in [v']_2\}$, thus fulfilling the restriction that $v'{\downarrow_S} = v$.

Now we show that this v' is admissible in D':

$$
\begin{aligned}
v \in ad(D) &\implies \forall s \in S : v(s) \leq_i \Gamma_D(v)(s) && (29)\\
&\implies \forall s \in S : v(s) \leq_i \bigsqcap\{w(C_s) \mid w \in [v]_2\} && (30)\\
&\implies \forall s \in S : v'(s) \leq_i \bigsqcap\{w(C_s) \mid w \in [v']_2\} && (31)\\
&\implies \forall s \in S : v'(s) \leq_i \bigsqcap\{w(C'_s) \mid w \in [v']_2\} && (32)\\
&\implies \forall s' \in S' : v'(s') \leq_i \bigsqcap\{w(C'_{s'}) \mid w \in [v']_2\} && (33)\\
&\implies \forall s' \in S' : v'(s') \leq_i \Gamma_{D'}(v')(s') && (34)\\
&\implies v' \in ad(D') && (35)
\end{aligned}
$$

(29) Follows from Definition 6.
(30) Follows from Definition 5.
(31) It follows from the construction of v' that $\forall s \in S : v'(s) = v(s)$.
(32) Follows from Lemma 5.
(33) Let $s' \in S'$. If $s' \in S$, then it follows from (31) that $v'(s') \leq_i \bigsqcap\{w(C'_{s'}) \mid w \in [v']_2\}$. If $s' \notin S$, then $s' = x^*$. It follows from the construction of v' that $v'(x^*) = \bigsqcap\{w(C'_{x^*}) \mid w \in [v']_2\}$.
(34) Follows from Definition 5.
(35) Follows from Definition 6.

Lemma 7. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its $\wedge$-*decomposition at* x. Let v be a three-valued interpretation of D'. Then $v \in ad(D') \implies v{\downarrow_S} \in ad(D)$.

Proof. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its $\wedge$-*decomposition at* x. Let v be a three-valued interpretation of D'. Suppose v is admissible in D'. We aim to show that $v \in ad(D') \implies v{\downarrow_S} \in ad(D)$.

$$
\begin{aligned}
v \in ad(D') &\implies \forall s' \in S' : v(s') \leq_i \Gamma_{D'}(v)(s') && (36)\\
&\implies \forall s' \in S' : v(s') \leq_i \bigsqcap\{w(C'_{s'}) \mid w \in [v]_2\} && (37)\\
&\implies \forall s \in S : v(s) \leq_i \bigsqcap\{w(C'_s) \mid w \in [v]_2\} && (38)\\
&\implies \forall s \in S : v(s) \leq_i \bigsqcap\{w(C_s) \mid w \in [v]_2\} && (39)\\
&\implies \forall s \in S : v{\downarrow_S}(s) \leq_i \bigsqcap\{w(C_s) \mid w \in [v{\downarrow_S}]_2\} && (40)\\
&\implies \forall s \in S : v{\downarrow_S}(s) \leq_i \Gamma_D(v{\downarrow_S})(s) && (41)\\
&\implies v{\downarrow_S} \in ad(D) && (42)
\end{aligned}
$$

(36) Follows from Definition 6.

(37) Follows from Definition 5.

(38) Follows from $S \subset S'$.

(39) Let $s \in S$. If $s \neq x$, then by Definition 8, the acceptance condition remains unchanged. Hence $\forall w \in [v]_2 : w(C_s) = w(C'_s)$ and therefore $\bigsqcap\{w(C_s) \mid w \in [v]_2\} = \bigsqcap\{w(C'_s) \mid w \in [v]_2\}$.

If $s = x$, then we show $v(x) \leq_i \bigsqcap\{w(C_x) \mid w \in [v]_2\}$ by two cases:

1. $C'_x = \phi \wedge \bot$, then $\phi \wedge \psi$ is a logical contradiction. Therefore $\forall w \in [v]_2 : w(\phi \wedge \bot) = \mathbf{f}$ and $w(\phi \wedge \psi) = \mathbf{f}$. From this it follows that $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \bigsqcap\{w(C'_x) \mid w \in [v]_2\} = \mathbf{f}$.

2. $C_x = \phi \wedge x^*$, then $\phi \wedge \psi$ is not a logical contradiction and we differentiate three cases:

   (a) $v(x) = \mathbf{t}$, then $\bigsqcap\{w(C'_x) \mid w \in [v]_2\} = \mathbf{t}$ and $\forall w \in [v]_2 : w(C'_x) = w(\phi \wedge x^*) = \mathbf{t}$. From this it follows that $\forall w \in [v]_2 : w(\phi) = \mathbf{t}$ and $w(x^*) = \mathbf{t}$ and thus $v(x^*) = \mathbf{t}$. Hence $\bigsqcap\{w(C'_{x^*}) \mid w \in [v]_2\} = \mathbf{t}$ and $\forall w \in [v]_2 : w(C'_{x^*}) = w(\psi) = \mathbf{t}$. Therefore $\forall w \in [v]_2 : w(\phi \wedge \psi) = \mathbf{t}$ and thus $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \bigsqcap\{w(C'_x) \mid w \in [v]_2\} = \mathbf{t}$.

