

An Abstract, Logical Approach to Characterizing Strong Equivalence in Non-monotonic Knowledge Representation Formalisms

Ringo Baumann

*Computer Science Institute, Leipzig University
Augustusplatz 10, 04109 Leipzig, Germany*

Hannes Strass

*Computational Logic Group
Institute of Artificial Intelligence
Faculty of Computer Science
TU Dresden, Germany*

Abstract

Two knowledge bases are strongly equivalent if and only if they are mutually interchangeable in arbitrary contexts. This notion is of high interest for any logical formalism, since it allows to locally replace parts of a given theory without changing its meaning. In contrast to classical logic, where strong equivalence coincides with standard equivalence (having the same models), it is possible to find ordinary but not strongly equivalent objects for any nonmonotonic formalism available in the literature. Consequently, much effort has been devoted to characterizing strong equivalence for knowledge representation formalisms such as logic programs under the stable model semantics, Reiter's default logic, or Dung's argumentation frameworks. For example, strong equivalence for logic programs under stable models can be characterized by so-called HT-models. More precisely, two logic programs are strongly equivalent if and only if they are standard equivalent in the logic of here and there. This means, the logic of here and there can be seen as a characterizing formalism for logic programs under stable model semantics.

The aim of this article is to study whether the existence of such characterization logics can be guaranteed for any logic. One main result is that every knowledge representation formalism that allows for a notion of strong equivalence on its finite knowledge bases also possesses a canonical characterizing formalism. In particular, we argue that those characterizing formalisms can be seen as classical, monotonic logics. Moreover, we will not only show the existence of characterizing formalism, but even that the model theory of any characterizing logic is uniquely determined (up to isomorphism).

Keywords: strong equivalence, knowledge representation formalisms, logic

1. Introduction

Reusability of human-made artifacts is of paramount importance in computer science. For example, the scale and depth of contemporary software development would not be possible without the availability of software *libraries*, which encapsulate certain functionalities that can then be used in different parts of a project and also across multiple projects. Crucially, libraries

make themselves use of libraries for more basic features, all the way down to the core of the programming language being employed in each specific case. We are convinced that any large-scale knowledge representation project where knowledge is engineered will likewise have to make use of a significant amount of modularity – not only to manage complexity, but also to avoid having to start from scratch in each iteration of the engineering process. To assess the reusability of (parts of) knowledge bases in logic-based knowledge representation languages, in turn, we have to know whether pieces of knowledge make certain context-dependent assumptions. In classical propositional logic, for example, there is no such context-dependence: whenever two (sets of) formulas are equivalent in the sense of having the same models, then they are mutually replaceable in arbitrary contexts.

In the field of knowledge representation and reasoning, however, not all commonly used formalisms are like classical logic in this regard. For example, the answer-set-programming paradigm uses (various extensions of) the formalism of normal logic programs to encode combinatorial problems such that the answer sets (stable models) of the logic programs correspond to and encode solutions to the encoded problem [1]. Alas, for normal logic programs, having the same stable models does *not* amount to mutual replaceability. For this, a stronger property is needed: it is called *strong equivalence*, and holds for two logic programs if and only if they keep the same stable models even if they are both extended with an arbitrary third logic program. Formally – as a logic program is a set of rules –, extending a logic program can be modeled by ordinary set union. Consequently, the notion of strong equivalence can be defined in a similar way for other knowledge representation formalisms for which set union is an adequate formalization of appending or otherwise combining knowledge bases.

As a grossly simplified motivating example, consider building a knowledge base of travel destinations across the world, represented as nodes of a graph, and information of possibilities for one-leg trips between destinations, represented as edges of the graph. In that setting, it seems fruitful to use a library that provides some functionalities commonly associated with directed graphs, for example for declaratively defining reachability, querying for shortest paths and similar scenarios. It is customary (in logic programming) to represent directed graphs as sets of ground-instantiated first-order atoms – for example, the graph $G = (\{a, b, c\}, \{(a, b), (c, b)\})$ could be represented by the logic program $\{n(a), n(b), n(c), e(a, b), e(c, b)\}$ making use of predicates for nodes (n) and edges (e). A graph library could then build upon the vocabulary $\{n/1, e/2\}$ (denoting the predicate names along with their arity) and offer additional predicates, e.g. $p/2$ whenever there is a (directed) path between two nodes in the graph. A straightforward “implementation” of reachability in (first-order) answer set programming is given by the rules $p(X, Y) \leftarrow e(X, Y)$ and $p(X, Z) \leftarrow p(X, Y), p(Y, Z)$, where upper-case letters are first-order variables and rules containing variables are understood as collections of their ground instantiations. Denoting this “library” by T_{graph} , it might at some point become necessary/worthwhile to replace T_{graph} by a, say, more optimized version providing the same functionality over that vocabulary, denoted T_{groppt} . Then it is vital to answer the question whether we can safely replace T_{graph} by T_{groppt} in our overall knowledge base. Providing solutions to such scenarios falls squarely into the realm of strong equivalence.

In a series of interesting developments, researchers have succeeded in precisely characterizing strong equivalence for several common knowledge representation formalisms, among them normal logic programs under the stable model semantics [2, 3].¹ What is more, it turned out that

¹Due to their widespread use, for logic programs there also exist further, intermediate equivalence notions: *uniform* equivalence [4], where extension is restricted to a set of facts; *modular* equivalence [5], which considers decomposing a logic program (into modules) along the strongly connected components of the logic programs’s positive dependency graph; and even further, parameterized versions [6].

sometimes, to characterize strong equivalence in formalism \mathcal{F} , we can use ordinary equivalence in a formalism \mathcal{F}' : for example, strong equivalence in normal logic programs under stable models can be characterized by the standard semantics of the logic of here-and-there [2]. Such results have rightly been recognized as important for the study of these concrete knowledge representation formalisms, as having a characterization of strong equivalence gives us a deeper level of understanding of the meaning of pieces of knowledge in that formalism.

However, such results about the existence of characterizing formalisms also raise a fundamental question: *Does every formalism have one?* In this paper, we answer this question with a qualified “yes”. More precisely, while not every formalism has one, we show that the important case of considering only *finite* knowledge bases (but still possibly infinite languages) guarantees the existence of a characterizing formalism, and that in a very general setting. Existing results on characterizing formalisms make use of specifics of each formalism [2, 3, 7, 8, 9]. In this paper, we completely abstract away from formalism specifics and address the core of the problem, the nature of strong equivalence itself. In fact, we will not only show the existence of just any characterizing formalism, but of characterizing formalisms whose model theory is *uniquely determined* (up to isomorphism), and structurally resembles that of classical logics. At this point, we appeal to the reader’s intuition on what makes logics classical; we will later define what we mean by “classical logic” in a precise mathematical way (and along the lines of previous work towards this end [10, 11]). Still, we consider this main result of our paper a surprising and important insight, as it tells us that for the overwhelming majority of knowledge representation formalisms, strong equivalence can be approached using established techniques from classical logic.

While our work is in its essence derived from first principles, building mostly upon classical logic and lattice theory, there have been important inspirations. Foremost, Truszczyński presented a general, algebraic account of strong equivalence [12] within approximation fixpoint theory [13]. His setting is indeed quite general, but most of this generality derives from algebraic commonalities in the semantics of logic programs and default logic. It is not immediately clear, for example, if and how it captures Dung’s abstract argumentation frameworks (AFs) [14], another important AI formalism whose strong equivalence has been studied in the recent past [15, 16]. More precisely, while AFs with all their semantics can be captured by approximation fixpoint theory [17], Truszczyński’s notion of expanding an operator does not coincide with the corresponding notion of expanding AFs and his results are not directly applicable.² In other words, while the operator associated to the union of two logic programs corresponds to the union of their respective associated operators, the same does not hold for the union of two AFs and their operators. Thus although AFs are essentially a restricted subclass of normal logic programs with respect to the ordinary equivalence of having the same models, this does not carry over to strong equivalence because the respective notions of knowledge base union are different in AFs and normal logic programs. For example, the AF where b attacks a corresponds to the logic program $P_1 = \{a \leftarrow \sim b\}$; likewise, $P_2 = \{a \leftarrow \sim c\}$ corresponds to the AF where c attacks a [18, 19, 17]. However, the AF where both b and c attack a (the union of the two AFs above) corresponds to the logic program $P_3 = \{a \leftarrow \sim b, \sim c\}$, where obviously the three programs are not subset-related: $P_1 \not\subseteq P_3$ and $P_2 \not\subseteq P_3$.

In contrast, we show how the approach we develop in this paper can be directly applied to argumentation frameworks. Thus as a consequence of our main theorem, we get the first semantical characterization of strong equivalence in AFs. This is significant because the only currently known characterizations are syntactical [15].

²As an AF is a pair (A, R) with A a set of arguments and $R \subseteq A \times A$ an attack relation, the union of AFs is defined component-wise: $(A_1, R_1) \dot{\cup} (A_2, R_2) = (A_1 \cup A_2, R_1 \cup R_2)$.

Due to its general, abstract setting, our approach is able to deal with cases such as infinite knowledge bases off the shelf. While it may seem counter-intuitive to consider infinite syntactical objects in knowledge representation at all, infinite sets of logic programming rules arise naturally due to ground instantiation, a typical method for obtaining the semantics of rules with variables in a first-order setting. In answer set programming, in order to obtain a formal semantics for solvers that are already being used in practice, Harrison et al. [20, 21] have even extended propositional logic by *infinite* conjunctions and disjunctions. This shows that considering infinite syntactical objects is not only beneficial, but sometimes even necessary when analyzing aspects of knowledge representation languages. A dedicated analysis of strong equivalence in this infinitary-logic setting was later given by Harrison et al. [22], while clearly our understanding of “logic” and thus the results of this paper also apply to the case of possibly infinite sets with possibly infinitary formulas. Although the bearing of our theory on questions regarding specific formalisms will be not as strong as that of more specialized theories, our main results still tell researchers looking for a “nice” characterization of strong equivalence in their formalism that the search – given the right conditions are met – will not be in vain.

The paper proceeds as follows. In the next section, we introduce the general setting in which we derive our results and present our conception of the term “classical logic”. Afterwards, in the main part of the paper, we define characterization logics and show two classes of formalisms that always possess them. We next apply our results, chiefly to abstract argumentation frameworks but briefly also to normal logic programs. The paper concludes with a discussion of related and future work.

2. An Abstract View on Model Theory

What is a classical logic?

We will spend this section introducing an abstract notion of logics with model-theoretic semantics and explaining when we call some of them classical. Formally, we consider logical languages \mathcal{L} , that is, non-empty sets of language elements. We make no assumption on the internal structure of pieces of knowledge $F \in \mathcal{L}$. These pieces of knowledge could be formulas of classical propositional logic, normal logic program rules, attacks between arguments, or Reiter-style defaults. A model-theoretic semantics for a language \mathcal{L} uses a set \mathcal{I} of interpretations and a *model* function $\sigma : 2^{\mathcal{L}} \rightarrow 2^{\mathcal{I}}$ with the intuition that σ assigns each language subset $T \subseteq \mathcal{L}$, a *theory*, the set $\sigma(T) \subseteq \mathcal{I}$ of its models. We make no assumptions on the internal structure of interpretations – there need not be an underlying vocabulary of atoms or the like (although in the concrete cases we consider there often will be) that are the same among syntax and semantics. This is the main abstraction in our setting. It goes beyond what is known from classical logic in that meaning is not assigned to language elements (formulas), but only to *theories*, that is, sets of language elements. This is a necessary requirement for being able to model a number of established knowledge representation formalisms: for example, in normal logic programs, meaning is not assigned to single rules, but only to sets thereof. Likewise, in default logic, meaning is not assigned to single defaults, but only to sets thereof.³ We illustrate our definitions so far by showing more precisely how existing formalisms can be embedded into our setting.

Example 1. Consider a set $\mathcal{A} = \{a_1, a_2, a_3, \dots\}$ representing the most basic building blocks of the following formalisms. For instance, in case of propositional logic \mathcal{A} stands for atomic formulas while in the context of abstract argumentation \mathcal{A} represents the set of all arguments.

³In this reading, we interpret the elements of the world knowledge as justification-free defaults with tautological prerequisite.

Classical propositional logic: The underlying language \mathcal{L}_{PL} is the set of all classical propositional formulas over \mathcal{A} and can be defined as usual by induction. The set of interpretations is then given by the set $\mathcal{I}_{PL} = \{v : \mathcal{A} \rightarrow \{\mathbf{t}, \mathbf{f}\}\}$ of all two-valued interpretations of \mathcal{A} . Lastly, $\sigma_{mod}(T)$ is the set of all models of the theory $T \subseteq \mathcal{L}_{PL}$, that is, the set of all interpretations satisfying all formulas in T .

Normal logic programs: The underlying language \mathcal{L}_{LP} is the set of all normal logic program rules $a_0 \leftarrow a_1, \dots, a_m, \sim a_{m+1}, \dots, \sim a_n$ with $0 \leq m \leq n$ and $a_0, a_1, \dots, a_n \in \mathcal{A}$. The set \mathcal{I}_{LP} of interpretations is then the set $\mathcal{I}_{LP} = 2^{\mathcal{A}}$ of all possible stable model candidates. Accordingly, $\sigma_{stb}(T)$ returns the set of stable models of the theory (normal logic program) $T \subseteq \mathcal{L}_{LP}$ [23]; we could also define $\sigma_{sup}(T)$ returning the set of supported models of T [24], or $\sigma_{mod}(T)$ the set of classical models of T , interpreting \leftarrow as material implication and \sim as classical negation [25].

Propositional default logic [26]: The underlying language \mathcal{L}_{DL} is the set of all defaults over formulas over \mathcal{A} , $\mathcal{L}_{DL} = \{(\varphi, \Psi, \xi) \mid \varphi, \xi \in \mathcal{L}_{PL}, \Psi \subseteq \mathcal{L}_{PL}\}$ where the triple $(\varphi, \{\psi_1, \dots, \psi_n\}, \xi)$ represents the default $\varphi : \psi_1, \dots, \psi_n / \xi$. The possible interpretations are given by $\mathcal{I}_{DL} = 2^{\mathcal{L}_{PL}}$, as each default theory is assigned a set of logical theories called extensions: $\sigma_{DL}(T) = \{E \subseteq \mathcal{L}_{PL} \mid E \text{ is an extension of } T\}$.

Abstract argumentation frameworks [14]: The underlying language \mathcal{L}_{AF} contains the fundamental building blocks of AFs, that is, arguments and attacks:

$$\mathcal{L}_{AF} = \{(\{a\}, \emptyset), (\{a, b\}, \{(a, b)\}) \mid a, b \in \mathcal{A}\}$$

Extension-based semantics can be incorporated by setting $\mathcal{I}_F = 2^{\mathcal{A}}$ and, depending on the argumentation semantics ρ we use, we set $\sigma_\rho(T) = \rho(F_T)$, where the tuple $F_T = (\bigcup_{(A,R) \in T} A, \bigcup_{(A,R) \in T} R)$ is the AF associated to $T \subseteq \mathcal{L}_{AF}$ (cf. Section 4.1 for more detailed information and discussion). \diamond

Before we delve into the main aim of this paper, characterizing strong equivalence, we briefly analyze some foundational properties of our way of abstractly modeling knowledge representation languages. We begin with the relationship between the model function and the consequence function of a logical language.

2.1. Models and Consequences

A consequence function for a language \mathcal{L} is a function $Cn : 2^{\mathcal{L}} \rightarrow 2^{\mathcal{L}}$ that assigns a given set T of language elements another set $Cn(T)$ of language elements. Intuitively, $Cn(T)$ is understood to be the set of logical consequences of the theory T . Given a language, we can define the consequence function in terms of the semantics. In words, the set of consequences of a given theory T is the union of all theories S such that any model of T is a model of S .

Definition 1. Let \mathcal{L} be a language and $\sigma : 2^{\mathcal{L}} \rightarrow 2^{\mathcal{I}}$ be a model function. The *canonical consequence function* of σ is defined as follows:

$$Cn^\sigma : 2^{\mathcal{L}} \rightarrow 2^{\mathcal{L}}, \quad T \mapsto \bigcup_{\substack{S \subseteq \mathcal{L}, \\ \sigma(T) \subseteq \sigma(S)}} S \quad \diamond$$

In classical definitions of logical consequence, one typically says that a single formula is a consequence of a theory if all models of the theory are models of the formula. In our case, the

focus is primarily on theories both for presumptions and consequences, so we stick to the above definition. For classical logic \mathcal{L}_{PL} , this definition coincides with the standard notion of logical consequence.

It will be of great interest in this paper that certain algebraic properties of the semantics induce certain useful properties of the consequence relation. We now introduce the most important properties.⁴

Definition 2. Let \mathcal{L} be a language.

- A model function $\sigma : 2^{\mathcal{L}} \rightarrow 2^{\mathcal{I}}$ is *antimonotone* iff
for all $T_1, T_2 \in 2^{\mathcal{L}}: T_1 \subseteq T_2 \implies \sigma(T_2) \subseteq \sigma(T_1)$.
- A consequence operator $Cn : 2^{\mathcal{L}} \rightarrow 2^{\mathcal{L}}$ is *monotone* iff
for all $T_1, T_2 \in 2^{\mathcal{L}}: T_1 \subseteq T_2 \implies Cn(T_1) \subseteq Cn(T_2)$.
- A consequence operator $Cn : 2^{\mathcal{L}} \rightarrow 2^{\mathcal{L}}$ is *increasing* iff
for all $T \in 2^{\mathcal{L}}$, we find $T \subseteq Cn(T)$.
- A consequence operator $Cn : 2^{\mathcal{L}} \rightarrow 2^{\mathcal{L}}$ is *idempotent* iff
for all $T \in 2^{\mathcal{L}}$, we find $Cn(Cn(T)) \subseteq Cn(T)$.
- A consequence operator $Cn : 2^{\mathcal{L}} \rightarrow 2^{\mathcal{L}}$ is a *closure operator* iff
 Cn is monotone, increasing, and idempotent. ◇

To save some space in what follows, we define a *logic* as a tuple $(\mathcal{L}, \mathcal{I}, \sigma)$ consisting of a language \mathcal{L} , an interpretation set \mathcal{I} , and a model function $\sigma : 2^{\mathcal{L}} \rightarrow 2^{\mathcal{I}}$. We will sometimes associate the canonical consequence function of σ to the whole logic for convenience.