   (b) $v(x) = \mathbf{f}$, then $\bigsqcap\{w(C'_x) \mid w \in [v]_2\} = \mathbf{f}$ and $\forall w \in [v]_2 : w(\phi \wedge x^*) = \mathbf{f}$. We show that $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{f}$ by three cases:

      i. $\forall w \in [v]_2 : w(x^*) = \mathbf{t}$, then $\forall w \in [v]_2 : w(\phi) = \mathbf{f}$, in order to always valuate the conjunction to $\mathbf{f}$. From this it follows that $\forall w \in [v]_2 : w(\phi \wedge \psi) = \mathbf{f}$ and thus $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{f}$.

      ii. $\forall w \in [v]_2 : w(x^*) = \mathbf{f}$, then $\bigsqcap\{w(C'_{x^*}) \mid w \in [v]_2\} = \mathbf{f}$ and $\forall w \in [v]_2 : w(C'_{x^*}) = w(\psi) = \mathbf{f}$. From this it follows that $\forall w \in [v]_2 : w(\phi \wedge \psi) = \mathbf{f}$ and thus $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{f}$.

      iii. $\exists w \in [v]_2 : w(x^*) = \mathbf{t}$ and $\exists w \in [v]_2 : w(x^*) = \mathbf{f}$, then $\forall w \in [v]_2 : w(\phi) = \mathbf{f}$, in order to always valuate the conjunction to $\mathbf{f}$. From this is follows that $\forall w \in [v]_2 : w(\phi \wedge \psi) = \mathbf{f}$ and thus $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{f}$.

      Hence whenever $v(x) = \mathbf{f}$ it holds in all cases that $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{f}$.

   (c) $v(x) = \mathbf{u}$, then it follows from Definition 4 that $v(x) \leq_i \bigsqcap\{w(C_x) \mid w \in [v]_2\}$.

(40) By Definition 2 it holds that $\forall s \in S : v\downarrow_S (s) = v(s)$.

(41) Follows from Definition 5.

(42) Follows from Definition 6.

Lemma 8. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its $\wedge$-*decomposition at* x. Let v be a three-valued interpretation of D. Then $v \in co(D) \implies \exists v' \in co(D') : v'\downarrow_S = v$.

Proof. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its $\wedge$-*decomposition at* x. Let v be a three-valued interpretation of D. Suppose v is complete in D. We construct an interpretation v' for D' such that $\forall s \in S : v'(s) = v(s)$ and $v'(x^*) = \bigsqcap\{w(C'_{x^*}) \mid w \in [v']_2\}$, thus fulfilling the restriction that $v'\downarrow_S = v$.

Now we show that this v' is complete in D':

$$
\begin{aligned}
v \in co(D) &\implies \forall s \in S : v(s) = \Gamma_D(v)(s) && (43)\\
&\implies \forall s \in S : v(s) = \bigsqcap\{w(C_s) \mid w \in [v]_2\} && (44)\\
&\implies \forall s \in S : v'(s) = \bigsqcap\{w(C_s) \mid w \in [v']_2\} && (45)\\
&\implies \forall s \in S : v'(s) = \bigsqcap\{w(C'_s) \mid w \in [v']_2\} && (46)\\
&\implies \forall s' \in S' : v'(s') = \bigsqcap\{w(C'_{s'}) \mid w \in [v']_2\} && (47)\\
&\implies \forall s' \in S' : v'(s') = \Gamma_{D'}(v')(s') && (48)\\
&\implies v' \in co(D') && (49)
\end{aligned}
$$

By the same reasoning as (29)-(35).

Lemma 9. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its $\wedge$-*decomposition at* x. Let v be a three-valued interpretation of D'. Then $v \in co(D') \implies v{\downarrow_S} \in co(D)$.

Proof. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its $\wedge$-*decomposition at* x. Let v be a three-valued interpretation of D'. Suppose v is complete in D'. We aim to show that $v \in co(D') \implies v{\downarrow_S} \in co(D)$.

$$
\begin{aligned}
v \in co(D') &\implies \forall s' \in S' : v(s') = \Gamma_{D'}(v)(s') && (50)\\
&\implies \forall s' \in S' : v(s') = \bigsqcap\{w(C'_{s'}) \mid w \in [v]_2\} && (51)\\
&\implies \forall s \in S : v(s) = \bigsqcap\{w(C'_s) \mid w \in [v]_2\} && (52)\\
&\implies \forall s \in S : v(s) = \bigsqcap\{w(C_s) \mid w \in [v]_2\} && (53)\\
&\implies \forall s \in S : v{\downarrow_S}(s) = \bigsqcap\{w(C_s) \mid w \in [v{\downarrow_S}]_2\} && (54)\\
&\implies \forall s \in S : v{\downarrow_S}(s) = \Gamma_D(v{\downarrow_S})(s) && (55)\\
&\implies v{\downarrow_S} \in co(D) && (56)
\end{aligned}
$$

(50) Follows from Definition 6.
(51) Follows from Definition 5.
(52) Follows from $S \subset S'$.
(53) Let $s \in S$. If $s \neq x$, then by Definition 8, the acceptance condition remains unchanged. Hence $\forall w \in [v]_2 : w(C_s) = w(C'_s)$ and therefore $\bigsqcap\{w(C_s) \mid w \in [v]_2\} = \bigsqcap\{w(C'_s) \mid w \in [v]_2\}$.

If $s = x$, then we show $v(x) = \bigsqcap\{w(C_x) \mid w \in [v]_2\}$ by two cases:

1. $C'_x = \phi \wedge \bot$, then $\phi \wedge \psi$ is a logical contradiction. Therefore $\forall w \in [v]_2 : w(\phi \wedge \bot) = \mathbf{f}$ and $w(\phi \wedge \psi) = \mathbf{f}$. From this it follows that $\bigsqcap\{w(C'_x) \mid w \in [v]_2\} = \bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{f}$.

2. $C_x = \phi \wedge x^*$, then $\phi \wedge \psi$ is not a logical contradiction and we differentiate three cases:
   (a) $v(x) = \mathbf{t}$, then $\bigsqcap\{w(C'_x) \mid w \in [v]_2\} = \mathbf{t}$ and $\forall w \in [v]_2 : w(C'_x) = w(\phi \wedge x^*) = \mathbf{t}$. From this it follows that $\forall w \in [v]_2 : w(\phi) = \mathbf{t}$ and $w(x^*) = \mathbf{t}$ and thus $v(x^*) = \mathbf{t}$. Hence $\bigsqcap\{w(C'_{x^*}) \mid w \in [v]_2\} = \mathbf{t}$ and $\forall w \in [v]_2 : w(C'_{x^*}) = w(\psi) = \mathbf{t}$. Therefore $\forall w \in [v]_2 : w(\phi \wedge \psi) = \mathbf{t}$ and thus $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \bigsqcap\{w(C'_x) \mid w \in [v]_2\} = \mathbf{t}$.

(b) $v(x) = \mathbf{f}$, then $\bigsqcap\{w(C'_x) \mid w \in [v]_2\} = \mathbf{f}$ and $\forall w \in [v]_2 : w(\phi \wedge x^*) = \mathbf{f}$. We show that $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{f}$ by three cases:

i. $\forall w \in [v]_2 : w(x^*) = \mathbf{t}$, then $\forall w \in [v]_2 : w(\phi) = \mathbf{f}$, in order to always valuate the conjunction to $\mathbf{f}$. From this it follows that $\forall w \in [v]_2 : w(\phi \wedge \psi) = \mathbf{f}$ and thus $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{f}$.

ii. $\forall w \in [v]_2 : w(x^*) = \mathbf{f}$, then $\bigsqcap\{w(C'_{x^*}) \mid w \in [v]_2\} = \mathbf{f}$ and $\forall w \in [v]_2 : w(C'_{x^*}) = w(\psi) = \mathbf{f}$. From this it follows that $\forall w \in [v]_2 : w(\phi \wedge \psi) = \mathbf{f}$ and thus $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{f}$.

iii. $\exists w \in [v]_2 : w(x^*) = \mathbf{t}$ and $\exists w \in [v]_2 : w(x^*) = \mathbf{f}$, then $\forall w \in [v]_2 : w(\phi) = \mathbf{f}$, in order to always valuate the conjunction to $\mathbf{f}$. From this is follows that $\forall w \in [v]_2 : w(\phi \wedge \psi) = \mathbf{f}$ and thus $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{f}$.

Hence whenever $v(x) = \mathbf{f}$ it holds in all cases that $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{f}$.

(c) $v(x) = \mathbf{u}$, then $\bigsqcap\{w(C'_x) \mid w \in [v]_2\} = \mathbf{u}$ and $\exists w \in [v]_2 : w(C'_x) = w(\phi \wedge x^*) = \mathbf{t}$ and $\exists w \in [v]_2 : w(C'_x) = (\phi \wedge x^*) = \mathbf{f}$. We show that $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{u}$ by three cases:

i. $\forall w \in [v]_2 : w(x^*) = \mathbf{t}$, then $\bigsqcap\{w(C'_{x^*}) \mid w \in [v]_2\} = \mathbf{t}$ and $\forall w \in [v]_2 : w(C'_{x^*}) = w(\psi) = \mathbf{t}$. From this it follows that $\forall w \in [v]_2 : w(x^*) = w(\psi)$ and thus $\exists w \in [v]_2 : w(\phi \wedge \psi) = \mathbf{t}$ and $\exists w \in [v]_2 : w(\phi \wedge \psi) = \mathbf{f}$. Therefore $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{u}$.

ii. $\forall w \in [v]_2 : w(x^*) = \mathbf{f}$. This case is *impossible* as there exists a w such that $w(\phi \wedge x^*) = \mathbf{t}$.

iii. $\exists w \in [v]_2 : w(x^*) = \mathbf{t}$ and $\exists w \in [v]_2 : w(x^*) = \mathbf{f}$, then $v(x^*) = \mathbf{u}$ and $\bigsqcap\{w(C'_{x^*}) \mid w \in [v]_2\} = \mathbf{u}$. From this it follows that $\exists w \in [v]_2 : w(C'_{x^*}) = w(\psi) = \mathbf{t}$ and $\exists w \in [v]_2 : w(C'_{x^*}) = w(\psi) = \mathbf{f}$. We have to examine three more cases:

A. $\forall w \in [v]_2 : w(\phi \wedge \psi) = \mathbf{t}$. This case is *impossible* as there always exists a w such that $w(\psi) = \mathbf{f}$.

B. $\forall w \in [v]_2 : w(\phi \wedge \psi) = \mathbf{f}$. This case is *impossible* as there exists a w such that $w(\phi \wedge x^*) = \mathbf{t}$.

C. $\exists w \in [v]_2 : w(\phi \wedge \psi) = \mathbf{t}$ and $\exists w \in [v]_2 : w(\phi \wedge \psi) = \mathbf{f}$, then $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{u}$.

Hence whenever $v(x) = \mathbf{u}$ it holds in all cases that $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{u}$.

(54) By Definition 2 it holds that $\forall s \in S : v\downarrow_S(s) = v(s)$.
(55) Follows from Definition 5.
(56) Follows from Definition 6.

Theorem. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its $\wedge$-decomposition at x. For any semantics $\sigma \in \{ad, pr, co, gr\}$: $\sigma(D) = \sigma(D')\downarrow_S$.