To start out in analyzing how semantics and consequence relate in our setting, we show some straightforward properties of the canonical consequence function defined above. More precisely, any induced consequence operator is increasing and moreover, it is monotone if the considered model function is antimonotone.

Proposition 1. Let $(\mathcal{L}, \mathcal{I}, \sigma)$ be a logic and Cn^σ its canonical consequence function.

1. Cn^σ is increasing.
2. If σ is antimonotone, then Cn^σ is monotone.

Proof.

1. Clearly for any $T \subseteq \mathcal{L}$ we find $\sigma(T) \subseteq \sigma(T)$. Consequently, we deduce $T \subseteq \bigcup_{\substack{S \subseteq \mathcal{L}, \\ \sigma(T) \subseteq \sigma(S)}} S = Cn^\sigma(T)$.
2. Let σ be antimonotone and $T_1 \subseteq T_2$. Then clearly $\sigma(T_2) \subseteq \sigma(T_1)$ by antimonotonicity. Thus for any $S \subseteq \mathcal{L}$ with $\sigma(T_1) \subseteq \sigma(S)$ we also find $\sigma(T_2) \subseteq \sigma(S)$. It follows that $\bigcup_{S \subseteq \mathcal{L}, \sigma(T_1) \subseteq \sigma(S)} S \subseteq \bigcup_{S \subseteq \mathcal{L}, \sigma(T_2) \subseteq \sigma(S)} S$ that is, $Cn^\sigma(T_1) \subseteq Cn^\sigma(T_2)$. Since T_1 and T_2 were arbitrary, Cn^σ is monotone.

⁴We could introduce these properties in an even more abstract setting of operators on partially ordered sets, but refrain from doing so lest we introduce even more notation.

The reverse of Item 2 does not hold, that is, monotone consequence functions do not necessitate antimonotone model functions.

Example 2. Consider $\mathcal{L} = \{a\}$ and $\mathcal{I} = \{1\}$ with $\sigma(\emptyset) = \emptyset$ and $\sigma(\{a\}) = \{1\}$. We get the following canonical consequence function:

$$\begin{aligned} Cn^\sigma(\emptyset) &= \bigcup_{\substack{S \subseteq \mathcal{L}, \\ \sigma(\emptyset) \subseteq \sigma(S)}} S = \bigcup_{\substack{S \subseteq \mathcal{L}, \\ \emptyset \subseteq \sigma(S)}} S = \bigcup_{S \subseteq \mathcal{L}} S = \mathcal{L} = \{a\} \\ Cn^\sigma(\{a\}) &= \bigcup_{\substack{S \subseteq \mathcal{L}, \\ \sigma(\{a\}) \subseteq \sigma(S)}} S = \bigcup_{\substack{S \subseteq \mathcal{L}, \\ \{1\} \subseteq \sigma(S)}} S = \{a\} \end{aligned}$$

Thus $Cn^\sigma(\emptyset) = \{a\} = Cn^\sigma(\{a\})$ and Cn^σ is monotone and increasing albeit σ is not antimonotone. \diamond

Also, not every antimonotone model function induces a closure operator, that is, an operator that is monotone, increasing and idempotent.

Example 3. Consider the logic $(\mathcal{L}, \mathcal{I}, \sigma)$ with language $\mathcal{L} = \{a, b, c\}$, interpretation set $\mathcal{I} = \{1\}$ and model function $\sigma : 2^\mathcal{L} \rightarrow 2^\mathcal{I}$ given by

$$\sigma(T) = \begin{cases} \{1\} & \text{if } T \in \{\emptyset, \{a\}, \{b\}\} \\ \emptyset & \text{otherwise} \end{cases}$$

In addition to σ being antimonotone we have $Cn^\sigma(\emptyset) = \{a, b\}$ and $Cn^\sigma(\{a, b\}) = \{a, b, c\}$. This implies

$$\{a, b, c\} = Cn^\sigma(\{a, b\}) = Cn^\sigma(Cn^\sigma(\emptyset)) \not\subseteq Cn^\sigma(\emptyset) = \{a, b\}$$

which shows that Cn^σ is not idempotent. \diamond

We can show that our restriction to semantics via model functions is not overly limiting. We could have chosen to start out from consequence functions as well, as long as these consequence functions are increasing (what is contained in a theory follows from it) and idempotent (all consequences are obtained in one step). More precisely, when given a consequence operator satisfying these two requirements, we can also define a canonical model function whose associated canonical consequence function is exactly the given consequence function we started with.

Proposition 2. *Let \mathcal{L} be a language and $C : 2^\mathcal{L} \rightarrow 2^\mathcal{L}$ be a consequence function that is increasing and idempotent.*

The model function $\sigma_C : 2^\mathcal{L} \rightarrow 2^\mathcal{L}$ with $T \mapsto \mathcal{L} \setminus C(T)$ is such that $Cn^{\sigma_C} = C$.⁵

⁵Note that we use $\mathcal{I} = \mathcal{L}$, that is, the assigned interpretation set of σ_C is just the language \mathcal{L} . This is purely for technical convenience.

Proof. Let $T \subseteq \mathcal{L}$. We find that

$$\begin{aligned}
Cn^{\sigma_C}(T) &= \bigcup_{\substack{S \subseteq \mathcal{L}, \\ \sigma_C(T) \subseteq \sigma_C(S)}} S && (\text{Def. } Cn^\sigma) \\
&= \bigcup_{\substack{S \subseteq \mathcal{L}, \\ \mathcal{L} \setminus C(T) \subseteq \mathcal{L} \setminus C(S)}} S && (\text{Def. } \sigma_C) \\
&= \bigcup_{\substack{S \subseteq \mathcal{L}, \\ C(S) \subseteq C(T)}} S && (\text{set algebra}) \\
&= \bigcup_{\substack{S \subseteq \mathcal{L}, \\ S \subseteq C(S) \subseteq C(T)}} S && (C \text{ is increasing})
\end{aligned}$$

Firstly, this shows that $Cn^{\sigma_C}(T) \subseteq C(T)$. Moreover, in the last equation we can substitute $S = C(T)$ to obtain that $C(T) \subseteq Cn^{\sigma_C}(T)$. \square

It is clear from Proposition 1 (Item 1) that no non-increasing consequence function C can be mimicked by Cn^σ for any σ . However, we consider the restrictions of possible consequence functions C having to be increasing and idempotent not to be too severe.

2.2. Standard and strong equivalence

This paper is chiefly about characterizing strong equivalence in one logic via standard equivalence in another logic. We will now formally introduce these concepts.

Definition 3. Let $(\mathcal{L}, \mathcal{I}, \sigma)$ be a logic and $T_1, T_2 \subseteq \mathcal{L}$ theories. We say that T_1 and T_2 are

- *ordinarily equivalent* if and only if $\sigma(T_1) = \sigma(T_2)$;
- *strongly equivalent* if and only if $\forall U \subseteq \mathcal{L} : \sigma(T_1 \cup U) = \sigma(T_2 \cup U)$. \diamond

The notion of strong equivalence is intimately connected with the possibility to simplify parts of a given theory without affecting its semantics. Consider the following example.

Example 4. Consider some logic $(\mathcal{L}, \mathcal{I}, \sigma)$, a theory $S \subseteq \mathcal{L}$ and a subtheory $T_1 \subseteq S$. Assume we have found T_2 , a “nicer” (shorter, more readable or beneficial in some other way) way of expressing T_1 . Whenever we know that T_1 and T_2 are not only ordinarily, but *strongly* equivalent, we may replace T_1 with T_2 in S (and denote the result as $S[T_1/T_2]$) and can be sure that the replacement does not affect the meaning of S , that is, S and $S[T_1/T_2]$ are ordinarily equivalent.

More precisely, $\sigma(S) = \sigma(S[T_1/T_2])$ with $S[T_1/T_2] = T_2 \cup (S \setminus T_1)$. This can be seen as follows: Since $T_1 \subseteq S$ we have $T_1 \cup (S \setminus T_1) = S$. Moreover, due to the strong equivalence of T_1 and T_2 we obtain $\sigma(T_1 \cup (S \setminus T_1)) = \sigma(T_2 \cup (S \setminus T_1))$. Hence, $\sigma(S) = \sigma(S[T_1/T_2])$ as claimed.

Clearly, strongly equivalent theories are ordinarily equivalent by definition. What about the converse direction? It is a matter of fact that in case of well-known nonmonotonic formalisms, such as logic programs [2], default logic [27], causal theories [28] and abstract argumentation [15, 16] strong equivalence and ordinary equivalence are indeed different concepts. However, there are logics like propositional logic or first order logic where both concepts coincide. In the following we will say that the model function σ has the *replacement property* if ordinary equivalence implies strong equivalence. The following natural question arises: What properties

must a logic possess in order for ordinary and strong equivalence to coincide? Propositional as well as first order logic possess a monotone consequence function. Does monotony of the consequence operator ensure the coincidence of both concepts? The following example provides us with a negative answer.

Example 5. Consider the language $\mathcal{L} = \{a, b\}$ with interpretation set $\mathcal{I} = \{1, 2\}$ and model function σ given by

$$\begin{aligned}\sigma(\emptyset) &= \{1, 2\} \\ \sigma(\{a\}) &= \{1, 2\} \\ \sigma(\{b\}) &= \{2\} \\ \sigma(\{a, b\}) &= \emptyset\end{aligned}$$

It is easy to verify that the semantics σ is antimonotone. Therefore, by Proposition 1, its consequence function Cn^σ is monotone. However, while \emptyset and $\{a\}$ are obviously ordinarily equivalent, they are not strongly equivalent, which can be seen by extending both with the theory $\{b\}$:

$$\sigma(\emptyset \cup \{b\}) = \sigma(\{b\}) = \{2\} \neq \emptyset = \sigma(\{a, b\}) = \sigma(\{a\} \cup \{b\})$$

We also inspect the induced consequence operator:

$$\begin{aligned}Cn^\sigma(\emptyset) &= \bigcup \{S \subseteq \mathcal{L} \mid \{1, 2\} \subseteq \sigma(S)\} = \{a\} \\ Cn^\sigma(\{a\}) &= \bigcup \{S \subseteq \mathcal{L} \mid \{1, 2\} \subseteq \sigma(S)\} = \{a\} \\ Cn^\sigma(\{b\}) &= \bigcup \{S \subseteq \mathcal{L} \mid \{2\} \subseteq \sigma(S)\} = \{a, b\} \\ Cn^\sigma(\{a, b\}) &= \bigcup \{S \subseteq \mathcal{L} \mid \emptyset \subseteq \sigma(S)\} = \{a, b\}\end{aligned}$$

Since the codomain of Cn^σ consists entirely of fixpoints, Cn^σ is idempotent. Therefore the induced consequence operator Cn^σ is increasing, monotone, and idempotent, thus a closure operator. Yet, the inducing semantics does not have the replacement property. \diamond

So having a monotone consequence function is, by itself, insufficient to guarantee the replacement property. We can however identify a property that is strong enough to guarantee replacement on its own. We call it the intersection property, because it basically says that the semantics of a theory can be obtained by only considering the semantics of the singleton sets constituting the theory.

Definition 4. Let $(\mathcal{L}, \mathcal{I}, \sigma)$ be a logic. Its model function $\sigma : 2^\mathcal{L} \rightarrow 2^\mathcal{I}$ has the *intersection property* iff for all $T \subseteq \mathcal{L}$:

$$\sigma(T) = \bigcap_{F \in T} \sigma(\{F\}) \quad \diamond$$

It follows from the definition that in particular for any two theories $T_1, T_2 \subseteq \mathcal{L}$, we have that $\sigma(T_1 \cup T_2) = \sigma(T_1) \cap \sigma(T_2)$. The intersection property is a certain locality, independence, or compositionality criterion.⁶ Towards an explanation of Example 5 we can now remark that its

⁶We will later consider a reformulation of this property that is easier to generalize to even more abstract settings, but for the moment this is the definition we work with.

model function σ does not have the intersection property:

$$\sigma(\{a, b\}) = \emptyset \neq \{2\} = \{1, 2\} \cap \{2\} = \sigma(\{a\}) \cap \sigma(\{b\})$$

Indeed, this is necessarily so: as we will show next (and as is easy to show), satisfying the intersection property is sufficient for satisfying the replacement property.

Proposition 3. *Let $(\mathcal{L}, \mathcal{I}, \sigma)$ be a logic. If σ satisfies the intersection property, then standard equivalence coincides with strong equivalence.*

Proof. Let σ satisfy the intersection property. It is clear that strong equivalence implies ordinary equivalence (set $U = \emptyset$), so it remains to show the converse. Let $T_1, T_2 \subseteq \mathcal{L}$ such that $\sigma(T_1) = \sigma(T_2)$ and consider any $U \subseteq \mathcal{L}$. We have

$$\begin{aligned} \sigma(T_1 \cup U) &= \sigma(T_1) \cap \sigma(U) && \text{(intersection)} \\ &= \sigma(T_2) \cap \sigma(U) && \text{(presumption)} \\ &= \sigma(T_2 \cup U) && \text{(intersection)} \quad \square \end{aligned}$$

Notably, monotonicity properties were not even needed in the above result. So why is it that all formalisms we know of that have the replacement property also happen to have monotone consequence functions? It holds because σ having the intersection property implies that σ is antimonotone (which in turn implies that Cn^σ is monotone).

Proposition 4. *Let $(\mathcal{L}, \mathcal{I}, \sigma)$ be a logic where σ has the intersection property. Then σ is antimonotone.*

Proof. Let $T_1 \subseteq T_2 \subseteq \mathcal{L}$. Then $T_1 \cup T_2 = T_2$, and we conclude the desired subset-inclusion via $\sigma(T_2) = \sigma(T_1 \cup T_2) = \sigma(T_1) \cap \sigma(T_2) \subseteq \sigma(T_1)$. \square

In the other direction, we can observe that the replacement property on its own does not guarantee antimonotonicity.

Example 6. Consider the language $\mathcal{L} = \{a\}$ and interpretation set $\mathcal{I} = \{1\}$. For semantics σ with $\sigma(\emptyset) = \emptyset$ and $\sigma(\{a\}) = \{1\}$, we can see that the replacement property holds trivially since there are no semantically equivalent theories that are syntactically different. However, σ is not antimonotone (as $\sigma(\{a\}) = \{1\} \not\subseteq \emptyset = \sigma(\emptyset)$) and does not have the intersection property:

$$\sigma(\emptyset \cup \{a\}) = \sigma(\{a\}) = \{1\} \neq \emptyset = \emptyset \cap \{1\} = \sigma(\emptyset) \cap \sigma(\{a\}) \quad \diamond$$

It is easy to see that classical propositional logic \mathcal{L}_{PL} has the intersection property simply *by definition*: the standard model semantics is typically firstly defined for single formulas $\varphi \in \mathcal{L}_{PL}$ and then generalized to theories T by *setting* $\sigma_{mod}(T) = \bigcap_{\varphi \in T} \sigma_{mod}(\{\varphi\})$.

As it turns out, for all logics, the intersection property also guarantees that each theory T has the same models as the set of all canonical consequences of T .

Proposition 5. *Let $(\mathcal{L}, \mathcal{I}, \sigma)$ be a logic that has the intersection property. Then for each $T \subseteq \mathcal{L}$ we find that $\sigma(T) = \sigma(Cn^\sigma(T))$.*

Proof.

$$\begin{aligned}
\sigma(\text{Cn}^\sigma(T)) &= \sigma\left(\bigcup_{\substack{S \subseteq \mathcal{L}, \\ \sigma(T) \subseteq \sigma(S)}} S\right) && \text{(Definition 1)} \\
&= \bigcap_{\substack{S \subseteq \mathcal{L}, \\ \sigma(T) \subseteq \sigma(S)}} \sigma(S) && \text{(intersection property)} \\
&= \sigma(T)
\end{aligned}$$

In the last equality, the \supseteq -direction is clear as we intersect only supersets of $\sigma(T)$, and the \subseteq -direction is clear as we can substitute T for S in the line above. \square

Finally, this means that the intersection property only holds for semantics whose canonical consequence functions are closure operators.

Proposition 6. *Let $(\mathcal{L}, \mathcal{I}, \sigma)$ be a logic that has the intersection property. Then Cn^σ is a closure operator.*

Proof. We have to show that Cn^σ is increasing, monotone and idempotent. First, Cn^σ is increasing in any case (Proposition 1). Moreover, since σ has the intersection property, it is also antimonotone (Proposition 4) and thus, Cn^σ is monotone (Proposition 1). Finally, it follows from Proposition 5 that Cn^σ is also idempotent. More precisely, if $\varphi \in \text{Cn}^\sigma(\text{Cn}^\sigma(T))$ then there is an $S \subseteq \mathcal{L}$ with $\varphi \in S$ and $\sigma(\text{Cn}^\sigma(T)) \subseteq \sigma(S)$, and by $\sigma(T) = \sigma(\text{Cn}^\sigma(T))$ it is clear that in this case $\sigma(T) \subseteq \sigma(S)$ and $\varphi \in S \subseteq \text{Cn}^\sigma(T)$. Hence, Cn^σ is a closure operator. \square

We have shown in Proposition 2 that we also could have started with a consequence function. Clearly the replacement property could be easily defined for consequence functions $C : 2^\mathcal{L} \rightarrow 2^\mathcal{L}$ in the sense that C has the replacement property if and only if for all theories $T_1, T_2 \subseteq \mathcal{L}$, “classical consequence-equivalence” $C(T_1) = C(T_2)$ coincides with “strong consequence-equivalence” $\forall U \subseteq \mathcal{L} : C(T_1 \cup U) = C(T_2 \cup U)$. It is also clear that any semantics having the replacement property induces a canonical consequence function having the consequence-function version of the replacement property. However, we must remark that we have not found a consequence-function equivalent of the intersection property. Even setting $C(T) = \bigcup_{F \in T} C(\{F\})$ would be too weak to capture interactions between different subtheories of T . On the other hand, and on the positive side, we will see that the intersection property for model functions as we defined it in Definition 4 will give us a good handle on characterizing strong equivalence.