Proof. The proof follows the same structure as the proof of the theorem for $\neg$-decomposition, with Lemma 6 - Lemma 9 taking the place of Lemma 1 - Lemma 4 respectively.

Disjunction Decomposition

Lemma 10. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ it *$\vee$-decomposition at x*. Let v be a three-valued interpretation of D and let v' be a three-valued interpretation of D' such that

$\forall s \in S : v'(s) = v(s)$ and $v'(x^*) = \bigsqcap\{w(C'_{x^*}) \mid w \in [v']_2\}$. Then $\forall s \in S : \bigsqcap\{w(C_x) \mid w \in [v']_2\} = \bigsqcap\{w(C'_x) \mid w \in [v']_2\}$.

Proof. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ it $\vee$-*decomposition at* x. Let v be a three-valued interpretation of D and let v' be a three-valued interpretation of D' such that $\forall s \in S : v'(s) = v(s)$ and $v'(x^*) = \bigsqcap\{w(C'_{x^*}) \mid w \in [v']_2\}$. Let $s \in S$. We aim to show that $\bigsqcap\{w(C_x) \mid w \in [v']_2\} = \bigsqcap\{w(C'_x) \mid w \in [v']_2\}$.

If $s \neq x$, then by Definition 9, the acceptance condition remains unchanged in D'. Hence $\forall w \in [v']_2 : w(C_s) = w(C'_s)$ and therefore $\bigsqcap\{w(C_s) \mid w \in [v']_2\} = \bigsqcap\{w(C'_s) \mid w \in [v']_2\}$.

If $s = x$, we prove the equality by two cases:

1. $C_x = \phi \vee \psi$ is a tautology, then $C'_x = \phi \vee \top$. From this it follows that $\forall w \in [v']_2 : w(C_x) = w(\phi \vee \psi) = \mathbf{t}$ and $w(C'_x) = w(\phi \vee \top) = \mathbf{t}$. Therefore $\bigsqcap\{w(C_x) \mid w \in [v']_2\} = \bigsqcap\{w(C'_x) \mid w \in [v']_2\} = \mathbf{t}$.
2. $C_x = \phi \vee \psi$ is not a tautology, then $C'_x = \phi \vee x^*$. We differentiate three cases:
   (a) $\forall w \in [v']_2 : w(C'_{x^*}) = w(\psi) = \mathbf{t}$, then $v'(x^*) = \mathbf{t}$ (by construction of v') and therefore $\forall w(x^*) = \mathbf{t}$. Hence $\forall w \in [v']_2 : w(x^*) = w(\psi) = \mathbf{t}$ and thus $w(\phi \vee \psi) = w(\phi \vee x^*)$. From $C_x = \phi \vee \psi$ and $C'_x = \phi \vee x^*$ it follows that $\bigsqcap\{w(C_x) \mid w \in [v']_2\} = \bigsqcap\{w(C'_x) \mid w \in [v']_2\} = \mathbf{t}$.

   (b) $\forall w \in [v']_2 : w(C'_{x^*}) = w(\psi) = \mathbf{f}$, then $v'(x^*) = \mathbf{f}$ (by construction of v') and therefore $\forall w(x^*) = \mathbf{f}$. Hence $\forall w \in [v']_2 : w(x^*) = w(\psi) = \mathbf{f}$ and thus $w(\phi \vee \psi) = w(\phi \vee x^*)$. From $C_x = \phi \vee \psi$ and $C'_x = \phi \vee x^*$ it follows that $\bigsqcap\{w(C_x) \mid w \in [v']_2\} = \bigsqcap\{w(C'_x) \mid w \in [v']_2\}$.

   (c) $\exists w \in [v']_2 : w(C'_{x^*}) = w(\psi) = \mathbf{t}$ and $\exists w \in [v']_2 : w(C'_{x^*}) = w(\psi) = \mathbf{f}$, then $v'(x^*) = \mathbf{u}$ (by construction of v'). The valuation of $C_x = \phi \vee \psi$ and $C'_x = \phi \vee x^*$ now depends on the value of ϕ.
      i. $\forall w \in [v']_2 : w(\phi) = \mathbf{t}$, then $\forall w \in [v']_2 : w(\phi \vee \psi) = \mathbf{t}$ and $w(\phi \vee x^*) = \mathbf{t}$. From this it follows that $\bigsqcap\{w(C_x) \mid w \in [v']_2\} = \bigsqcap\{w(C'_x) \mid w \in [v']_2\} = \mathbf{t}$.

      ii. $\forall w \in [v']_2 : w(\phi) = \mathbf{f}$, then $\exists w \in [v']_2 : w(\phi \vee \psi) = \mathbf{t}$ and $\exists w \in [v']_2 : w(\phi \vee \psi) = \mathbf{f}$. Similarly, $\exists w \in [v']_2 : w(\phi \vee x^*) = \mathbf{t}$ and $\exists w \in [v']_2 : w(\phi \vee x^*) = \mathbf{f}$. From this it follows that $\bigsqcap\{w(C_x) \mid w \in [v']_2\} = \bigsqcap\{w(C'_x) \mid w \in [v']_2\} = \mathbf{u}$.

      iii. $\exists w \in [v']_2 : w(\phi) = \mathbf{t}$ and $\exists w \in [v']_2 : w(\phi) = \mathbf{f}$, then we have to examine three more cases:
         A. $\forall w \in [v']_2 : w(\phi \vee \psi) = \mathbf{t}$, then even though both ϕ and ψ sometimes valuate to $\mathbf{t}$ and sometimes to $\mathbf{f}$, their disjunction always valuates to $\mathbf{t}$. This implies that $\phi \vee \psi$ is a tautology, and thus it contradicts the case that $\phi \vee \psi$ is <u>not</u> a tautology.