2.3. Galois correspondences

Up to here, we have considered various properties of logics in our abstract setting. Mostly, those were algebraic properties of the model-theoretic semantics. In this subsection, we conclude our argument for defining as “classical” those logics whose semantics satisfy the intersection property.

For this, it is firstly necessary to slightly extend (and, for the time being, also slightly constrain) our abstract notion of “logic”. Up to now, we only assumed the existence of a model function $\sigma : 2^\mathcal{L} \rightarrow 2^\mathcal{I}$ that assigns a set of interpretations to a theory (intuitively, its models). In the converse direction, we now also assume a *theory function* $\tau : 2^\mathcal{I} \rightarrow 2^\mathcal{L}$, that takes as input a set $K \subseteq \mathcal{I}$ of interpretations and intuitively returns the set $\tau(K)$ of all language elements that

are true under all interpretations in K . A *logic* will now be a tuple $(\mathcal{L}, \mathcal{I}, \sigma, \tau)$ with $\mathcal{L}, \mathcal{I}, \sigma$ as before and $\tau : 2^{\mathcal{I}} \rightarrow 2^{\mathcal{L}}$ a theory function.

This immediately yields another way of defining a consequence operator: given σ and τ , we can define $Cn^{\sigma, \tau}(T) = \tau(\sigma(T))$. Symmetrically, we can define an operator on interpretation sets by $K \mapsto \sigma(\tau(K))$.

We next consider a class of logics where the interplay of model function and theory function satisfies certain conditions. Below, we denote the composition of functions $f : A \rightarrow B$ and $g : B \rightarrow C$ by $g \circ f$, that is, $g \circ f : A \rightarrow C$ with $x \mapsto g(f(x))$.

Definition 5. Let $(\mathcal{L}, \mathcal{I}, \sigma, \tau)$ be a logic. The functions σ and τ are in *Galois correspondence* if and only if:

1. both σ and τ are antimonotone;
2. both $\sigma \circ \tau$ and $\tau \circ \sigma$ are increasing.

If σ and τ are in Galois correspondence, we also say that $(\mathcal{L}, \mathcal{I}, \sigma, \tau)$ is a *Galois logic*.

Galois correspondences have mainly been studied in lattice theory [29, 30, 31, 32]. Further work on them was done in model theory by Goguen and Burstall [10], who also aim at formalizing the intuitive notion of a “logical system”, which they do by introducing an abstract notion called *institution* making use of category theory. Their notion of “logic” has a more distinct first-order flavor, as they include the syntactical notion of signatures in the definition, but they observe that their definition of *theories* of an institution (Def. 19) induces a Galois connection and moreover leads to a semantics satisfying the intersection property (Prop. 21, Item 8).

As our first technical remark, we observe that the antimonotonicity of σ and τ imply that the resulting consequence function is monotone; by the last property, it is also increasing. It follows that Galois correspondences induce closure operators, that is, operators that are monotone, increasing, and idempotent.

Proposition 7. Let $(\mathcal{L}, \mathcal{I}, \sigma, \tau)$ be a Galois logic. Then the operators

- $\sigma \circ \tau : 2^{\mathcal{I}} \rightarrow 2^{\mathcal{I}}, K \mapsto \sigma(\tau(K))$
- $\tau \circ \sigma : 2^{\mathcal{L}} \rightarrow 2^{\mathcal{L}}, T \mapsto \tau(\sigma(T))$

are closure operators.

Proof. We have to show that $\sigma \circ \tau$ and $\tau \circ \sigma$ are (1) increasing, (2) monotone and (3) idempotent. We only consider $\sigma \circ \tau$ as the proof for $\tau \circ \sigma$ is absolutely symmetric.

1. The second item of Definition 5 states verbatim that $\sigma \circ \tau$ is increasing.
2. It is easy to show that $\sigma \circ \tau$ is monotone since both σ and τ are antimonotone: if $K_1 \subseteq K_2$, then $\tau(K_2) \subseteq \tau(K_1)$. Consequently, $\sigma(\tau(K_1)) \subseteq \sigma(\tau(K_2))$ is implied, which means $(\sigma \circ \tau)(K_1) \subseteq (\sigma \circ \tau)(K_2)$.
3. Consider $K \subseteq \mathcal{I}$. We have to show that $(\sigma \circ \tau)((\sigma \circ \tau)(K)) \subseteq (\sigma \circ \tau)(K)$, that is, $\sigma(\tau(\sigma(\tau(K)))) \subseteq \sigma(\tau(K))$. Since $\tau \circ \sigma$ is increasing by definition, we have that $\tau(K) \subseteq \tau(\sigma(\tau(K)))$. Since σ is antimonotone, $\sigma(\tau(\sigma(\tau(K)))) \subseteq \sigma(\tau(K))$.

Towards showing that having the intersection property and being in a Galois correspondence are one and the same, we firstly derive a slightly modified characterization of the intersection property. Instead of decomposing theories into its singletons, this version considers families of theories.

Proposition 8. Let $(\mathcal{L}, \mathcal{I}, \sigma)$ be a logic. Its model function $\sigma : 2^{\mathcal{L}} \rightarrow 2^{\mathcal{I}}$ satisfies the intersection property if and only if for all $\mathcal{T} \subseteq 2^{\mathcal{L}}$:

$$\sigma\left(\bigcup_{T \in \mathcal{T}} T\right) = \bigcap_{T \in \mathcal{T}} \sigma(T)$$

Proof. if: Let $T \subseteq \mathcal{L}$ and define $\mathcal{T} = \{\{F\} \mid F \in T\}$. Clearly

$$T = \bigcup_{F \in T} \{F\} \tag{1}$$

and we derive:

$$\begin{aligned} \sigma(T) &= \sigma\left(\bigcup_{F \in T} \{F\}\right) && \text{(Equation (1))} \\ &= \sigma\left(\bigcup_{\{F\} \in \mathcal{T}} \{F\}\right) && \text{(definition of } \mathcal{T}\text{)} \\ &= \bigcap_{\{F\} \in \mathcal{T}} \sigma(\{F\}) && \text{(presumption)} \\ &= \bigcap_{F \in T} \sigma(\{F\}) && \text{(definition of } \mathcal{T}\text{)} \end{aligned}$$

only if: Let $\mathcal{T} \subseteq 2^{\mathcal{L}}$ and define $T = \bigcup_{U \in \mathcal{T}} U$. Clearly

$$\forall F \in \mathcal{L} : \left(F \in \bigcup_{U \in \mathcal{T}} U \iff \exists U \in \mathcal{T} : F \in U \right) \tag{2}$$

Now we have that

$$\begin{aligned} \sigma\left(\bigcup_{U \in \mathcal{T}} U\right) &= \sigma(T) && \text{(definition of } T\text{)} \\ &= \bigcap_{F \in T} \sigma(\{F\}) && \text{(presumption)} \\ &= \bigcap_{F \in \bigcup_{U \in \mathcal{T}} U} \sigma(\{F\}) && \text{(definition of } T\text{)} \\ &= \bigcap_{\substack{U \in \mathcal{T}, \\ F \in U}} \sigma(\{F\}) && \text{(Equation (2))} \\ &= \bigcap_{U \in \mathcal{T}} \left(\bigcap_{F \in U} \sigma(\{F\}) \right) && \text{(associativity of } \cap\text{)} \\ &= \bigcap_{U \in \mathcal{T}} \sigma(U) && \text{(presumption)} \quad \square \end{aligned}$$

In what follows, we will make implicit use of this result and consider the two formulations of the intersection property to be interchangeable.

We will also use an alternative, but equivalent formulation of Galois correspondences [32].

Proposition 9. *Let $(\mathcal{L}, \mathcal{I}, \sigma, \tau)$ be a logic. It holds that σ and τ are in Galois correspondence iff for all $T \subseteq \mathcal{L}$ and $K \subseteq \mathcal{I}$:*

$$K \subseteq \sigma(T) \iff T \subseteq \tau(K)$$

Proof. if: Assume that for all $T \subseteq \mathcal{L}$ and $K \subseteq \mathcal{I}$ we have, $K \subseteq \sigma(T)$ iff $T \subseteq \tau(K)$.

1. We start with showing that $\tau \circ \sigma$ is increasing.

Let $T \subseteq \mathcal{L}$. Obviously, $\sigma(T) \subseteq \sigma(T)$. Hence, if substituting $K = \sigma(T)$ we obtain by presumption $T \subseteq \tau(\sigma(T))$.

In the same spirit one may easily show that $\sigma \circ \tau$ is increasing.

2. We show now that σ is antimonotone.

Let $T_1 \subseteq T_2$. Then $T_1 \subseteq T_2 \subseteq \tau(\sigma(T_2))$ by the above. By presumption (with $K = \sigma(T_2)$), it follows that $\sigma(T_2) \subseteq \sigma(T_1)$.

Analogously one may show that τ is antimonotone.

only if: Let $T \subseteq \mathcal{L}$ and $K \subseteq \mathcal{I}$. If $K \subseteq \sigma(T)$, then $\tau(\sigma(T)) \subseteq \tau(K)$. Hence, we conclude that $T \subseteq \tau(\sigma(T)) \subseteq \tau(K)$. The reverse implication follows symmetrically. \square

We now conclude this section with its main result. It states that for any logic $(\mathcal{L}, \mathcal{I}, \sigma)$, the conditions “ σ has the intersection property” and “ σ is in a Galois correspondence with some τ ” are equivalent. The proof can be adapted from the literature [32, Propositions 7.31 and 7.33] to our slightly different setting with acceptable effort.

Theorem 10. *Let $(\mathcal{L}, \mathcal{I}, \sigma)$ be a logic.*

1. *If there is a $\tau : 2^{\mathcal{I}} \rightarrow 2^{\mathcal{L}}$ such that σ and τ are in Galois correspondence, then σ has the intersection property.*
2. *If σ has the intersection property, then we can define a theory function $\tau : 2^{\mathcal{I}} \rightarrow 2^{\mathcal{L}}$ with*

$$K \mapsto \bigcup_{\substack{T \subseteq \mathcal{L}, \\ K \subseteq \sigma(T)}} T$$

such that σ and τ are in Galois correspondence.

Proof. 1. Assume the presumption. We have to show that for any $\mathcal{T} \subseteq 2^{\mathcal{L}}$, we find that

$$\sigma\left(\bigcup_{T \in \mathcal{T}} T\right) = \bigcap_{T \in \mathcal{T}} \sigma(T)$$

Now denote $Z = \bigcup_{T \in \mathcal{T}} T$. We first show that $\sigma(Z)$ is a lower bound for the set $\sigma(\mathcal{T}) = \{\sigma(T) \mid T \in \mathcal{T}\}$. Clearly, for each $T \in \mathcal{T}$ we have $T \subseteq Z$. Applying antimonotonicity of σ we obtain $\sigma(Z) \subseteq \sigma(T)$ for each $T \in \mathcal{T}$. Now let $Q \subseteq \mathcal{I}$ be any lower bound for $\sigma(\mathcal{T})$. Then $Q \subseteq \sigma(T)$ for all $T \in \mathcal{T}$. By Proposition 9, we get $T \subseteq \tau(Q)$ for all $T \in \mathcal{T}$. By definition, this entails that $\bigcup_{T \in \mathcal{T}} T = Z \subseteq \tau(Q)$. Now using Proposition 9 again yields $Q \subseteq \sigma(Z)$ which implies that $\sigma(Z)$ is the greatest lower bound of $\sigma(\mathcal{T})$.

2. Let σ have the intersection property. By Proposition 4, it follows that σ is antimonotone. For antimonotonicity of τ , consider $K_1 \subseteq K_2$. Then

$$\tau(K_2) = \bigcup_{\substack{T \subseteq \mathcal{L}, \\ K_2 \subseteq \sigma(T)}} T \subseteq \bigcup_{\substack{T \subseteq \mathcal{L}, \\ K_1 \subseteq \sigma(T)}} T = \tau(K_1)$$

Now for showing that $\sigma \circ \tau$ and $\tau \circ \sigma$ are increasing.
Let $K \subseteq \mathcal{I}$. We find

$$K \subseteq \bigcap_{\substack{T \subseteq \mathcal{L}, \\ K \subseteq \sigma(T)}} \sigma(T) = \sigma \left(\bigcup_{\substack{T \subseteq \mathcal{L}, \\ K \subseteq \sigma(T)}} T \right) = \sigma(\tau(K))$$

Now let $T \subseteq \mathcal{L}$. By using $\sigma(T) \subseteq \sigma(T)$ it is easy to verify that

$$T \subseteq \bigcup_{\substack{S \subseteq \mathcal{L}, \\ \sigma(T) \subseteq \sigma(S)}} S = \tau(\sigma(T)) \quad \square$$

This is the main motivation for our definition saying that *classical* logics are exactly those that have the intersection property, or, equivalently, that have model and theory functions that are in Galois correspondence.⁷ Furthermore, as partly mentioned before, many well-studied logics (that we would call classical due to their ubiquity alone) *have* the intersection property simply by definition, as their fundamental building blocks are *formulas* instead of theories.

3. Characterization Logics

From now on we omit \mathcal{I} from the presentation of logics and thus write (\mathcal{L}, σ) , since concrete interpretations are immaterial for strong equivalence. Furthermore, we will distinguish between two important cases regarding the domain of σ . The first one is $\text{dom}(\sigma) = 2^{\mathcal{L}}$ (called *full* logics) and the second one is $\text{dom}(\sigma) = (2^{\mathcal{L}})_{\text{fin}} = \{T \in 2^{\mathcal{L}} \mid T \text{ is finite}\}$ (*finite-theory* logics), the restriction of \mathcal{L} to finite knowledge bases.

Definition 6. Let (\mathcal{L}, σ) be a logic. We define the binary relation *strong equivalence* $\equiv_s^\sigma \subseteq \text{dom}(\sigma) \times \text{dom}(\sigma)$ by

$$T_1 \equiv_s^\sigma T_2 \iff \forall U \in \text{dom}(\sigma) : \sigma(T_1 \cup U) = \sigma(T_2 \cup U). \quad \diamond$$

It is straightforward to show that \equiv_s^σ is an equivalence relation; we denote the equivalence class of a theory $T \in \text{dom}(\sigma) \subseteq 2^{\mathcal{L}}$ by $[T]_s^\sigma$. We recall and will pervasively use that for all theories $T_1, T_2 \subseteq \mathcal{L}$, we have $T_1 \in [T_2]_s^\sigma$ if and only if $[T_1]_s^\sigma = [T_2]_s^\sigma$.

Given an arbitrary logic (\mathcal{L}, σ) , we want to find a characterizing classical logic, that is, a semantics σ' that has the intersection property and whose ordinary equivalence coincides with strong σ -equivalence. Such logics get a name.

⁷Along with the aforementioned parallels to Goguen and Burstall's [10] treatment of abstract logical systems, likewise leading to Galois correspondences.

Definition 7. Let (\mathcal{L}, σ) be a (full) logic. The logic (\mathcal{L}, σ') is a (full) *characterization logic* for (\mathcal{L}, σ) if and only if:

$$\forall T_1, T_2 \subseteq \mathcal{L} : \sigma'(T_1) = \sigma'(T_2) \iff [T_1]_s^\sigma = [T_2]_s^\sigma \quad (\text{characterization})$$

$$\forall \mathcal{T} \subseteq 2^\mathcal{L} : \sigma' \left(\bigcup_{T \in \mathcal{T}} T \right) = \bigcap_{T \in \mathcal{T}} \sigma'(T) \quad (\text{intersection})$$

◇

Since characterization logics are the centerpiece of our study we would like to take a look at this central definition from another angle, namely in terms of consequence functions. The analysis of consequence relations has been prominent in the formative years of nonmonotonic reasoning; we refer the interested reader to works by Gabbay [33], Kraus, Lehmann and Magidor [34] as well as Makison [35].⁸ Let us consider the canonical consequence functions of σ and σ' according to Definition 1. Doing so reveals that the characterizing and characterized consequence relations fit together nicely although we do not require anything from σ . It is part of future work to study further properties in terms of consequence relations.

Proposition 11. Let (\mathcal{L}, σ) be a logic with characterization logic (\mathcal{L}, σ') with Cn^σ and $Cn^{\sigma'}$ the canonical consequence functions of the corresponding logics. For any $T \subseteq \mathcal{L}$, we have:

1. $Cn^{\sigma'}(T) \subseteq Cn^\sigma(T)$ (sublogic)
2. $Cn^\sigma(T) = Cn^{\sigma'}(Cn^{\sigma'}(T))$ (right absorption)
3. $Cn^\sigma(T) \subseteq Cn^{\sigma'}(Cn^\sigma(T))$, and
 $Cn^\sigma(T) = Cn^{\sigma'}(Cn^\sigma(T))$ if Cn^σ satisfies idempotence. (left absorption)

Proof. Remember that the induced canonical consequence $Cn^{\sigma'}$ is a closure operator as (\mathcal{L}, σ') is a logic possessing the intersection property (Definition 7, Proposition 6). In contrast, Cn^σ is not necessarily idempotent (Example 3) but always increasing (Proposition 1).

1. In order to show $Cn^{\sigma'}(T) \subseteq Cn^\sigma(T)$ we first prove the subsequent property (*). For any $U \subseteq \mathcal{L}$: If $U \subseteq Cn^{\sigma'}(T)$, then $Cn^{\sigma'}(T) = Cn^{\sigma'}(T \cup U)$. Since $T \subseteq T \cup U$ we derive $Cn^{\sigma'}(T) \subseteq Cn^{\sigma'}(T \cup U)$ (monotonicity). Moreover, we have $T \subseteq Cn^{\sigma'}(T)$ (increasing) and $U \subseteq Cn^{\sigma'}(T)$ (assumption) justifying $T \cup U \subseteq Cn^{\sigma'}(T)$. Hence, $Cn^{\sigma'}(T \cup U) \subseteq Cn^{\sigma'}(Cn^{\sigma'}(T))$ (monotonicity) and finally, $Cn^{\sigma'}(T \cup U) \subseteq Cn^{\sigma'}(T)$ (idempotence) concluding the proof for property (*).