         B. $\forall w \in [v']_2 : w(\phi \vee \psi) = \mathbf{f}$. This case is *impossible* as there always exists a w such that $w(\phi) = \mathbf{t}$ or a w such that $w(\psi) = \mathbf{t}$.

         C. $\exists w \in [v']_2 : w(\phi \vee \psi) = \mathbf{t}$ and $\exists w \in [v']_2 : w(\phi \vee \psi) = \mathbf{f}$, then $\bigsqcap\{w(C_x) \mid w \in [v']_2\} = \mathbf{u}$. From $v(x^*) = \mathbf{u}$ it follows that $\exists w \in [v']_2 : w(\phi \vee x^*) = \mathbf{t}$ and $\exists w \in [v']_2 : w(\phi \vee x^*) = \mathbf{f}$. Therefore $\bigsqcap\{w(C'_x) \mid w \in [v']_2\} = \mathbf{u} = \bigsqcap\{w(C_x) \mid w \in [v']_2\}$.

Hence in all cases it follows that $\bigsqcap\{w(C_x) \mid w \in [v']_2\} = \bigsqcap\{w(C'_x) \mid w \in [v']_2\}$.

Lemma 11. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its $\vee$-*decomposition at* x. Let v be a three-valued interpretation of D. Then $v \in ad(D) \implies \exists v' \in ad(D') : v'{\downarrow_S} = v$.

Proof. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its $\vee$-*decomposition at* x. Let v be a three-valued interpretation of D. Suppose v is admissible in D. We construct an interpretation v' for D' such that $\forall s \in S : v'(s) = v(s)$ and $v'(x^*) = \bigsqcap\{w(C'_{x^*}) \mid w \in [v']_2\}$, thus fulfilling the restriction that $v'{\downarrow_S} = v$.

Now we show that this v' is admissible in D':

$$
\begin{aligned}
v \in ad(D) &\implies \forall s \in S : v(s) \leq_i \Gamma_D(v)(s) && (57)\\
&\implies \forall s \in S : v(s) \leq_i \bigsqcap\{w(C_s) \mid w \in [v]_2\} && (58)\\
&\implies \forall s \in S : v'(s) \leq_i \bigsqcap\{w(C_s) \mid w \in [v']_2\} && (59)\\
&\implies \forall s \in S : v'(s) \leq_i \bigsqcap\{w(C'_s) \mid w \in [v']_2\} && (60)\\
&\implies \forall s' \in S' : v'(s') \leq_i \bigsqcap\{w(C'_{s'}) \mid w \in [v']_2\} && (61)\\
&\implies \forall s' \in S' : v'(s') \leq_i \Gamma_{D'}(v')(s') && (62)\\
&\implies v' \in ad(D') && (63)
\end{aligned}
$$

(57) Follows from Definition 6.
(58) Follows from Definition 5.
(59) It follows from the construction of v' that $\forall s \in S : v'(s) = v(s)$.
(60) Follows from Lemma 10.
(61) Let $s' \in S'$. If $s' \in S$, then it follows from (59) that $v'(s') \leq_i \bigsqcap\{w(C'_{s'}) \mid w \in [v']_2\}$. If $s' \notin S$, then $s' = x^*$. It follows from the construction of v' that $v'(x^*) = \bigsqcap\{w(C'_{x^*}) \mid w \in [v']_2\}$.
(62) Follows from Definition 5.
(63) Follows from Definition 6.

Lemma 12. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its $\vee$-*decomposition at* x. Let v be a three-valued interpretation of D'. Then $v \in ad(D') \implies v{\downarrow_S} \in ad(D)$.

Proof. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its $\vee$-*decomposition at* x. Let v be a three-valued interpretation of D'. Suppose v is admissible in D'. We aim to show that $v \in ad(D') \implies v{\downarrow_S} \in ad(D)$.

$$
\begin{aligned}
v \in ad(D') &\implies \forall s' \in S' : v(s') \leq_i \Gamma_{D'}(v)(s') && (64)\\
&\implies \forall s' \in S' : v(s') \leq_i \bigsqcap\{w(C'_{s'}) \mid w \in [v]_2\} && (65)\\
&\implies \forall s \in S : v(s) \leq_i \bigsqcap\{w(C'_s) \mid w \in [v]_2\} && (66)\\
&\implies \forall s \in S : v(s) \leq_i \bigsqcap\{w(C_s) \mid w \in [v]_2\} && (67)\\
&\implies \forall s \in S : v{\downarrow_S}(s) \leq_i \bigsqcap\{w(C_s) \mid w \in [v{\downarrow_S}]_2\} && (68)\\
&\implies \forall s \in S : v{\downarrow_S}(s) \leq_i \Gamma_D(v{\downarrow_S})(s) && (69)\\
&\implies v{\downarrow_S} \in ad(D) && (70)
\end{aligned}
$$

(64) Follows from Definition 6.
(65) Follows from Definition 5.
(66) Follows from $S \subset S'$.
(67) Let $s \in S$. If $s \neq x$, then by Definition 9, the acceptance condition remains unchanged. Hence $\forall w \in [v]_2 : w(C_s) = w(C'_s)$ and therefore $\bigsqcap\{w(C_s) \mid w \in [v]_2\} = \bigsqcap\{w(C'_s) \mid w \in [v]_2\}$.