Now, let $U \subseteq Cn^{\sigma'}(T)$. We have to show $U \subseteq Cn^\sigma(T)$. Due to Proposition 5 and property (*) we obtain $\sigma'(T) = \sigma'(Cn^{\sigma'}(T)) = \sigma'(Cn^{\sigma'}(T \cup U)) = \sigma'(T \cup U)$. Since (\mathcal{L}, σ') is assumed to be a characterization logic we derive $[T]_s^\sigma = [T \cup U]_s^\sigma$. Consequently, $\sigma(T) = \sigma(T \cup U)$ implying $Cn^\sigma(T) = Cn^\sigma(T \cup U)$ in consideration of Definition 1. Using the previous equality and that canonical consequence is increasing (Proposition 1) we finally obtain $U \subseteq T \cup U \subseteq Cn^\sigma(T \cup U) = Cn^\sigma(T)$.

⁸Similarly, Straßburger [11] “defines” a logic as a pair (\mathcal{L}, \models) of a language with a consequence pre-order $\models \subseteq \mathcal{L} \times \mathcal{L}$, that is, a relation on *single formulas*. It is clear that a consequence function C in our sense can be reconstructed whenever the language \mathcal{L} allows for (possibly generalized) conjunction, i.e. $C(T) = \{F \in \mathcal{L} \mid \bigwedge T \models F\}$. Straßburger’s main focus was not non-monotonicity, but the presence of conjunction in \mathcal{L} would also allow to define strong equivalence via mapping theories to formulas and theory union to formula conjunction.

2. Due to Proposition 5 we have $\sigma'(T) = \sigma'(Cn^{\sigma'}(T))$. Since (\mathcal{L}, σ') is a characterization logic we further conclude $[T]_s^\sigma = [Cn^{\sigma'}(T)]_s^\sigma$. Hence, $\sigma(T) = \sigma(Cn^{\sigma'}(T))$ which guarantees $Cn^\sigma(T) = Cn^\sigma(Cn^{\sigma'}(T))$ by Definition 1.
3. We have $Cn^\sigma(T) \subseteq Cn^{\sigma'}(Cn^\sigma(T))$ since $Cn^{\sigma'}$ is increasing. Assuming idempotence of Cn^σ yields the converse direction and thus, equality. First, $Cn^{\sigma'}(Cn^\sigma(T)) \subseteq Cn^\sigma(Cn^\sigma(T))$ due to item 1 (sublogic) and secondly, $Cn^{\sigma'}(Cn^\sigma(T)) \subseteq Cn^\sigma(T)$ due to idempotence. Hence, $Cn^\sigma(T) = Cn^{\sigma'}(Cn^\sigma(T))$ as claimed. \square

We now start our analysis of characterization logics in terms of model functions. We first show that characterization logics are unique up to isomorphism. More precisely, for any model function σ , the algebras corresponding to the model theories of any two characterizing model functions σ' and σ'' are isomorphic. To do that, we first show that the model theory of any characterization logic⁹ is a complete lattice, that is, a partially ordered set where each subset of the carrier set has both a greatest lower bound (glb) and a least upper bound (lub).

Theorem 12. *Let (\mathcal{L}, σ) be a full logic with characterization logic (\mathcal{L}, σ') . The pair $(\sigma'(2^\mathcal{L}), \subseteq)$ is a complete lattice where glb \bigwedge and lub \bigvee are given such that for all $\mathcal{K} \subseteq \sigma'(2^\mathcal{L})$,*

$$\bigwedge_{K \in \mathcal{K}} K = \bigcap_{K \in \mathcal{K}} K \text{ and } \bigvee_{K \in \mathcal{K}} K = \bigwedge_{L \in \mathcal{K}^u} L$$

where $\mathcal{K}^u = \{L \in \sigma'(2^\mathcal{L}) \mid \forall K \in \mathcal{K} : K \subseteq L\}$ is the set of upper bounds of \mathcal{K} .

Proof. Let $\mathcal{K} \subseteq \sigma'(2^\mathcal{L})$; we first show $\bigcap_{K \in \mathcal{K}} K \in \sigma'(2^\mathcal{L})$. Clearly for each $K \in \mathcal{K} \subseteq \sigma'(2^\mathcal{L})$ there exists a $T \subseteq \mathcal{L}$ with $\sigma'(T) = K$. Thus by the axiom of choice there is a $\mathcal{T} \subseteq 2^\mathcal{L}$ that contains a $T \in \mathcal{T}$ with $\sigma'(T) = K$ for each $K \in \mathcal{K}$. Since $\bigcup_{T \in \mathcal{T}} T \in 2^\mathcal{L}$ and σ' has the intersection property, $\bigcap_{K \in \mathcal{K}} K = \bigcap_{T \in \mathcal{T}} \sigma'(T) = \sigma'(\bigcup_{T \in \mathcal{T}} T) \in \sigma'(2^\mathcal{L})$.

Now consider $\bigvee_{K \in \mathcal{K}} K$. We show that $\bigwedge_{L \in \mathcal{K}^u} L$ is the least element of \mathcal{K}^u , the set of all upper bounds of \mathcal{K} . Clearly, $L \in \mathcal{K}^u$ implies that $\forall K \in \mathcal{K} : K \subseteq L$. Thus, $\forall K \in \mathcal{K} : K \subseteq \bigwedge_{L \in \mathcal{K}^u} L$ whence $\bigwedge_{L \in \mathcal{K}^u} L \in \mathcal{K}^u$. In particular, if $M \in \mathcal{K}^u$ then $\bigwedge_{L \in \mathcal{K}^u} L \subseteq M$ and $\bigwedge_{L \in \mathcal{K}^u} L$ is the least upper bound of \mathcal{K} . \square

It is vital that the least upper bound is defined in terms of the greatest lower bound, as ordinary set union will not work.

Example 7. Consider $\mathcal{L} = \{a, b, c\}$ and semantics σ with

$$\begin{aligned} [\emptyset]_s^\sigma &= \{\emptyset\}, \\ [\{a\}]_s^\sigma &= \{\{a\}\}, \\ [\{b\}]_s^\sigma &= \{\{b\}\}, \\ [\{c\}]_s^\sigma &= \{\{c\}\}, \\ [\{a, b\}]_s^\sigma &= [\{a, c\}]_s^\sigma = [\{b, c\}]_s^\sigma = [\{a, b, c\}]_s^\sigma = \{\{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\} \end{aligned}$$

⁹The proof of Theorem 12 reveals that (\mathcal{L}, σ') does not necessarily have to be a characterization logic of (\mathcal{L}, σ) . Indeed, the stated properties regarding the model theory hold for any logic possessing the intersection property.

Assume that (\mathcal{L}, σ') is a characterization logic for (\mathcal{L}, σ) . We claim

$$\sigma'(\{b\}) \cup \sigma'(\{c\}) \notin \sigma'(2^{\mathcal{L}})$$

implying that $\sigma'(2^{\mathcal{L}})$ is not closed under set union and thus cannot be the carrier set of any sublattice of $(2^{\mathcal{L}}, \subseteq)$. Assume to the contrary that there is a $T \subseteq \mathcal{L}$ with $\sigma'(T) = \sigma'(\{b\}) \cup \sigma'(\{c\})$. Then

$$\begin{aligned} \sigma'(\{a\} \cup T) &= \sigma'(\{a\}) \cap \sigma'(T) \\ &= \sigma'(\{a\}) \cap (\sigma'(\{b\}) \cup \sigma'(\{c\})) \\ &= (\sigma'(\{a\}) \cap \sigma'(\{b\})) \cup (\sigma'(\{a\}) \cap \sigma'(\{c\})) \\ &= \sigma'(\{a, b\}) \cup \sigma'(\{a, c\}) \\ &= \sigma'(\{a, b\}) = \sigma'(\{a, c\}) = \sigma'(\{b, c\}) = \sigma'(\{a, b, c\}) \end{aligned}$$

Thus $T = \{b\}$ or $T = \{c\}$ or $T \in [\{a, b, c\}]_s^\sigma$. Since σ' has the intersection property, it is in particular antimonotone (Proposition 4). Hence for $T \in [\{a, b, c\}]_s^\sigma$, we get $\sigma'(T) \subseteq \sigma'(\{b\})$ and since $[\{b\}]_s^\sigma \neq [\{a, b, c\}]_s^\sigma$ even $\sigma'(T) \subsetneq \sigma'(\{b\})$. Substituting c for b in the argument above, we get $\sigma'(T) \subsetneq \sigma'(\{c\})$ in the same fashion. In combination $\sigma'(T) \subsetneq \sigma'(\{b\}) \cup \sigma'(\{c\})$, contradiction. Thus $T = \{b\}$ or $T = \{c\}$. Clearly $[\{b\}]_s^\sigma \neq [\{b, c\}]_s^\sigma$ and $[\{c\}]_s^\sigma \neq [\{b, c\}]_s^\sigma$ imply that

$$\begin{aligned} \sigma'(\{b\}) \cap \sigma'(\{c\}) &= \sigma'(\{b, c\}) \subsetneq \sigma'(\{b\}) \\ \sigma'(\{b\}) \cap \sigma'(\{c\}) &= \sigma'(\{b, c\}) \subsetneq \sigma'(\{c\}) \end{aligned}$$

Thus $\sigma'(\{b\}) \setminus \sigma'(\{c\}) \neq \emptyset$ and $\sigma'(\{c\}) \setminus \sigma'(\{b\}) \neq \emptyset$. Therefore

$$\sigma'(\{b\}) \subsetneq \sigma'(\{b\}) \cup \sigma'(\{c\}) = \sigma'(T)$$

as well as

$$\sigma'(\{c\}) \subsetneq \sigma'(\{b\}) \cup \sigma'(\{c\}) = \sigma'(T).$$

Contradiction. Thus T does not exist and $\sigma'(\{b\}) \cup \sigma'(\{c\}) \notin \sigma'(2^{\mathcal{L}})$. \diamond

This example shows in particular that the resulting complete lattice induced by the characterization logic is not necessarily distributive, not even in the case of finite logics. After these necessary preliminaries, we now present the result on uniqueness of characterization logics.

Theorem 13. *Let (\mathcal{L}, σ) be a full logic with characterization logics (\mathcal{L}, σ') and (\mathcal{L}, σ'') . Then the complete lattices $(\sigma'(2^{\mathcal{L}}), \subseteq)$ and $(\sigma''(2^{\mathcal{L}}), \subseteq)$ are isomorphic.*

Proof. We provide a bijection $\phi : \sigma'(2^{\mathcal{L}}) \rightarrow \sigma''(2^{\mathcal{L}})$ with

$$\phi\left(\bigwedge_{K \in \mathcal{K}} K\right) = \bigwedge_{K \in \mathcal{K}} \phi(K) \text{ and } \phi\left(\bigvee_{K \in \mathcal{K}} K\right) = \bigvee_{K \in \mathcal{K}} \phi(K)$$

Given any $K \in \sigma'(2^{\mathcal{L}})$, there clearly exists a $T \subseteq \mathcal{L}$ with $\sigma'(T) = K$; now define $\phi(K)$ such that $\phi(K) = \sigma''(T)$. This means, $\phi(\sigma'(T)) = \phi(\sigma''(T))$.

- ϕ is injective: Let $K_1, K_2 \in \sigma'(2^{\mathcal{L}})$ with $\phi(K_1) = \phi(K_2)$. Clearly there exist $T_1, T_2 \subseteq \mathcal{L}$ such that $\sigma'(T_1) = K_1$ and $\sigma'(T_2) = K_2$. Thus $\phi(K_1) = \phi(K_2)$ implies that $\sigma''(T_1) = \phi(K_1) = \phi(K_2) = \sigma''(T_2)$. Since σ'' has the characterization property, we get $[T_1]_s^\sigma = [T_2]_s^\sigma$. Since σ' has the characterization property, we get $K_1 = \sigma'(T_1) = \sigma'(T_2) = K_2$.

- ϕ is surjective: Let $M \in \sigma''(2^{\mathcal{L}})$. Then there exists a $T \subseteq \mathcal{L}$ with $\sigma''(T) = M$. It follows by definition that $\phi(\sigma'(T)) = \sigma''(T) = M$.
- ϕ is structure-preserving: Let $\mathcal{K} \subseteq \sigma'(2^{\mathcal{L}})$ and define $\mathcal{T} \subseteq 2^{\mathcal{L}}$ such that for each $K \in \mathcal{K}$, the family \mathcal{T} contains an element of the preimage of K with respect to σ' . (\mathcal{T} need not be unique, but exists by the axiom of choice.)

$$\begin{aligned} \phi\left(\bigwedge_{K \in \mathcal{K}} K\right) &= \phi\left(\bigcap_{K \in \mathcal{K}} K\right) = \phi\left(\bigcap_{T \in \mathcal{T}} \sigma'(T)\right) = \phi\left(\sigma'\left(\bigcup_{T \in \mathcal{T}} T\right)\right) \\ &= \sigma''\left(\bigcup_{T \in \mathcal{T}} T\right) = \bigcap_{T \in \mathcal{T}} \sigma''(T) = \bigcap_{K \in \mathcal{K}} \phi(K) = \bigwedge_{K \in \mathcal{K}} \phi(K) \end{aligned}$$

Since \vee is defined in terms of \wedge , it follows that

$$\phi\left(\bigvee_{K \in \mathcal{K}} K\right) = \phi\left(\bigwedge_{K \in \mathcal{K}^u} K\right) = \bigwedge_{K \in \mathcal{K}^u} \phi(K) = \bigvee_{K \in \mathcal{K}} \phi(K)$$

Thus $(\sigma'(2^{\mathcal{L}}), \subseteq)$ and $(\sigma''(2^{\mathcal{L}}), \subseteq)$ are isomorphic. \square

Thus if a classical characterization logic exists, it is (up to isomorphism on its model theory) uniquely determined. However, as we show next, in some cases there simply is no characterization logic.

Example 8. Let $\mathcal{L} = \mathbb{N}$ be the natural numbers and $\mathcal{I} \neq \emptyset$ arbitrary. We define the semantics $\sigma : 2^{\mathcal{L}} \rightarrow 2^{\mathcal{I}}$ such that

$$\sigma(T) = \begin{cases} \emptyset & \text{if } T \text{ is finite,} \\ \mathcal{I} & \text{otherwise.} \end{cases}$$

There are two strong equivalence classes: $[\emptyset]_s^\sigma$, the set of all finite subsets of \mathbb{N} , and $[\mathbb{N}]_s^\sigma$, the set of all infinite subsets of \mathbb{N} . Assume that (\mathcal{L}, σ') is a characterization logic for (\mathcal{L}, σ) . By the model intersection property, we get

$$\sigma'(\mathbb{N}) = \sigma'\left(\bigcup_{n \in \mathbb{N}} \{n\}\right) = \bigcap_{n \in \mathbb{N}} \sigma'(\{n\}) = \sigma'(\emptyset)$$

in contradiction to the characterization property. \diamond

The example above motivates the study of special features of logics warranting the existence of characterization logics. We now proceed with some useful properties that will be needed in this endeavour later on. Most importantly, we show that strong equivalence classes have an expansion property: It is not completely obvious, but follows easily from the definition of strong equivalence that two strongly equivalent theories can both be expanded (via set union) with the same theory and are again strongly equivalent; the converse holds as well. Furthermore, the union of two strongly equivalent theories is again strongly equivalent to the two theories. Finally, for any two strongly equivalent theories that are in subset relation, every theory in between them is also strongly equivalent to them.

Lemma 14. Let (\mathcal{L}, σ) be a full logic and $T, T_1, T_2, T_3 \subseteq \mathcal{L}$.

1. Strong equivalence is invariant to expansion:

$$[T_1]_s^\sigma = [T_2]_s^\sigma \iff \left(\forall U \subseteq \mathcal{L} : [T_1 \cup U]_s^\sigma = [T_2 \cup U]_s^\sigma \right)$$

2. Each strong equivalence class is a join-semilattice:

$$T_1, T_2 \in [T]_s^\sigma \implies T_1 \cup T_2 \in [T]_s^\sigma$$

3. Each strong equivalence class is convex:

$$T_1 \in [T]_s^\sigma \wedge T_3 \in [T]_s^\sigma \wedge T_1 \subseteq T_2 \subseteq T_3 \implies T_2 \in [T]_s^\sigma$$

Proof. 1.

$$\begin{aligned} & [T_1]_s^\sigma = [T_2]_s^\sigma \\ \iff & \forall U \subseteq \mathcal{L} : \sigma(T_1 \cup U) = \sigma(T_2 \cup U) && (\text{Def. } \equiv_s^\sigma) \\ \iff & \forall U' \subseteq \mathcal{L} : \forall U'' \subseteq \mathcal{L} : \sigma(T_1 \cup (U' \cup U'')) = \sigma(T_2 \cup (U' \cup U'')) && (\text{write } U = U' \cup U'') \\ \iff & \forall U' \subseteq \mathcal{L} : \forall U'' \subseteq \mathcal{L} : \sigma((T_1 \cup U') \cup U'') = \sigma((T_2 \cup U') \cup U'') && (\text{associativity } \cup) \\ \iff & \forall U' \subseteq \mathcal{L} : [T_1 \cup U']_s^\sigma = [T_2 \cup U']_s^\sigma && (\text{Def. } \equiv_s^\sigma) \\ \iff & \forall U \subseteq \mathcal{L} : [T_1 \cup U]_s^\sigma = [T_2 \cup U]_s^\sigma && (\text{rename } U' = U) \end{aligned}$$

2.

$$\begin{aligned} & T_1, T_2 \in [T]_s^\sigma \\ \iff & [T_1]_s^\sigma = [T_2]_s^\sigma = [T]_s^\sigma \\ \implies & [T_1 \cup T_2]_s^\sigma = [T_2 \cup T_2]_s^\sigma = [T_2]_s^\sigma = [T]_s^\sigma \\ \implies & T_1 \cup T_2 \in [T]_s^\sigma \end{aligned}$$

3. Let $T_1, T_3 \in [T]_s^\sigma$ and $T_1 \subseteq T_2 \subseteq T_3$.

$$\begin{aligned} & [T_1]_s^\sigma = [T_3]_s^\sigma \\ \implies & [T_1 \cup T_2]_s^\sigma = [T_3 \cup T_2]_s^\sigma && (\text{Item 1}) \\ \implies & [T_2]_s^\sigma = [T_3]_s^\sigma && (T_1 \subseteq T_2 \subseteq T_3) \\ \implies & T_2 \in [T_3]_s^\sigma = [T]_s^\sigma && \square \end{aligned}$$

We also take a closer look on the structure of strong equivalence classes and their relationships with each other. We first define a binary relation \sqsubseteq on strong equivalence classes where two classes are in \sqsubseteq -relation if and only if there are \sqsubseteq -comparable representatives of the classes.

Definition 8. Let (\mathcal{L}, σ) be a logic. Define the following relation on strong equivalence classes:

$$[S]_s^\sigma \sqsubseteq [T]_s^\sigma \iff \exists S' \in [S]_s^\sigma : \exists T' \in [T]_s^\sigma : S' \subseteq T' \quad \diamond$$

It is easy to show (and later useful) that \sqsubseteq is a partial order.

Lemma 15. The relation \sqsubseteq is a partial order, that is, reflexive, antisymmetric and transitive.

Proof. Reflexivity is clear.

Antisymmetric: Let $[S]_s^\sigma \sqsubseteq [T]_s^\sigma$ and $[T]_s^\sigma \sqsubseteq [S]_s^\sigma$. Then there are $S_1, S_2 \in [S]_s^\sigma$ as well as $T_1, T_2 \in [T]_s^\sigma$ with $S_1 \subseteq T_1$ and $T_2 \subseteq S_2$. Clearly

$$\begin{aligned} & [T_1]_s^\sigma = [T_2]_s^\sigma \\ \implies & [T_1 \cup S_2]_s^\sigma = [T_2 \cup S_2]_s^\sigma && \text{(Item 1)} \\ \implies & [T_1 \cup S_2]_s^\sigma = [S_2]_s^\sigma && (T_2 \subseteq S_2) \\ \implies & [T_1 \cup S_2]_s^\sigma = [S]_s^\sigma && (S_2 \in [S]_s^\sigma) \end{aligned}$$

Likewise

$$\begin{aligned} & [S_1]_s^\sigma = [S_2]_s^\sigma \\ \implies & [S_1 \cup T_2]_s^\sigma = [S_2 \cup T_2]_s^\sigma && \text{(Item 1)} \\ \implies & [S_1 \cup T_2]_s^\sigma = [S_2]_s^\sigma && (T_2 \subseteq S_2) \\ \implies & [S_1 \cup T_2 \cup T_1]_s^\sigma = [S_2 \cup T_1]_s^\sigma && \text{(Item 1)} \\ \implies & [T_2 \cup T_1]_s^\sigma = [T_1 \cup S_2]_s^\sigma && (S_1 \subseteq T_1) \\ \implies & [T]_s^\sigma = [T_1 \cup S_2]_s^\sigma && (T_1, T_2 \in [T]_s^\sigma, \text{ Lemma 14, Item 2}) \end{aligned}$$

In combination, $[S]_s^\sigma = [T_1 \cup S_2]_s^\sigma = [T]_s^\sigma$ as desired.

Transitive: Let $[T_1]_s^\sigma \sqsubseteq [T_2]_s^\sigma$ and $[T_2]_s^\sigma \sqsubseteq [T_3]_s^\sigma$. Then there exist $T'_1 \in [T_1]_s^\sigma$, $T'_2, T''_2 \in [T_2]_s^\sigma$ and $T''_3 \in [T_3]_s^\sigma$ with $T'_1 \subseteq T'_2$ and $T''_2 \subseteq T''_3$. Since $[T'_2]_s^\sigma = [T''_2]_s^\sigma$, we can conclude that $[T'_2 \cup T''_3]_s^\sigma = [T''_2 \cup T''_3]_s^\sigma = [T''_3]_s^\sigma$. Now it is clear that $T'_1 \subseteq T'_2 \subseteq T'_2 \cup T''_3$ with $T'_1 \in [T_1]_s^\sigma$ and $T'_2 \cup T''_3 \in [T_3]_s^\sigma$ imply that $[T_1]_s^\sigma \sqsubseteq [T_3]_s^\sigma$. \square

It follows from Lemma 14 in particular that in the case of logics (\mathcal{L}, σ) with \mathcal{L} finite, each strong equivalence class $[T]_s^\sigma$ has a \subseteq -greatest element that equals the union of all elements. However, for logics with infinite \mathcal{L} , this need not be the case: in the logic of Example 8, the class $[\emptyset]_s^\sigma$ has no maximal elements, in particular no greatest element; the class $[\mathbb{N}]_s^\sigma$ has no minimal elements, in particular no least element. We will see that having a \subseteq -greatest element in each equivalence class is sufficient for the existence of a characterization logic. We therefore decided to name this class of logics and study it in some greater detail.

3.1. Covered Logics

Definition 9. Let (\mathcal{L}, σ) be a logic with strong equivalence relation \equiv_s^σ . For an equivalence class $[T]_s^\sigma \in (2^\mathcal{L})_{/\equiv_s^\sigma} = \{[T]_s^\sigma \mid T \subseteq \mathcal{L}\}$, we define its *cover* to be the set

$$\widehat{[T]_s^\sigma} = \bigcup_{S \in [T]_s^\sigma} S$$

and say that a logic (\mathcal{L}, σ) is *covered* if and only if $\forall T \subseteq \mathcal{L} : \widehat{[T]_s^\sigma} \in [T]_s^\sigma$. \diamond

Roughly, the existence of greatest elements in equivalence classes guarantees that these classes are closed under arbitrary set union.¹⁰ Clearly any finite logic is covered. Furthermore, two familiar representatives of covered logics are classical logic and abstract argumentation theory. In the former case, it is clear that arbitrary unions of families of equivalent theories are again

¹⁰Strong equivalence classes are always closed under set unions with non-empty, finite index sets (Lemma 14).

theories that are equivalent to each of its members. In the latter case it is not immediately clear but can be shown with reasonable effort.

Towards the main result of this section, we show that covered logics behave “nicely” in an algebraic sense, which will pave the way for obtaining characterization logics for them. Most importantly, while we know from Theorem 12 that the strong equivalence classes of any logic form a complete lattice, for covered logics we can even specify the join and meet operations directly via set operations on covers and subsequent class formation.

Lemma 16. *Let (\mathcal{L}, σ) be a covered logic with strong equivalence relation \equiv_s^σ . The pair $((2^\mathcal{L})_{/\equiv_s^\sigma}, \sqsubseteq)$ is a complete lattice with operations*

$$\bigsqcup_{C \in \mathcal{C}} C = \left[\bigcup_{C \in \mathcal{C}} \widehat{C} \right]_s^\sigma \quad \text{and} \quad \prod_{C \in \mathcal{C}} C = \left[\bigcap_{C \in \mathcal{C}} \widehat{C} \right]_s^\sigma$$

Proof. Let $\mathcal{C} \subseteq \{[T]_s^\sigma \mid T \subseteq \mathcal{L}\}$.

- $D = \bigsqcup_{C \in \mathcal{C}} C$ is the least upper bound of \mathcal{C} :

For any $C \in \mathcal{C}$, we get $\widehat{C} \subseteq \bigcup_{B \in \mathcal{C}} \widehat{B}$ immediately and $\widehat{C} \in C$ since (\mathcal{L}, σ) is covered. The fact that $\bigcup_{B \in \mathcal{C}} \widehat{B} \in \left[\bigcup_{B \in \mathcal{C}} \widehat{B} \right]_s^\sigma = D$ is also immediate. Consequently, $C \sqsubseteq D$ and by arbitrary choice of C we get that D is an upper bound of \mathcal{C} .

Now let E be any upper bound of \mathcal{C} and consider an arbitrary $C \in \mathcal{C}$. Since $C \sqsubseteq E$, there are $T_C \in C$ and $T'_C \in E$ such that $T_C \subseteq T'_C$. Since (\mathcal{L}, σ) is covered, $[T_C]_s^\sigma = C = [\widehat{C}]_s^\sigma$. Hence, by strong equivalence $[T_C \cup T'_C]_s^\sigma = [\widehat{C} \cup T'_C]_s^\sigma$. That is, $\widehat{C} \cup T'_C \in [T_C \cup T'_C]_s^\sigma = [T'_C]_s^\sigma = E$. Since C was arbitrary, we have that $\forall C \in \mathcal{C} : \exists T'_C \in E : \widehat{C} \cup T'_C \in E$. In particular, $\forall C \in \mathcal{C} : \exists T'_C \in E : \widehat{C} \cup T'_C \subseteq \widehat{E}$ whence $\bigcup_{C \in \mathcal{C}} (\widehat{C} \cup T'_C) \subseteq \widehat{E}$. Now fix a specific $B \in \mathcal{C}$; then we have in particular that there exists a $T'_B \in E$ such that $\widehat{B} \cup T'_B \in E$. Furthermore, $\widehat{B} \cup T'_B \subseteq \bigcup_{C \in \mathcal{C}} (\widehat{C} \cup T'_C) \subseteq \widehat{E}$ with $\widehat{E} \in E$ since (\mathcal{L}, σ) is covered. By Lemma 14 (Item 3) saying that strong equivalence classes are convex, we get $\bigcup_{C \in \mathcal{C}} (\widehat{C} \cup T'_C) \in E$. Together with $\bigcup_{C \in \mathcal{C}} \widehat{C} \subseteq \bigcup_{C \in \mathcal{C}} (\widehat{C} \cup T'_C)$ and $\bigcup_{C \in \mathcal{C}} \widehat{C} \in D$ we get $D \sqsubseteq E$ and D is the least upper bound of \mathcal{C} .

- $D = \prod_{C \in \mathcal{C}} C$ is the greatest lower bound of \mathcal{C} :

As above, $C \in \mathcal{C}$ implies $\bigcap_{B \in \mathcal{C}} \widehat{B} \subseteq \widehat{C}$. Consequently, by $\widehat{C} \in C$ and $\bigcap_{B \in \mathcal{C}} \widehat{B} \in D$ we get $D \sqsubseteq C$. Since $C \in \mathcal{C}$ was arbitrarily chosen, D is a lower bound of \mathcal{C} .

Now let E be any lower bound of \mathcal{C} . Thus by definition, for each $C \in \mathcal{C}$ there exist $T_C \in C$ and $T'_C \in E$ such that $T'_C \subseteq T_C$. Now fix one $C \in \mathcal{C}$ and consider an arbitrary $B \in \mathcal{C}$. Clearly $[T'_C]_s^\sigma = [T'_B]_s^\sigma = E$ justifying $[T'_C \cup T_B]_s^\sigma = [T'_B \cup T_B]_s^\sigma = [T_B]_s^\sigma$ by strong equivalence and since $T'_B \subseteq T_B$. Therefore, $T'_C \cup T_B \in B$, that is, $T'_C \subseteq T'_C \cup T_B \subseteq \widehat{B}$ with $\widehat{B} \in B$ since (\mathcal{L}, σ) is covered. Since B was chosen arbitrarily, we get the following: for every $B \in \mathcal{C}$, we have that $T'_C \subseteq \widehat{B}$. Consequently $T'_C \subseteq \bigcap_{B \in \mathcal{C}} \widehat{B}$ with $T'_C \in E$ and $\bigcap_{B \in \mathcal{C}} \widehat{B} \in D$. Thus $E \sqsubseteq D$ and D is the greatest lower bound of \mathcal{C} . \square

In particular, the least and greatest elements of that lattice are given by

$$\bigsqcup_{C \in \emptyset} C = \left[\bigcup_{C \in \emptyset} \widehat{C} \right]_s^\sigma = [\emptyset]_s^\sigma \quad \text{and} \quad \prod_{C \in \emptyset} C = \left[\bigcap_{C \in \emptyset} \widehat{C} \right]_s^\sigma = [\mathcal{L}]_s^\sigma.$$

As we show next, it follows that the mapping $[\cdot]_s^\sigma : 2^\mathcal{L} \rightarrow (2^\mathcal{L})_{/\equiv_s^\sigma}$ assigning a theory T its equivalence class $[T]_s^\sigma$ is join-preserving for arbitrary joins. This is akin to the intersection property for the characterizing semantics, only that the semantics is not yet a model theory in the form of a complete lattice of sets but, more generally, a complete lattice (that is not necessarily a lattice of sets).

Lemma 17. *Let (\mathcal{L}, σ) be a covered logic with strong equivalence relation \equiv_s^σ . For all $\mathcal{T} \subseteq 2^\mathcal{L}$, we have:*

$$\left[\bigcup_{T \in \mathcal{T}} T \right]_s^\sigma = \bigsqcup_{T \in \mathcal{T}} [T]_s^\sigma$$

Proof. Let $\mathcal{T} \subseteq 2^\mathcal{L}$ and denote $\mathcal{C} = \{[T]_s^\sigma \mid T \in \mathcal{T}\}$. Clearly $\bigsqcup_{T \in \mathcal{T}} [T]_s^\sigma = \bigsqcup \mathcal{C}$. Consequently, it suffices to show that

$$\left[\bigcup_{T \in \mathcal{T}} T \right]_s^\sigma = \bigsqcup \mathcal{C}.$$

This will be done by showing that $D = \left[\bigcup_{T \in \mathcal{T}} T \right]_s^\sigma$ is the least upper bound of \mathcal{C} in $((2^\mathcal{L})_{/\equiv_s^\sigma}, \sqsubseteq)$.

Let $C \in \mathcal{C}$. Then there is a $T \in \mathcal{T}$ with $C = [T]_s^\sigma$. Clearly $T \in C$ and

$$T \subseteq \bigcup_{S \in \mathcal{T}} S \subseteq \left[\bigcup_{S \in \mathcal{T}} S \right]_s^\sigma = \widehat{D}$$

with $\widehat{D} \in D$ since (\mathcal{L}, σ) is covered. Thus $C \sqsubseteq D$. Since $C \in \mathcal{C}$ was arbitrarily chosen, D is an upper bound of \mathcal{C} .

Now let E be any upper bound of \mathcal{C} . Consider an arbitrary $C \in \mathcal{C}$. There clearly is a $T \in \mathcal{T}$ with $C = [T]_s^\sigma$ and since E is an upper bound for \mathcal{C} there also are $T_C \in C$ and $T'_C \in E$ with $T_C \subseteq T'_C$.

We have $[T \cup T'_C]_s^\sigma = [T_C \cup T'_C]_s^\sigma = [T'_C]_s^\sigma = [E]_s^\sigma$ which implies $T \cup T'_C \in E$. Now since every $C \in \mathcal{C}$ originates in some $T \in \mathcal{T}$ we get that for every $T \in \mathcal{T}$ there exists a $T'_C \in E$ with $T \cup T'_C \in E$, that

$$\text{is,} \quad T \cup T'_C \subseteq \widehat{E}. \quad \text{Then clearly}$$

$\bigcup_{T \in \mathcal{T}} T \subseteq \bigcup_{T \in \mathcal{T}} (T \cup T'_C) \subseteq \widehat{E}$, and by $\bigcup_{T \in \mathcal{T}} T \in D$ and $\widehat{E} \in E$ (the logic (\mathcal{L}, σ) is covered) we get $D \sqsubseteq E$. Since E was an arbitrarily chosen upper bound of \mathcal{C} , D is the least upper bound of \mathcal{C} . \square

While it preserves arbitrary joins, the mapping from theories to their strong equivalence classes does not necessarily preserve meets. It is easy to see that for the meet operation in the complete lattice of strong equivalence classes we have $[\bigcap_{T \in \mathcal{T}} T]_s^\sigma \sqsubseteq \prod_{T \in \mathcal{T}} [T]_s^\sigma$, as

$$\bigcap_{T \in \mathcal{T}} T \subseteq \bigcap_{T \in \mathcal{T}} \widehat{[T]_s^\sigma} \in \left[\bigcap_{T \in \mathcal{T}} \widehat{[T]_s^\sigma} \right]_s^\sigma = \prod_{T \in \mathcal{T}} [T]_s^\sigma$$

The reverse relation does not hold, as is witnessed by the following small, finite logic.

Example 9. Let $\mathcal{L} = \{a, b\}$ and \equiv_s^σ such that $\emptyset \not\equiv_s^\sigma \{a\} \equiv_s^\sigma \{b\} \equiv_s^\sigma \{a, b\}$. Then we get:

$$\begin{aligned} [\{a\} \cap \{b\}]_s^\sigma &= [\emptyset]_s^\sigma = \{\emptyset\} \\ [\{a\}]_s^\sigma \cap [\{b\}]_s^\sigma &= [\{a, b\}]_s^\sigma \cap [\{a, b\}]_s^\sigma = [\{a, b\}]_s^\sigma = \{\{a\}, \{b\}, \{a, b\}\} \neq \{\emptyset\} \end{aligned} \quad \diamond$$

However, this is not a hindrance since the intersection property only needs to hold for arbitrary unions of theories and does not say anything about theory intersection.

We conclude this section with its main theorem showing that any full logic being covered possesses a characterization logic. The relevant characterization logic can even be defined (more or less) explicitly via a Herbrand-style canonical construction: roughly, the semantics of the characterization logic maps a theory T of the original language to the union of all strong equivalence classes $[S]_s^\sigma$ that are in \sqsubseteq -relation to the class $[T]_s^\sigma$ of the input theory.

Theorem 18. *Let (\mathcal{L}, σ) be a logic. If (\mathcal{L}, σ) is covered then a characterization logic for (\mathcal{L}, σ) is given by (\mathcal{L}, σ') with*

$$\sigma' : 2^\mathcal{L} \rightarrow 2^{2^\mathcal{L}}, \quad T \mapsto \bigcup_{\substack{S \in 2^\mathcal{L}, \\ [T]_s^\sigma \sqsubseteq [S]_s^\sigma}} [S]_s^\sigma$$

Proof. characterization: We have to show $\forall T_1, T_2 \subseteq \mathcal{L} : \sigma'(T_1) = \sigma'(T_2) \iff [T_1]_s^\sigma = [T_2]_s^\sigma$.

Let $T_1, T_2 \subseteq \mathcal{L}$.