If $s = x$, then we differentiate two cases:

1. $C'_x = \phi \vee \top$, then $\phi \vee \psi$ is a tautology. Therefore $\forall w \in [v']_2 : w(\phi \vee \top) = \mathbf{t}$ and $w(\phi \vee \psi) = \mathbf{t}$. From this it follows that $\bigsqcap\{w(C_x) \mid w \in [v']_2\} = \bigsqcap\{w(C'_x) \mid w \in [v']_2\}$.

2. $C'_x = \phi \vee x^*$, then $\phi \vee \psi$ is not a tautology and we differentiate three cases:
   (a) $v(x) = \mathbf{t}$, then $\bigsqcap\{w(C'_x) \mid w \in [v]_2\} = \mathbf{t}$ and $\forall w \in [v]_2 : w(C'_x) = w(\phi \vee x^*) = \mathbf{t}$. We show that $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{t}$ by three cases:
      i. $\forall w \in [v]_2 : w(x^*) = \mathbf{t}$, then $\bigsqcap\{w(C'_{x^*}) \mid w \in [v]_2\} = \mathbf{t}$ and $\forall w \in [v]_2 : w(C'_{x^*}) = w(\psi) = \mathbf{t}$. From this it follows that $\forall w \in [v]_2 : w(\phi \vee \psi) = w(C_x) = \mathbf{t}$ and thus $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{t}$.

      ii. $\forall w \in [v]_2 : w(x^*) = \mathbf{f}$, then $\forall w \in [v]_2 : w(\phi) = \mathbf{t}$, in order to always valuate the conjunction to $\mathbf{t}$. From this it follows that $\forall w \in [v]_2 : w(\phi \vee \psi) = w(C_x) = \mathbf{t}$ and thus $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{t}$.

      iii. $\exists w \in [v]_2 : w(x^*) = \mathbf{t}$ and $\exists w \in [v]_2 : w(x^*) = \mathbf{f}$, then $\forall w \in [v]_2 : w(\phi) = \mathbf{t}$, in order to always valuate the disjunction to $\mathbf{t}$. From this it follows that $\forall w \in [v]_2 : w(\phi \vee \psi) = w(C_x) = \mathbf{t}$ and thus $\bigsqcap\{w(C_x) \mid w \in [v']_2\} = \mathbf{t}$.
   Hence whenever $v(x) = \mathbf{t}$ it holds in all cases that $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{t}$.

   (b) $v(x) = \mathbf{f}$, then $\bigsqcap\{w(C'_x) \mid w \in [v]_2\} = \mathbf{f}$ and $\forall w \in [v]_2 : w(C'_x) = w(\phi \vee x^*) = \mathbf{f}$. From this it follows that $\forall w \in [v]_2 : w(\phi) = \mathbf{f}$ and $w(x^*) = \mathbf{f}$ and thus $v(x^*) = \mathbf{f}$. Hence $\bigsqcap\{w(C'_{x^*}) \mid w \in [v]_2\} = \mathbf{f}$ and $\forall w \in [v]_2 : w(C'_{x^*}) = w(\psi) = \mathbf{f}$. Therefore $\forall w \in [v]_2 : w(\phi \vee \psi) = w(C_x) = \mathbf{f}$ and thus $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \bigsqcap\{w(C'_x) \mid w \in [v]_2\} = \mathbf{f}$.

   (c) $v(x) = \mathbf{u}$, then it follows from Definition 4 that $v(x) \leq_i \bigsqcap\{w(C_x) \mid w \in [v]_2\}$.

(68) By Definition 2 it holds that $\forall s \in S : v\downarrow_S(s) = v(s)$.
(69) Follows from Definition 5.
(70) Follows from Definition 6.

Lemma 13. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its $\vee$-*decomposition at* x. Let v be a three-valued interpretation of D. Then $v \in co(D) \implies \exists v' \in co(D') : v'\downarrow_S = v$.

Proof. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its $\vee$-*decomposition at* x. Let v be a three-valued interpretation of D. Suppose v is complete in D. We construct an interpretation v' for D' such that $\forall s \in S : v'(s) = v(s)$ and $v'(x^*) = \bigsqcap\{w(C'_{x^*}) \mid w \in [v']_2\}$, thus fulfilling the restriction that $v'\downarrow_S = v$.

Now we show that this v' is complete in D':

$$v \in co(D) \implies \forall s \in S : v(s) = \Gamma_D(v)(s) \tag{71}$$
$$\implies \forall s \in S : v(s) = \bigsqcap\{w(C_s) \mid w \in [v]_2\} \tag{72}$$
$$\implies \forall s \in S : v'(s) = \bigsqcap\{w(C_s) \mid w \in [v']_2\} \tag{73}$$
$$\implies \forall s \in S : v'(s) = \bigsqcap\{w(C'_s) \mid w \in [v']_2\} \tag{74}$$
$$\implies \forall s' \in S' : v'(s') = \bigsqcap\{w(C'_{s'}) \mid w \in [v']_2\} \tag{75}$$
$$\implies \forall s' \in S' : v'(s') = \Gamma_{D'}(v')(s') \tag{76}$$
$$\implies v' \in co(D') \tag{77}$$

By the same reasoning as (57)-(63).

Lemma 14. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its $\vee$-*decomposition at* x. Let v be a three-valued interpretation of D'. Then $v \in co(D') \implies v{\downarrow_S} \in co(D)$.