\implies : Let $\sigma'(T_1) = \sigma'(T_2)$. Then by definition of σ' we get

$$\bigcup_{\substack{S \in 2^\mathcal{L}, \\ [T_1]_s^\sigma \sqsubseteq [S]_s^\sigma}} [S]_s^\sigma = \bigcup_{\substack{S \in 2^\mathcal{L}, \\ [T_2]_s^\sigma \sqsubseteq [S]_s^\sigma}} [S]_s^\sigma$$

that is, for any $S \in 2^\mathcal{L}$ we find that $[T_1]_s^\sigma \sqsubseteq [S]_s^\sigma$ iff $[T_2]_s^\sigma \sqsubseteq [S]_s^\sigma$. (Clearly, the set $\{[S]_s^\sigma \mid S \in 2^\mathcal{L}\}$ corresponds to a partition of $2^\mathcal{L}$; thus for any element $T \in \bigcup_{S \in 2^\mathcal{L}, [T_1]_s^\sigma \sqsubseteq [S]_s^\sigma} [S]_s^\sigma$ we have $[T_1]_s^\sigma \sqsubseteq [T]_s^\sigma$, and if also $T \in \bigcup_{S \in 2^\mathcal{L}, [T_2]_s^\sigma \sqsubseteq [S]_s^\sigma} [S]_s^\sigma$, then $[T_2]_s^\sigma \sqsubseteq [T]_s^\sigma$ as well. Symmetry applies to obtain the above conclusion.)

In particular, $T_1, T_2 \in 2^\mathcal{L}$ whence $[T_1]_s^\sigma \sqsubseteq [T_1]_s^\sigma$ iff $[T_2]_s^\sigma \sqsubseteq [T_1]_s^\sigma$, and $[T_1]_s^\sigma \sqsubseteq [T_2]_s^\sigma$ iff $[T_2]_s^\sigma \sqsubseteq [T_2]_s^\sigma$. This shows $[T_1]_s^\sigma \sqsubseteq [T_2]_s^\sigma$ and $[T_2]_s^\sigma \sqsubseteq [T_1]_s^\sigma$, that is, $[T_1]_s^\sigma = [T_2]_s^\sigma$.

\impliedby : Let $[T_1]_s^\sigma = [T_2]_s^\sigma$. Then immediately

$$\sigma'(T_1) = \bigcup_{\substack{S \in 2^\mathcal{L}, \\ [T_1]_s^\sigma \sqsubseteq [S]_s^\sigma}} [S]_s^\sigma = \bigcup_{\substack{S \in 2^\mathcal{L}, \\ [T_2]_s^\sigma \sqsubseteq [S]_s^\sigma}} [S]_s^\sigma = \sigma'(T_2)$$

intersection: We have to show $\forall \mathcal{T} \subseteq 2^\mathcal{L} : \sigma'(\bigcup_{T \in \mathcal{T}} T) = \bigcap_{T \in \mathcal{T}} \sigma'(T)$.

Let $\mathcal{T} \subseteq 2^\mathcal{L}$. From Lemma 17, we know that $[\bigcup_{T \in \mathcal{T}} T]_s^\sigma = \bigsqcup_{T \in \mathcal{T}} [T]_s^\sigma$ is the least upper bound of $\{[T]_s^\sigma \mid T \in \mathcal{T}\}$, whence for all $S \in 2^\mathcal{L}$, we have that $\bigsqcup_{T \in \mathcal{T}} [T]_s^\sigma \sqsubseteq [S]_s^\sigma$ if and only

if $[S]_s^\sigma$ is an upper bound of $\{[T]_s^\sigma \mid T \in \mathcal{T}\}$, that is,

$$\left[\bigcup_{T \in \mathcal{T}} T \right]_s^\sigma \sqsubseteq [S]_s^\sigma \iff \forall T \in \mathcal{T} : [T]_s^\sigma \sqsubseteq [S]_s^\sigma$$

Using this relationship, we get that

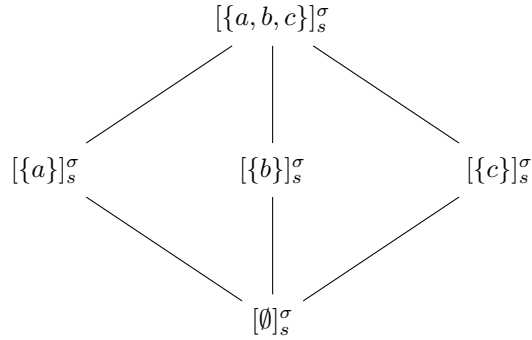
$$\begin{aligned} \sigma' \left(\bigcup_{T \in \mathcal{T}} T \right) &= \bigcup_{\substack{S \in 2^{\mathcal{L}}, \\ [\bigcup_{T \in \mathcal{T}} T]_s^\sigma \sqsubseteq [S]_s^\sigma}} [S]_s^\sigma \\ &= \bigcup_{\substack{S \in 2^{\mathcal{L}}, \\ \forall T \in \mathcal{T} : [T]_s^\sigma \sqsubseteq [S]_s^\sigma}} [S]_s^\sigma \\ &= \bigcap_{T \in \mathcal{T}} \left(\bigcup_{\substack{S \in 2^{\mathcal{L}}, \\ [T]_s^\sigma \sqsubseteq [S]_s^\sigma}} [S]_s^\sigma \right) \\ &= \bigcap_{T \in \mathcal{T}} \sigma'(T) \end{aligned} \quad \square$$

The construction from the statement of the theorem looks quite abstract, so we illustrate with a small concrete logic.

Example 10. We reconsider the logic from Example 7. There, $\mathcal{L} = \{a, b, c\}$ and

$$\begin{aligned} [\emptyset]_s^\sigma &= \{\emptyset\}, \\ [\{a\}]_s^\sigma &= \{\{a\}\}, \\ [\{b\}]_s^\sigma &= \{\{b\}\}, \\ [\{c\}]_s^\sigma &= \{\{c\}\}, \\ [\{a, b\}]_s^\sigma &= [\{a, c\}]_s^\sigma = [\{b, c\}]_s^\sigma = [\{a, b, c\}]_s^\sigma = \{\{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}. \end{aligned}$$

The resulting lattice of strong equivalence classes is depicted below:



According to Theorem 18, the characterization logic $\sigma' : 2^{\mathcal{L}} \rightarrow 2^{2^{\mathcal{L}}}$ assigns as follows:

$$\begin{aligned}
\sigma'(\emptyset) &= \bigcup_{\substack{S \in 2^{\mathcal{L}}, \\ [\emptyset]_s^\sigma \sqsubseteq [S]_s^\sigma}} [S]_s^\sigma = \bigcup_{S \in 2^{\mathcal{L}}} [S]_s^\sigma = 2^{\mathcal{L}} \\
\sigma'(\{a\}) &= \bigcup_{\substack{S \in 2^{\mathcal{L}}, \\ [\{a\}]_s^\sigma \sqsubseteq [S]_s^\sigma}} [S]_s^\sigma = [\{a\}]_s^\sigma \cup [\{a, b, c\}]_s^\sigma = \{\{a\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\} \\
\sigma'(\{b\}) &= \bigcup_{\substack{S \in 2^{\mathcal{L}}, \\ [\{b\}]_s^\sigma \sqsubseteq [S]_s^\sigma}} [S]_s^\sigma = [\{b\}]_s^\sigma \cup [\{a, b, c\}]_s^\sigma = \{\{b\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\} \\
\sigma'(\{c\}) &= \bigcup_{\substack{S \in 2^{\mathcal{L}}, \\ [\{c\}]_s^\sigma \sqsubseteq [S]_s^\sigma}} [S]_s^\sigma = [\{c\}]_s^\sigma \cup [\{a, b, c\}]_s^\sigma = \{\{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\} \\
\sigma'(\{a, b, c\}) &= \bigcup_{\substack{S \in 2^{\mathcal{L}}, \\ [\{a, b, c\}]_s^\sigma \sqsubseteq [S]_s^\sigma}} [S]_s^\sigma = [\{a, b, c\}]_s^\sigma = \{\{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}
\end{aligned}$$

Now it holds for example that

$$\begin{aligned}
&\sigma'(\{a\}) \cap \sigma'(\{b\}) \\
&= \{\{a\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\} \cap \{\{b\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\} \\
&= \{\{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\} \\
&= \sigma'(\{a, b\}) \\
&= \sigma'(\{a\} \cup \{b\}) \quad \diamond
\end{aligned}$$

3.2. Finite-Theory Characterization Logics

In the field of knowledge representation it is a common assumption that knowledge bases are finite. This is indeed not overly limiting, as finite knowledge bases will be most relevant for practical purposes. The following definition translates this assumption into our setting: the *finite-theory version* of a given logic (or simply, a *finite-theory logic*) considers only the finite knowledge bases of a language.

Definition 10. Given a full logic (\mathcal{L}, σ) , the finite-theory version $(\mathcal{L}, \sigma_{\text{fin}})$ of (\mathcal{L}, σ) is defined by the semantics

$$\sigma_{\text{fin}} : (2^{\mathcal{L}})_{\text{fin}} \rightarrow \sigma(2^{\mathcal{L}}) \quad \text{with} \quad \sigma_{\text{fin}}(T) = \sigma(T)$$

where $(2^{\mathcal{L}})_{\text{fin}} = \{T \in 2^{\mathcal{L}} \mid T \text{ is finite}\}$. \(\diamond\)

For finite-theory restrictions of logics, we adequately relax our requirements on characterization logics.

Definition 11. Let (\mathcal{L}, σ) be a full logic and $(\mathcal{L}, \sigma_{\text{fin}})$ its finite-theory version. We say that $(\mathcal{L}, \sigma'_{\text{fin}})$ is a *finite-theory characterization logic* for (\mathcal{L}, σ) if and only if:

1. $\forall T_1, T_2 \in (2^{\mathcal{L}})_{\text{fin}} : \sigma'_{\text{fin}}(T_1) = \sigma'_{\text{fin}}(T_2)$ iff $[T_1]_s^{\sigma_{\text{fin}}} = [T_2]_s^{\sigma_{\text{fin}}}$; (finite-theory characterization)
2. $\forall T_1, T_2 \in (2^{\mathcal{L}})_{\text{fin}} : \sigma'_{\text{fin}}(T_1 \cup T_2) = \sigma'_{\text{fin}}(T_1) \cap \sigma'_{\text{fin}}(T_2)$. (finite-theory intersection)

The second item requires binary intersection only; this is due to the fact that arbitrary unions of (finite) theories are not necessarily finite, and thus their semantics might not be well-defined.

As we did in the general case before, we first analyze the algebraic structure of the resulting model theories. We show that the model theory of any finite-theory characterization logic forms a lattice, that is, a partially ordered set where each non-empty finite subset has both a greatest lower bound and a least upper bound. (This is in contrast to *complete* lattices in the general case.) The proof is, although similar in procedure, slightly more involved than in the general case.

Proposition 19. *Let $(\mathcal{L}, \sigma_{\text{fin}})$ be a finite-theory logic with characterization logic $(\mathcal{L}, \sigma'_{\text{fin}})$. Denoting $\mathcal{K} = \{\sigma'_{\text{fin}}(T) \mid T \in (2^{\mathcal{L}})_{\text{fin}}\}$, the pair (\mathcal{K}, \subseteq) is a lattice where glb and lub are given such that for all $K_1, K_2 \in \mathcal{K}$:*

$$K_1 \wedge K_2 = K_1 \cap K_2 \text{ and } K_1 \vee K_2 = \bigwedge \{K_1, K_2\}^u$$

where $\{K_1, K_2\}^u = \{K \in \mathcal{K} \mid K_1 \subseteq K, K_2 \subseteq K\}$.

Proof. Let $K_1, K_2 \in \mathcal{K}$. Clearly there exist finite $T_1, T_2 \subseteq \mathcal{L}$ with $\sigma'_{\text{fin}}(T_1) = K_1$ as well as $\sigma'_{\text{fin}}(T_2) = K_2$.

glb: Obviously $T_1 \cup T_2 \in (2^{\mathcal{L}})_{\text{fin}}$. Therefore, it follows that

$$K_1 \wedge K_2 = \sigma'_{\text{fin}}(T_1) \wedge \sigma'_{\text{fin}}(T_2) = \sigma'_{\text{fin}}(T_1) \cap \sigma'_{\text{fin}}(T_2) = \sigma'_{\text{fin}}(T_1 \cup T_2) \in \mathcal{K}$$

lub: It follows from the finite intersection property that σ'_{fin} is antimonotone. Now $\emptyset \subseteq T_1$ implies that $K_1 = \sigma'_{\text{fin}}(T_1) \subseteq \sigma'_{\text{fin}}(\emptyset)$ and likewise for T_2 . Thus $\sigma'_{\text{fin}}(\emptyset)$ is an upper bound of K_1 and K_2 guaranteeing the non-emptiness of $\{K_1, K_2\}^u$. We will now show that $\{K_1, K_2\}^u$ is finite. Clearly both T_1 and T_2 have only finitely many subsets T'_1 and T'_2 . For each of these subsets, σ'_{fin} being antimonotone means that $\emptyset \subseteq T'_1 \subseteq T_1$ implies $\sigma'_{\text{fin}}(T_1) \subseteq \sigma'_{\text{fin}}(T'_1) \subseteq \sigma'_{\text{fin}}(\emptyset)$. Thus both K_1 and K_2 have only finitely many supersets in \mathcal{K} , that is, the sets K_1^\uparrow and K_2^\uparrow are finite. Hence, $\{K_1, K_2\}^u \subseteq K_1^\uparrow \cup K_2^\uparrow$ is finite. Consequently, $\{K_1, K_2\}^u$ is a finite, non-empty subset of \mathcal{K} . It therefore possesses a greatest lower bound $K_l = \bigwedge \{K_1, K_2\}^u \in \mathcal{K}$. Since $\{K_1, K_2\}^u$ is closed under intersection we have that $K_l = \bigwedge \{K_1, K_2\}^u = \bigcap \{K_1, K_2\}^u \in \{K_1, K_2\}^u$ is the least element of $\{K_1, K_2\}^u$ and therefore the least upper bound of K_1 and K_2 concluding the proof. \square

As before, we can show (with reasonable effort) that finite-theory characterization logics are unique up to isomorphism.

Theorem 20. Let $(\mathcal{L}, \sigma_{\text{fin}})$ be a finite-theory logic having two finite-theory characterization logics $(\mathcal{L}, \sigma'_{\text{fin}})$ and $(\mathcal{L}, \sigma''_{\text{fin}})$. Denoting the two carrier sets by $\mathcal{K}' = \{\sigma'_{\text{fin}}(T) \mid T \in (2^{\mathcal{L}})_{\text{fin}}\}$ and $\mathcal{K}'' = \{\sigma''_{\text{fin}}(T) \mid T \in (2^{\mathcal{L}})_{\text{fin}}\}$, the lattices $(\mathcal{K}', \subseteq)$ and $(\mathcal{K}'', \subseteq)$ are isomorphic.

Proof. We provide a bijection $\phi : \mathcal{K}' \rightarrow \mathcal{K}''$ such that for all $K_1, K_2 \in \mathcal{K}'$, we find that $\phi(K_1 \wedge K_2) = \phi(K_1) \wedge \phi(K_2)$ and $\phi(K_1 \vee K_2) = \phi(K_1) \vee \phi(K_2)$. Let $K \in \mathcal{K}'$. By definition, there exists a finite $T \subseteq \mathcal{L}$ with $\sigma'_{\text{fin}}(T) = K$. Define $\phi(K) = \sigma''_{\text{fin}}(T)$.

ϕ is bijective: The proof is as in the general case.