Proof. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its $\vee$-*decomposition at* x. Let v be a three-valued interpretation of D'. Suppose v is complete in D'. We aim to show that $v \in co(D') \implies v{\downarrow_S} \in co(D)$.

$$v \in co(D') \implies \forall s' \in S' : v(s') = \Gamma_{D'}(v)(s') \tag{78}$$
$$\implies \forall s' \in S' : v(s') = \bigsqcap\{w(C'_{s'}) \mid w \in [v]_2\} \tag{79}$$
$$\implies \forall s \in S : v(s) = \bigsqcap\{w(C'_s) \mid w \in [v]_2\} \tag{80}$$
$$\implies \forall s \in S : v(s) = \bigsqcap\{w(C_s) \mid w \in [v]_2\} \tag{81}$$
$$\implies \forall s \in S : v{\downarrow_S}(s) = \bigsqcap\{w(C_s) \mid w \in [v{\downarrow_S}]_2\} \tag{82}$$
$$\implies \forall s \in S : v{\downarrow_S}(s) = \Gamma_D(v{\downarrow_S})(s) \tag{83}$$
$$\implies v{\downarrow_S} \in co(D) \tag{84}$$

(78) Follows from Definition 6.
(79) Follows from Definition 5.
(80) Follows from $S \subset S'$.
(81) Let $s \in S$. If $s \neq x$, then by Definition 9, the acceptance condition remains unchanged. Hence $\forall w \in [v]_2 : w(C_s) = w(C'_s)$ and therefore $\bigsqcap\{w(C_s) \mid w \in [v]_2\} = \bigsqcap\{w(C'_s) \mid w \in [v]_2\}$.

If $s = x$, then we show $v(x) = \bigsqcap\{w(C_x) \mid w \in [v]_2\}$ by two cases:

1. $C'_x = \phi \vee \top$, then $\phi \vee \psi$ is a tautology. Therefore $\forall w \in [v']_2 : w(\phi \vee \psi) = \mathbf{t}$ and $w(\phi \vee \top) = \mathbf{t}$. From this it follows that $\bigsqcap\{w(C_x) \mid w \in [v']_2\} = \bigsqcap\{w(C'_x) \mid w \in [v']_2\}$.

2. $C'_x = \phi \vee x^*$, then $\phi \vee \psi$ is not a tautology and we differentiate three cases:
   (a) $v(x) = \mathbf{t}$, then $\bigsqcap\{w(C'_x) \mid w \in [v]_2\} = \mathbf{t}$ and $\forall w \in [v]_2 : w(C'_x) = w(\phi \vee x^*) = \mathbf{t}$. We show that $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{t}$ by three cases:
      i. $\forall w \in [v]_2 : w(x^*) = \mathbf{t}$, then $\bigsqcap\{w(C'_{x^*}) \mid w \in [v]_2\} = \mathbf{t}$ and $\forall w \in [v]_2 : w(C'_{x^*}) = w(\psi) = \mathbf{t}$. From this it follows that $\forall w \in [v]_2 : w(\phi \vee \psi) = w(C_x) = \mathbf{t}$ and thus $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{t}$.

      ii. $\forall w \in [v]_2 : w(x^*) = \mathbf{f}$, then $\forall w \in [v]_2 : w(\phi) = \mathbf{t}$, in order to always valuate the conjunction to $\mathbf{t}$. From this it follows that $\forall w \in [v]_2 : w(\phi \vee \psi) = w(C_x) = \mathbf{t}$ and thus $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{t}$.

      iii. $\exists w \in [v]_2 : w(x^*) = \mathbf{t}$ and $\exists w \in [v]_2 : w(x^*) = \mathbf{f}$, then $\forall w \in [v]_2 : w(\phi) = \mathbf{t}$, in order to always valuate the disjunction to $\mathbf{t}$. From this it follows that $\forall w \in [v]_2 : w(\phi \vee \psi) = w(C_x) = \mathbf{t}$ and thus $\bigsqcap\{w(C_x) \mid w \in [v']_2\} = \mathbf{t}$.
   Hence whenever $v(x) = \mathbf{t}$ it holds in all cases that $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{t}$.

   (b) $v(x) = \mathbf{f}$, then $\bigsqcap\{w(C'_x) \mid w \in [v]_2\} = \mathbf{f}$ and $\forall w \in [v]_2 : w(C'_x) = w(\phi \vee x^*) = \mathbf{f}$. From this it follows that $\forall w \in [v]_2 : w(\phi) = \mathbf{f}$ and $w(x^*) = \mathbf{f}$ and thus $v(x^*) = \mathbf{f}$. Hence $\bigsqcap\{w(C'_{x^*}) \mid w \in [v]_2\} = \mathbf{f}$ and $\forall w \in [v]_2 : w(C'_{x^*}) = w(\psi) = \mathbf{f}$. Therefore $\forall w \in [v]_2 : w(\phi \vee \psi) = w(C_x) = \mathbf{f}$ and thus $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \bigsqcap\{w(C'_x) \mid w \in [v]_2\} = \mathbf{f}$.

   (c) $v(x) = \mathbf{u}$, then $\bigsqcap\{w(C'_x) \mid w \in [v]_2\} = \mathbf{u}$ and $\exists w \in [v]_2 : w(\phi \vee x^*) = \mathbf{t}$ and $\exists w \in [v]_2 : w(\phi \vee x^*) = \mathbf{f}$. We prove $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{u}$ by three cases:

i. $\forall w \in [v]_2 : w(x^*) = \mathbf{t}$. This case is *impossible* as there exists a w such that $w(\phi \vee x^*) = \mathbf{f}$.

ii. $\forall w \in [v]_2 : w(x^*) = \mathbf{f}$, then $\bigsqcap\{w(C'_{x^*}) \mid w \in [v]_2\} = \mathbf{f}$ and $\forall w \in [v]_2 : w(\psi) = \mathbf{f}$. From this it follows that $\forall w \in [v]_2 : w(x^*) = w(\psi) = \mathbf{f}$ and thus $\exists w \in [v]_2 : w(\phi \vee \psi) = \mathbf{t}$ and $\exists w \in [v]_2 : w(\phi \vee \psi) = \mathbf{f}$. Therefore $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{u}$.

iii. $\exists w \in [v]_2 : w(x^*) = \mathbf{t}$ and $\exists w \in [v]_2 : w(x^*) = \mathbf{f}$, then $v(x^*) = \mathbf{u}$ and $\bigsqcap\{w(C_{x^*}) \mid w \in [v]_2\} = \mathbf{u}$. From this it follows that $\exists w \in [v]_2 : w(\psi) = \mathbf{t}$ and $\exists w \in [v]_2 : w(\psi) = \mathbf{f}$. We have to examine three more cases:

A. $\forall w \in [v]_2 : w(\phi \vee \psi) = \mathbf{t}$. This case is *impossible* as there always exists a w such that $w(\phi \vee \psi) = \mathbf{f}$.

B. $\forall w \in [v]_2 : w(\phi \vee \psi) = \mathbf{f}$. This case is *impossible* as there always exists a w such that $w(\psi) = \mathbf{t}$.

C. $\exists w \in [v]_2 : w(\phi \vee \psi) = \mathbf{t}$ and $\exists w \in [v]_2 : w(\phi \vee \psi) = \mathbf{f}$, then $\bigsqcap\{w(C_x) \mid w \in [v]_2\} = \mathbf{u}$.

Hence whenever $v(x) = \mathbf{u}$ it holds in all cases that $\bigsqcap\{w(C_s) \mid w \in [v]_2\} = \mathbf{u}$.

(82) By Definition 2 it holds that $\forall s \in S : v\downarrow_S(s) = v(s)$.

(83) Follows from Definition 5.

(84) Follows from Definition 6.

Theorem. Let $D = (S, C)$ be an ADF and $D' = (S', C')$ its $\vee$-decomposition at x. For any semantics $\sigma \in \{ad, pr, co, gr\}$: $\sigma(D) = \sigma(D')\downarrow_S$.

Proof. The proof follows the same structure as the proof of the theorem for $\neg$-decomposition, with Lemma 11 - Lemma 14 taking the place of Lemma 1 - Lemma 4 respectively.

Appendix B: Full Conjunction Decomposition

Definition. *Let $D = (S, C)$ be and ADF. Let $x \in S$ with $C_x = \phi \wedge \psi$, where ϕ and ψ are some propositional statements. Then the* full conjunction decomposition at *x of D is the ADF $D' = (S', C')$, where $S' = S \cup \{x^*, x^{**}\}$ and C' is defined by*

- $C'_y = C_y$ *if* $y \notin \{x, x^*, x^{**}\}$
- $C'_x = \bot \wedge \bot$ *if $\phi \wedge \psi$ is a logical contradiction and $C'_x = x^* \wedge x^{**}$ otherwise*
- $C'_{x^*} = \phi$
- $C'_{x^{**}} = \psi$

Theorem. *Let $D = (S, C)$ be an ADF and $D^f = (S^f, C^f)$ its* full conjunction decomposition at *x. Let $D' = (S', C')$ be the* conjunction decomposition at *x of D and let $D'' = (S'', C'')$ be the* conjunction decomposition at *x of D'. Then $D'' = D^f$.*

Proof. Let $D = (S, C)$ be an ADF with $x \in S$ such that $C_x = \phi \wedge \psi$, where ϕ and ψ are some propositional statements. Let $D^f = (S^f, C^f)$ be the *full conjunction decomposition at x* of D such that:

- $S^f = S \cup \{x^*, x^{**}\}$
- $C^f_y = C_y$ if $y \notin \{x, x^*, x^{**}\}$
- $C^f_x = \bot \wedge \bot$ if $\phi \wedge \psi$ is a logical contradiction and $C^f_x = x^* \wedge x^{**}$ otherwise

- $C^f_{x^*} = \phi$
- $C^f_{x^{**}} = \psi$

Let $D' = (S', C')$ be the *partial conjunction decomposition at* x of D, such that:

- $S' = S \cup \{x^*\}$
- $C'_y = C_y$ if $y \notin \{x, x^*\}$
- $C'_x = \bot \wedge \psi$ if $\phi \wedge \psi$ is a logical contradiction and $C'_x = x^* \wedge \psi$ otherwise
- $C'_{x^*} = \phi$

Let $D'' = (S'', C'')$ be the *partial conjunction decomposition at* x of D', such that:

- $S'' = S' \cup \{x^{**}\}$
- $C''_y = C'_y$ if $y \notin \{x, x^{**}\}$
- $C''_x = \bot \wedge \bot$ if $\phi \wedge \psi$ is a logical contradiction and $C'_x = x^* \wedge x^{**}$ otherwise
- $C''_{x^{**}} = \psi$

From this it follows that:

- $S'' = S' \cup \{x^{**}\} = \{S \cup \{x^*\}\} \cup \{x^{**}\} = S \cup \{x^*, x^{**}\} = S^f$
- $C''_x = \bot \wedge \bot = C^f_x$ if $\phi \wedge \psi$ is a logical contradiction and $C''_x = x^* \wedge x^{**} = C^f_x$ otherwise
- $C''_{x^*} = C'_{x^*} = \phi = C^f_{x^*}$
- $C''_{x^{**}} = \psi = C^f_{x^{**}}$
- $C''_y = C'_y = C_y = C^f_y$ for all $y \in S \setminus \{x, x^*, x^{**}\}$

Hence $S'' = S^f$ and $C'' = C^f$ and thus $D'' = D^f$.