ϕ is structure-preserving: Let $K_1, K_2 \in \mathcal{K}'$. Clearly there exist $T_1, T_2 \subseteq \mathcal{L}$ s.t. $\sigma'_{\text{fin}}(T_1) = K_1$ and $\sigma'_{\text{fin}}(T_2) = K_2$. We have

$$\begin{aligned}
\phi(K_1 \wedge K_2) &= \phi(K_1 \cap K_2) && (\text{Def. } \wedge) \\
&= \phi(\sigma'_{\text{fin}}(T_1) \cap \sigma'_{\text{fin}}(T_2)) && (\sigma'_{\text{fin}} \text{ is onto } \mathcal{K}') \\
&= \phi(\sigma'_{\text{fin}}(T_1 \cup T_2)) && (\text{intersection } \sigma'_{\text{fin}}) \\
&= \sigma''_{\text{fin}}(T_1 \cup T_2) && (\text{Def. } \phi) \\
&= \sigma''_{\text{fin}}(T_1) \cap \sigma''_{\text{fin}}(T_2) && (\text{intersection } \sigma''_{\text{fin}}) \\
&= \phi(\sigma'_{\text{fin}}(T_1)) \cap \phi(\sigma'_{\text{fin}}(T_2)) && (\text{Def. } \phi) \\
&= \phi(K_1) \cap \phi(K_2) && (\text{assumption}) \\
&= \phi(K_1) \wedge \phi(K_2) && (\text{Def. } \wedge)
\end{aligned}$$

With reasonable effort, we can also show that

$$\begin{aligned}
& \phi(K_1 \vee K_2) \\
&= \phi\left(\bigwedge \{K_1, K_2\}^u\right) \\
&\quad (\text{Def. } \vee) \\
&= \bigwedge \phi(\{K_1, K_2\}^u) \\
&\quad (\phi \text{ preserves } \wedge) \\
&= \bigwedge \phi(\{K \in \mathcal{K}' \mid K_1 \subseteq K, K_2 \subseteq K\}) \\
&\quad (\text{Def. } \cdot^u) \\
&= \bigwedge \{\phi(K) \mid K \in \mathcal{K}', K_1 \subseteq K, K_2 \subseteq K\} \\
&\quad (\text{notation}) \\
&= \bigwedge \{\phi(\sigma'_{\text{fin}}(T)) \mid T \in (2^{\mathcal{L}})_{\text{fin}}, \sigma'_{\text{fin}}(T_1) \subseteq \sigma'_{\text{fin}}(T), \sigma'_{\text{fin}}(T_2) \subseteq \sigma'_{\text{fin}}(T)\} \\
&\quad (\sigma'_{\text{fin}} \text{ is onto } \mathcal{K}') \\
&= \bigwedge \{\phi(\sigma'_{\text{fin}}(T)) \mid T \in (2^{\mathcal{L}})_{\text{fin}}, \sigma'_{\text{fin}}(T_1) \cap \sigma'_{\text{fin}}(T) = \sigma'_{\text{fin}}(T_1), \sigma'_{\text{fin}}(T_2) \cap \sigma'_{\text{fin}}(T) = \sigma'_{\text{fin}}(T_2)\} \\
&\quad (\text{elementary}) \\
&= \bigwedge \{\phi(\sigma'_{\text{fin}}(T)) \mid T \in (2^{\mathcal{L}})_{\text{fin}}, \sigma'_{\text{fin}}(T_1 \cup T) = \sigma'_{\text{fin}}(T_1), \sigma'_{\text{fin}}(T_2 \cup T) = \sigma'_{\text{fin}}(T_2)\} \\
&\quad (\text{intersection } \sigma'_{\text{fin}}) \\
&= \bigwedge \{\phi(\sigma'_{\text{fin}}(T)) \mid T \in (2^{\mathcal{L}})_{\text{fin}}, [T_1 \cup T]_s^\sigma = [T_1]_s^\sigma, [T_2 \cup T]_s^\sigma = [T_2]_s^\sigma\} \\
&\quad (\text{characterization } \sigma'_{\text{fin}}) \\
&= \bigwedge \{\phi(\sigma''_{\text{fin}}(T)) \mid T \in (2^{\mathcal{L}})_{\text{fin}}, \sigma''_{\text{fin}}(T_1 \cup T) = \sigma''_{\text{fin}}(T_1), \sigma''_{\text{fin}}(T_2 \cup T) = \sigma''_{\text{fin}}(T_2)\} \\
&\quad (\text{characterization } \sigma''_{\text{fin}}) \\
&= \bigwedge \{\sigma''_{\text{fin}}(T) \mid T \in (2^{\mathcal{L}})_{\text{fin}}, \sigma''_{\text{fin}}(T_1) \cap \sigma''_{\text{fin}}(T) = \sigma''_{\text{fin}}(T_1), \sigma''_{\text{fin}}(T_2) \cap \sigma''_{\text{fin}}(T) = \sigma''_{\text{fin}}(T_2)\} \\
&\quad (\text{intersection } \sigma''_{\text{fin}}) \\
&= \bigwedge \{\sigma''_{\text{fin}}(T) \mid T \in (2^{\mathcal{L}})_{\text{fin}}, \sigma''_{\text{fin}}(T_1) \subseteq \sigma''_{\text{fin}}(T), \sigma''_{\text{fin}}(T_2) \subseteq \sigma''_{\text{fin}}(T)\} \\
&\quad (\text{elementary}) \\
&= \bigwedge \{K \mid K \in \mathcal{K}'', \sigma''_{\text{fin}}(T_1) \subseteq K, \sigma''_{\text{fin}}(T_2) \subseteq K\} \\
&\quad (\sigma''_{\text{fin}} \text{ is onto } \mathcal{K}'') \\
&= \bigwedge \{\sigma''_{\text{fin}}(T_1), \sigma''_{\text{fin}}(T_2)\}^u \\
&\quad (\text{Def. } \cdot^u) \\
&= \sigma''_{\text{fin}}(T_1) \vee \sigma''_{\text{fin}}(T_2) \\
&\quad (\text{Def. } \vee) \\
&= \phi(\sigma'_{\text{fin}}(T_1)) \vee \phi(\sigma'_{\text{fin}}(T_2)) \\
&\quad (\text{Def. } \phi) \\
&= \phi(K_1) \vee \phi(K_2) \\
&\quad (\text{assumption})
\end{aligned}$$

Thus ϕ is a structure-preserving bijection from \mathcal{K}' to \mathcal{K}'' , and the two lattices are isomorphic. \square

As a final ingredient of our main result, we show one more property of the partial ordering \sqsubseteq : Also in the finite case, the mapping from theories to their strong equivalence classes preserves lattice joins.

Lemma 21. *Let $(\mathcal{L}, \sigma_{\text{fin}})$ be a finite-theory logic. For $T_1, T_2 \subseteq (2^{\mathcal{L}})_{\text{fin}}$, we find that $[T_1]_s^{\sigma_{\text{fin}}} \sqcup [T_2]_s^{\sigma_{\text{fin}}} = [T_1 \cup T_2]_s^{\sigma_{\text{fin}}}$, that is, the class $[T_1 \cup T_2]_s^{\sigma_{\text{fin}}}$ is the \sqsubseteq -least upper bound of $\{[T_1]_s^{\sigma_{\text{fin}}}, [T_2]_s^{\sigma_{\text{fin}}}\}$.*

Proof. It is immediate that $T_1 \cup T_2 \in (2^{\mathcal{L}})_{\text{fin}}$, hence $\sigma_{\text{fin}}(T_1 \cup T_2)$ and thus $[T_1 \cup T_2]_s^{\sigma_{\text{fin}}}$ are well-defined. Furthermore, $[T_1 \cup T_2]_s^{\sigma_{\text{fin}}}$ is clearly an upper bound of $\{[T_1]_s^{\sigma_{\text{fin}}}, [T_2]_s^{\sigma_{\text{fin}}}\}$ due to the obvious witnesses. It remains to show that it is the least upper bound. Let $T \in (2^{\mathcal{L}})_{\text{fin}}$ be such that $[T]_s^{\sigma_{\text{fin}}}$ is an upper bound of $\{[T_1]_s^{\sigma_{\text{fin}}}, [T_2]_s^{\sigma_{\text{fin}}}\}$. Thus there exist $T', T'' \in [T]_s^{\sigma_{\text{fin}}}$, $T'_1 \in [T_1]_s^{\sigma_{\text{fin}}}$, and $T'_2 \in [T_2]_s^{\sigma_{\text{fin}}}$ such that $T'_1 \subseteq T'$ and $T'_2 \subseteq T''$. Now we use $[T_1]_s^{\sigma_{\text{fin}}} = [T'_1]_s^{\sigma_{\text{fin}}}$ to deduce

$$\begin{aligned} [T_1]_s^{\sigma_{\text{fin}}} &= [T'_1]_s^{\sigma_{\text{fin}}} \\ \implies [T_1 \cup T']_s^{\sigma_{\text{fin}}} &= [T'_1 \cup T']_s^{\sigma_{\text{fin}}} && \text{(Lemma 14, Item 1)} \\ \implies [T_1 \cup T']_s^{\sigma_{\text{fin}}} &= [T']_s^{\sigma_{\text{fin}}} && (T'_1 \subseteq T') \\ \implies [T_1 \cup T' \cup T'']_s^{\sigma_{\text{fin}}} &= [T' \cup T'']_s^{\sigma_{\text{fin}}} && \text{(Lemma 14, Item 1)} \\ \implies [T_1 \cup (T' \cup T'')]_s^{\sigma_{\text{fin}}} &= [T]_s^{\sigma_{\text{fin}}} && \text{(Lemma 14, Item 2)} \end{aligned}$$

Analogously, we use $[T_2]_s^{\sigma_{\text{fin}}} = [T'_2]_s^{\sigma_{\text{fin}}}$ to infer

$$\begin{aligned} [T_2]_s^{\sigma_{\text{fin}}} &= [T'_2]_s^{\sigma_{\text{fin}}} \\ \implies [T_2 \cup T'']_s^{\sigma_{\text{fin}}} &= [T'_2 \cup T'']_s^{\sigma_{\text{fin}}} && \text{(Lemma 14, Item 1)} \\ \implies [T_2 \cup T'']_s^{\sigma_{\text{fin}}} &= [T'']_s^{\sigma_{\text{fin}}} && (T'_2 \subseteq T'') \\ \implies [T_2 \cup T'' \cup T']_s^{\sigma_{\text{fin}}} &= [T'' \cup T']_s^{\sigma_{\text{fin}}} && \text{(Lemma 14, Item 1)} \\ \implies [T_2 \cup (T' \cup T'')]_s^{\sigma_{\text{fin}}} &= [T]_s^{\sigma_{\text{fin}}} && \text{(Lemma 14, Item 2)} \end{aligned}$$

In combination, $[T_1 \cup (T' \cup T'')]_s^{\sigma_{\text{fin}}} = [T]_s^{\sigma_{\text{fin}}} = [T_2 \cup (T' \cup T'')]_s^{\sigma_{\text{fin}}}$. Finally,

$$\begin{aligned} [T_1 \cup (T' \cup T'')]_s^{\sigma_{\text{fin}}} &= [T_2 \cup (T' \cup T'')]_s^{\sigma_{\text{fin}}} \\ \implies [T_1 \cup (T' \cup T'') \cup T_2]_s^{\sigma_{\text{fin}}} &= [T_2 \cup (T' \cup T'')]_s^{\sigma_{\text{fin}}} && \text{(Lemma 14, Item 1)} \\ \implies [T_1 \cup T_2 \cup (T' \cup T'')]_s^{\sigma_{\text{fin}}} &= [T]_s^{\sigma_{\text{fin}}} && \text{(above)} \end{aligned}$$

In particular, $T_1 \cup T_2 \cup (T' \cup T'') \in [T]_s^{\sigma_{\text{fin}}}$, which shows that $[T_1 \cup T_2]_s^{\sigma_{\text{fin}}} \sqsubseteq [T]_s^{\sigma_{\text{fin}}}$ and concludes the proof. \square

The following theorem shows that any logic possesses a finite-theory characterization logic. This means that the most important case for knowledge representation behaves well in the sense that characterization logics always exist. The characterization logic defined below makes use of the lattice-theoretic concept of a *principal filter* given by the up-set $(\cdot)^\uparrow$ of a specific element, the set of all elements that are at or “above” the element in the lattice [32]. More specifically, in this case, for a theory $T \in (2^{\mathcal{L}})_{\text{fin}}$, the principal filter of its strong equivalence class $[T]_s^{\sigma_{\text{fin}}}$ with respect to the ordering \sqsubseteq is the set

$$([T]_s^{\sigma_{\text{fin}}})^\uparrow = \{[S]_s^{\sigma_{\text{fin}}} \mid S \in (2^{\mathcal{L}})_{\text{fin}}, [T]_s^{\sigma_{\text{fin}}} \sqsubseteq [S]_s^{\sigma_{\text{fin}}}\}$$

The characterization semantics then simply assigns a theory T to the set of theories contained in some class of that filter.

Theorem 22. *Let (\mathcal{L}, σ) be a full logic. Then a finite-theory characterization logic for (\mathcal{L}, σ) is given by $(\mathcal{L}, \sigma'_{\text{fin}})$ with*

$$\sigma'_{\text{fin}} : (2^{\mathcal{L}})_{\text{fin}} \rightarrow 2^{2^{\mathcal{L}}}, \quad T \mapsto \bigcup ([T]_s^{\sigma_{\text{fin}}})^{\uparrow}$$

Proof. We first observe the following equivalent formulations for $S, T \in (2^{\mathcal{L}})_{\text{fin}}$:

$$S \in \sigma'_{\text{fin}}(T) \iff S \in \bigcup ([T]_s^{\sigma_{\text{fin}}})^{\uparrow} \iff [S]_s^{\sigma_{\text{fin}}} \in ([T]_s^{\sigma_{\text{fin}}})^{\uparrow} \iff [T]_s^{\sigma_{\text{fin}}} \sqsubseteq [S]_s^{\sigma_{\text{fin}}} \quad (3)$$

As a special case, this entails $T \in \sigma'_{\text{fin}}(T)$ for every $T \in (2^{\mathcal{L}})_{\text{fin}}$. Now for the main proof, where we show finite-theory characterization and finite-theory intersection in accordance with Definition 11.

1. Let $T_1, T_2 \in (2^{\mathcal{L}})_{\text{fin}}$. We have to show $\sigma'_{\text{fin}}(T_1) = \sigma'_{\text{fin}}(T_2)$ iff $[T_1]_s^{\sigma_{\text{fin}}} = [T_2]_s^{\sigma_{\text{fin}}}$.
 - if: Let $[T_1]_s^{\sigma_{\text{fin}}} = [T_2]_s^{\sigma_{\text{fin}}}$. Then clearly $([T_1]_s^{\sigma_{\text{fin}}})^{\uparrow} = ([T_2]_s^{\sigma_{\text{fin}}})^{\uparrow}$ whence immediately $\sigma'_{\text{fin}}(T_1) = \sigma'_{\text{fin}}(T_2)$ by definition.
 - only if: Let $\sigma'_{\text{fin}}(T_1) = \sigma'_{\text{fin}}(T_2)$. Then by $T_1 \in \sigma'_{\text{fin}}(T_1)$, we have $T_1 \in \sigma'_{\text{fin}}(T_2)$. By Equation (3), this yields $[T_2]_s^{\sigma_{\text{fin}}} \sqsubseteq [T_1]_s^{\sigma_{\text{fin}}}$. Symmetrically, we can show $[T_1]_s^{\sigma_{\text{fin}}} \sqsubseteq [T_2]_s^{\sigma_{\text{fin}}}$. In combination, $[T_1]_s^{\sigma_{\text{fin}}} = [T_2]_s^{\sigma_{\text{fin}}}$ as desired.
2. Let $T_1, T_2 \in (2^{\mathcal{L}})_{\text{fin}}$. We have to show $\sigma'_{\text{fin}}(T_1 \cup T_2) = \sigma'_{\text{fin}}(T_1) \cap \sigma'_{\text{fin}}(T_2)$.
 - \sqsubseteq : Let $T \in \sigma'_{\text{fin}}(T_1 \cup T_2)$. By Equation (3), we get $[T_1 \cup T_2]_s^{\sigma_{\text{fin}}} \sqsubseteq [T]_s^{\sigma_{\text{fin}}}$. Now clearly $[T_1]_s^{\sigma_{\text{fin}}} \sqsubseteq [T_1 \cup T_2]_s^{\sigma_{\text{fin}}}$ and by transitivity also $[T_1]_s^{\sigma_{\text{fin}}} \sqsubseteq [T]_s^{\sigma_{\text{fin}}}$. By Equation (3) we get $T \in \sigma'_{\text{fin}}(T_1)$. Symmetrically, we can show $T \in \sigma'_{\text{fin}}(T_2)$, and $T \in \sigma'_{\text{fin}}(T_1) \cap \sigma'_{\text{fin}}(T_2)$ follows.
 - \supseteq : Let $T \in \sigma'_{\text{fin}}(T_1) \cap \sigma'_{\text{fin}}(T_2)$. Thus in particular $T \in \sigma'_{\text{fin}}(T_1)$. By applying Equation (3) it follows that $[T_1]_s^{\sigma_{\text{fin}}} \sqsubseteq [T]_s^{\sigma_{\text{fin}}}$; symmetrically, we obtain $[T_2]_s^{\sigma_{\text{fin}}} \sqsubseteq [T]_s^{\sigma_{\text{fin}}}$. Thus $[T]_s^{\sigma_{\text{fin}}}$ is an upper bound of $\{[T_1]_s^{\sigma_{\text{fin}}}, [T_2]_s^{\sigma_{\text{fin}}}\}$. By Lemma 21, $[T_1 \cup T_2]_s^{\sigma_{\text{fin}}}$ is the \sqsubseteq -least upper bound of $\{[T_1]_s^{\sigma_{\text{fin}}}, [T_2]_s^{\sigma_{\text{fin}}}\}$, therefore $[T_1 \cup T_2]_s^{\sigma_{\text{fin}}} \sqsubseteq [T]_s^{\sigma_{\text{fin}}}$. Again using Equation (3), we get $T \in \sigma'_{\text{fin}}(T_1 \cup T_2)$. \square

Intuitively, this canonical construction of a characterization semantics σ'_{fin} is similar to Herbrand interpretations in first-order logic, in that we re-use (sets of) syntactical elements when defining a semantics with the ultimate aim of showing a fundamental property of the logic.

4. Applying Canonical Constructions to Nonmonotonic Formalisms

In the previous section we have seen that (under certain conditions) the existence of characterization logics for knowledge representation formalisms are guaranteed. These results are achieved by defining in a sense Herbrand-style canonical construction. More precisely, the characterization semantics of the new logic is defined in terms of certain unions of strong equivalence classes of the original language. Such a characterization semantics is usually far from being intuitive or self-explanatory. The intended role of this semantics was to serve as a witness for the existence of characterization logics. Nevertheless, we will discuss the application of our general, abstract results to some of the formalisms presented in Example 1. We start with abstract argumentation theory, which is a vibrant as well as immensely growing research area in AI [14]. Surprisingly, we get a meaningful result very similar to the recently introduced Dung logics [36].

4.1. Abstract Argumentation Theory

We start with a brief introduction to Dung's argumentation theory (cf. [37] for a recent and comprehensive overview).

An *argumentation framework* (AF) is a pair $F = (A, R)$ such that $R \subseteq A \times A$. Although there exists some work on unrestricted AFs [38, 39] it is common to assume that A , the set of arguments, is a finite subset of a fixed infinite background set \mathcal{U} . Let us denote the class of all finite AFs by AF_{fin} . An (extension-based) *argumentation semantics* is a function $\rho : AF_{\text{fin}} \rightarrow 2^{2^{\mathcal{U}}}$ where elements of $\rho(F)$ are called ρ -*extensions* of F . The most prominent one is stable semantics (abbreviated by *stb*) which was already defined by Dung in 1995. A set E is a *stb*-extension of F if 1. there are no $a, b \in A$, such that $(a, b) \in R$ (*conflict-freeness*) and 2. for any $c \in A \setminus E$, there is an $a \in A$ with $(a, c) \in R$ (*full range*).

We proceed with some notational conventions and the precise definition of strong equivalence in case of AFs. The union $F \dot{\cup} G$ as well as subset-relation $F \dot{\subseteq} G$ of two AFs is understood to be pointwise, that is, $(A_1, R_1) \dot{\cup} (A_2, R_2) = (A_1 \cup A_2, R_1 \cup R_2)$, and, similarly, $(A_1, R_1) \dot{\subseteq} (A_2, R_2)$ if and only if $A_1 \subseteq A_2$ and $R_1 \subseteq R_2$.

Definition 12. Given an argumentation semantics ρ , two AFs F and G are *strongly ρ -equivalent* iff for any $H \in AF_{\text{fin}}$, $\rho(F \dot{\cup} H) = \rho(G \dot{\cup} H)$. In this case, we write $F \equiv_s^\rho G$. \diamond

The first work regarding characterizing strong equivalence for AFs was presented by Oikarinen and Woltran [15]. It turned out that deciding this notion is deeply linked to the syntax of AFs. In general, any argument being part of an AF may contribute towards future extensions. However, for each semantics, there are patterns of redundant attacks captured by so-called *kernels*. Formally, a kernel is a function $k : AF_{\text{fin}} \rightarrow AF_{\text{fin}}$ where $k(F) = F^k$ is obtained from F by deleting certain redundant attacks. We briefly recall the following definition and characterization theorem. An exhaustive overview on well-known semantics, further equivalence notions and their characterization can be found in the Handbook of Formal Argumentation [40].

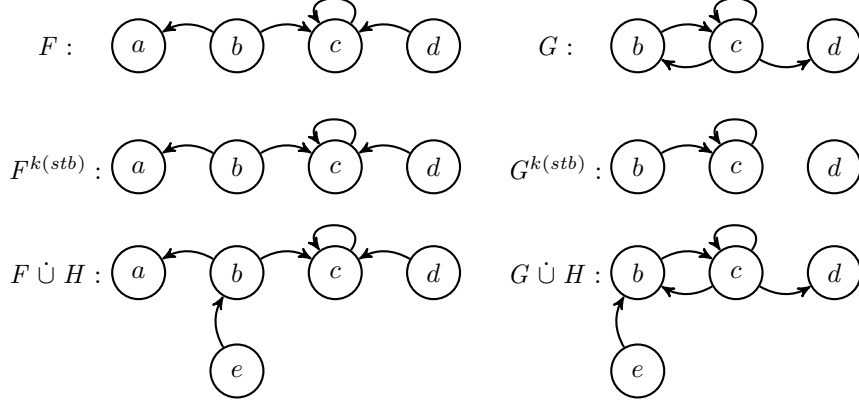
Definition 13. Given an AF $F = (A, R)$, the *stb-kernel* $F^{k(\text{stb})} = (A, R^{k(\text{stb})})$ is defined by $R^{k(\text{stb})} = R \setminus \{(a, b) \mid a \neq b \wedge (a, a) \in R\}$. \diamond

Theorem 23 ([15]). For two AFs F, G we have:

$$F \equiv_s^{\text{stb}} G \Leftrightarrow F^{k(\text{stb})} = G^{k(\text{stb})}$$

Let us illustrate the introduced concepts with an example.

Example 11. Consider the following AFs F and G . Both possess the same stable extensions, namely $\text{stb}(F) = \text{stb}(G) = \{\{b, d\}\}$. The AFs $F^{k(\text{stb})}$ and $G^{k(\text{stb})}$ depicted in the second line are the associated stable kernels of the initial frameworks (cf. Definition 13). Obviously, $F^{k(\text{stb})} \neq G^{k(\text{stb})}$ which proves that F and G are not strongly *stb*-equivalent (Theorem 23).



The last line illustrates a witnessing expansion, namely adding $H = (\{e, b\}, \{(e, b)\})$ to both frameworks simultaneously. In fact, $stb(F \dot{\cup} H) = \{\{a, d, e\}\} \neq \emptyset = stb(G \dot{\cup} H)$.

Let us consider now argumentation theory in the general setup. In Example 1 we have seen that the embedding of abstract argumentation in our setting is a bit more involved than in case of propositional logic, logic programs or default logic. The main reason for this is that in contrast to the other considered formalisms we have that abstract argumentation frameworks possess two sorts of building blocks, namely arguments and attacks. Moreover, the latter are dependent since adding attacks requires the presence of the corresponding arguments. In order to cast AFs into our general setup we have to have theories which correspond to AFs, s.t. the standard set union \cup of such theories correspond to $\dot{\cup}$ on the AF-level. Moreover, the semantics of theories has to correspond to the argumentation semantics of the associated AFs.

We start with the introduction of a ρ -logic which formally captures a specific argumentation semantics ρ on the level of theories.

Definition 14. Let \mathcal{U} be a background set of arguments and ρ be an AF semantics. A ρ -logic is a triple $(\mathcal{L}_{AF}, \mathcal{I}, \sigma_\rho)$ where $\mathcal{L}_{AF} = \{(\{a\}, \emptyset), (\{a, b\}, \{(a, b)\}) \mid a, b \in \mathcal{U}\}$, $\mathcal{I} = 2^{\mathcal{U}}$ and $\sigma_\rho : 2^{\mathcal{L}_{AF}} \rightarrow 2^{\mathcal{I}}$ with $\sigma_\rho(T) = \rho(\dot{\bigcup}_{t \in T} t)$. \diamond

The representational issue implies that two different theories may represent the same framework which causes some additional effort. More precisely, in what follows it will be a typical task to show that the presented results are independent of the concrete representation of a certain framework. Let us start with an illustrating example.

Example 12. Let $T = \{(\{b, c\}, \{(b, c)\}), (\{c, b\}, \{(c, b)\}), (\{c, d\}, \{(c, d)\}), (\{c\}, \{(c, c)\})\}$ and $S = T \cup \{(\{b\}, \emptyset), (\{c\}, \emptyset), (\{d\}, \emptyset)\}$. Observe that $\dot{\bigcup}_{t \in T} t = \dot{\bigcup}_{s \in S} s = G$ as depicted in Example 11. Moreover by definition we have $\sigma_{stb}(T) = \sigma_{stb}(S) = stb(G) = \{\{b, d\}\}$. \diamond

The following functions (restricted to the finite case) will be frequently used. First, we define the *associated AF* of a given theory via $AF : (2^{\mathcal{L}_{AF}})_{\text{fin}} \rightarrow AF_{\text{fin}}$ where $T \mapsto \dot{\bigcup}_{t \in T} t$. As already discussed the function $AF(\cdot)$ is not injective as demonstrated in Example 12. Secondly, the *canonical representation* of a given AF is defined by $C : AF_{\text{fin}} \rightarrow (2^{\mathcal{L}_{AF}})_{\text{fin}}$ where (A, R) is represented by the \mathcal{L}_{AF} -theory $\{(\{a\}, \emptyset) \mid a \in A\} \cup \{(\{a, b\}, \{(a, b)\}) \mid (a, b) \in R\}$. Observe that for any AF H , we find $AF(C(H)) = H$. Moreover, regarding Examples 11 and 12 we have $C(G) = S \neq T$.

Note that the assumption of finiteness of AFs can be reflected by considering the finite-theory versions of ρ -logics. Before applying our canonical construction presented in Theorem 22 we have

to ensure that a constructed ρ -logic correctly reflects AFs under semantics ρ . We start with two simple properties showing that the concrete representation (of an AF via a theory) is not “seen” by set union as well as semantics σ_ρ .

Proposition 24. *Let $(\mathcal{L}_{AF}, \mathcal{I}, \sigma_\rho)$ be a ρ -logic and consider any theories $S, T \subseteq \mathcal{L}_{AF}$.*

1. $AF(S \cup T) = AF(S) \dot{\cup} AF(T)$ and
2. $\sigma_\rho(S \cup T) = \rho(AF(S) \dot{\cup} AF(T))$.

Proof. 1. Both statements can be easily seen. Consider the following equations.

$$\begin{aligned}
& AF(S \cup T) \\
&= \dot{\bigcup}_{u \in S \cup T} u && \text{(Definition AF)} \\
&= \dot{\bigcup}_{s \in S} s \dot{\cup} \dot{\bigcup}_{t \in T} t && \text{(associativity } \dot{\bigcup} \text{)} \\
&= AF(S) \dot{\cup} AF(T) && \text{(Definition AF)}
\end{aligned}$$

2.

$$\begin{aligned}
& \sigma_\rho(S \cup T) \\
&= \rho\left(\dot{\bigcup}_{u \in S \cup T} u\right) && \text{(Definition } \sigma_\rho \text{)} \\
&= \rho(AF(S \cup T)) && \text{(Definition AF)} \\
&= \rho(AF(S) \dot{\cup} AF(T)) && \text{(Item 1)} \quad \square
\end{aligned}$$

The following theorem shows that two \mathcal{L}_{AF} -theories S and T are strongly equivalent under σ_ρ if and only if the AFs $AF(S)$ and $AF(T)$ are strongly equivalent under ρ (denoted by $AF(S) \equiv_s^\rho AF(T)$). We mention that the theorem does not require finiteness and is thus valid for arbitrary cardinalities of theories as well as AFs.

Theorem 25. *Let $(\mathcal{L}_{AF}, \mathcal{I}, \sigma_\rho)$ be a ρ -logic. For $S, T \subseteq \mathcal{L}_{AF}$ we have*

$$AF(S) \equiv_s^\rho AF(T) \text{ iff } [S]_s^{\sigma_\rho} = [T]_s^{\sigma_\rho}.$$

Proof. “ \implies ”: Let $AF(S) \equiv_s^\rho AF(T)$ and $V \in 2^{\mathcal{L}_{AF}}$. We have to show that both theories are strongly equivalent, i.e. $\sigma_\rho(S \cup V) = \sigma_\rho(T \cup V)$.

$$\begin{aligned}
& \sigma_\rho(S \cup V) \\
&= \rho(AF(S) \dot{\cup} AF(V)) && \text{(Proposition 24)} \\
&= \rho(AF(T) \dot{\cup} AF(V)) && \text{(assumption } AF(S) \equiv_s^\rho AF(T) \text{)} \\
&= \sigma_\rho(T \cup V) && \text{(Proposition 24)}
\end{aligned}$$

“ \Leftarrow ”: Let $[S]_s^{\sigma_\rho} = [T]_s^{\sigma_\rho}$ and H be an AF. We have to show that the corresponding AFs are strongly equivalent, i.e. $\rho(AF(S) \dot{\cup} H) = \rho(AF(T) \dot{\cup} H)$.

$$\begin{aligned}
& \rho(AF(S) \dot{\cup} H) \\
&= \rho(AF(S) \dot{\cup} AF(C(H))) && (AF(C(H)) = H) \\
&= \sigma_\rho(S \cup C(H)) && (\text{Proposition 24}) \\
&= \sigma_\rho(T \cup C(H)) && (\text{assumption } [S]_s^{\sigma_\rho} = [T]_s^{\sigma_\rho}) \\
&= \rho(AF(T) \dot{\cup} AF(C(H))) && (\text{Proposition 24}) \\
&= \rho(AF(T) \dot{\cup} H) && (AF(C(H)) = H) \quad \square
\end{aligned}$$

Due to Theorem 22 we are able to present finite-theory characterization logics for any ρ -logic.

Corollary 26. Let $(\mathcal{L}_{AF}, \mathcal{I}, \sigma_\rho)$ be a ρ -logic. The following logic $(\mathcal{L}_{AF}, \kappa)$ is a finite-theory characterization logic of $(\mathcal{L}_{AF}, \mathcal{I}, \sigma_\rho)$:

$$\kappa : (2^{\mathcal{L}_{AF}})_{\text{fin}} \rightarrow 2^{2^{\mathcal{L}_{AF}}}, \quad T \mapsto \bigcup \left([T]_s^{(\sigma_\rho)_{\text{fin}}} \right)^\dagger$$

So far, so good, but how can we interpret these finite-theory characterization logics in terms of argumentation theory? In other words, what is the corresponding characterization semantics on the level of pure AFs (instead of theories associated with AFs)? We extend the function AF to sets of theories as usual, namely $AF : 2^{2^{\mathcal{L}_{AF}}} \rightarrow 2^{AF_{\text{fin}}}$ where $\mathcal{T} \mapsto \{AF(T) \mid T \in \mathcal{T}\}$.¹¹ Consider the following definition. We will see that all crucial properties of κ transfer to ρ' , that is, ρ' satisfies finite intersection and furthermore, it characterizes strong equivalence under ρ .

Definition 15. Given an argumentation semantics $\rho : AF_{\text{fin}} \rightarrow 2^{2^{\mathcal{L}_{AF}}}$. We define the function $\rho' : AF_{\text{fin}} \rightarrow 2^{AF_{\text{fin}}}$ with $F \mapsto AF(\kappa(C(F)))$. \diamond

Proposition 27. For any argumentation semantics ρ and semantics ρ' as defined above we have:

1. $\forall F, G \in AF_{\text{fin}} : \rho'(F) = \rho'(G) \iff F \equiv_s^\rho G;$ (finite-theory characterization)
2. $\forall F, G \in AF_{\text{fin}} : \rho'(F \dot{\cup} G) = \rho'(F) \cap \rho'(G).$ (finite-theory intersection)

Proof. finite-theory characterization: “ \Leftarrow ”: Let $F \equiv_s^\rho G$. Hence, $AF(C(F)) \equiv_s^\rho AF(C(G))$.

Consequently, $[C(F)]_s^{\sigma_\rho} = [C(G)]_s^{\sigma_\rho}$ (Theorem 25). Since $(\mathcal{L}_{AF}, \kappa)$ is a finite-theory characterization logic of $(\mathcal{L}_{AF}, \mathcal{I}, \sigma_\rho)$ (Corollary 26) we deduce that $\kappa(C(F)) = \kappa(C(G))$. Obviously, $AF(\kappa(C(F))) = AF(\kappa(C(G)))$ which means $\rho'(F) = \rho'(G)$ (Definition 15).

“ \Rightarrow ”: We prove the contrapositive. Hence, let $F \not\equiv_s^\rho G$. This means, $AF(C(F)) \not\equiv_s^\rho AF(C(G))$ and we obtain $[C(F)]_s^{\sigma_\rho} \neq [C(G)]_s^{\sigma_\rho}$ (Theorem 25). Consequently, $\kappa(C(F)) \neq \kappa(C(G))$ since κ characterizes strong equivalence under σ_ρ (Corollary 26). Since equivalence classes are disjoint we deduce the existence of a theory U , such that (without loss of generality) $[U]_s^{\sigma_\rho} \subseteq \kappa(C(F)) \setminus \kappa(C(G))$. Consequently, $AF(U) \in AF(\kappa(C(F))) \setminus AF(\kappa(C(G)))$ since for all other representations of U' , s.t. $AF(U) = AF(U')$, we have $U' \in [U]_s^{\sigma_\rho}$ (Theorem 25). Hence, $AF(\kappa(C(F))) \neq AF(\kappa(C(G)))$ which means $\rho'(F) \neq \rho'(G)$ (Definition 15).

¹¹We do not introduce a new symbol for the new function. Which function is meant will be clear from the context.

finite-theory intersection:

$$\begin{aligned}
& \rho'(F \dot{\cup} G) \\
&= AF(\kappa(C(F \dot{\cup} G))) && \text{(Definition 15)} \\
&= AF(\kappa(C(F) \cup C(G))) && (C(F \dot{\cup} G) = C(F) \cup C(G)) \\
&= AF(\kappa(C(F)) \cap \kappa(C(G))) && \text{(intersection } \kappa) \\
&= AF(\kappa(C(F))) \cap AF(\kappa(C(G))) && \text{(can be seen)} \\
&= \rho'(F) \cap \rho'(G) && \text{(Definition 15)} \quad \square
\end{aligned}$$

Finally, we present an equivalent definition of ρ' that does not rely on ρ -logics. This means the evaluation of ρ' can be done purely on the level of AFs.

Proposition 28. *Let $\rho : AF_{\text{fin}} \rightarrow 2^{AF_{\text{fin}}}$ be a semantics and ρ' as in Definition 15. For any $F \in AF_{\text{fin}}$ we have:*

$$\rho'(F) = \bigcup_{\substack{G \in AF_{\text{fin}}, \\ F \dot{\subseteq} G}} \{H \mid H \equiv_s^\rho G\}$$

Proof.

$$\begin{aligned}
\rho'(F) &= AF(\kappa(C(F))) && \text{(Definition 15)} \\
&= AF\left(\bigcup\left([C(F)]_s^{(\sigma_\rho)_{fin}}\right)^\uparrow\right) && \text{(Corollary 26)} \\
&= AF\left(\bigcup_{\substack{S \in (2^{\mathcal{L}AF})_{fin}, \\ C(F) \subseteq S}} [S]_s^{(\sigma_\rho)_{fin}}\right) && \text{(Definition of principal filter)} \\
&= AF\left(\bigcup_{\substack{S \in (2^{\mathcal{L}AF})_{fin}, \\ C(F) \subseteq C(AF(S))}} [S]_s^{(\sigma_\rho)_{fin}}\right) && (S \subseteq C(AF(S)), [S]_s^{(\sigma_\rho)_{fin}} = [C(AF(S))]_s^{(\sigma_\rho)_{fin}}) \\
&= \bigcup_{\substack{S \in (2^{\mathcal{L}AF})_{fin}, \\ C(F) \subseteq C(AF(S))}} AF\left([S]_s^{(\sigma_\rho)_{fin}}\right) && \text{(Definition } AF(\cdot)\text{)} \\
&= \bigcup_{\substack{S \in (2^{\mathcal{L}AF})_{fin}, \\ F \dot{\subseteq} AF(S)}} AF\left([S]_s^{(\sigma_\rho)_{fin}}\right) && (C(F) \subseteq C(AF(S)) \iff F \dot{\subseteq} AF(S)) \\
&= \bigcup_{\substack{S \in (2^{\mathcal{L}AF})_{fin}, \\ F \dot{\subseteq} AF(S)}} \left\{ AF(T) \mid T \in [S]_s^{(\sigma_\rho)_{fin}} \right\} && \text{(Definition } AF(\cdot)\text{)} \\
&= \bigcup_{\substack{S \in (2^{\mathcal{L}AF})_{fin}, \\ F \dot{\subseteq} AF(S)}} \left\{ AF(T) \mid [T]_s^{(\sigma_\rho)_{fin}} = [S]_s^{(\sigma_\rho)_{fin}} \right\} && \text{(equivalence relation)} \\
&= \bigcup_{\substack{G \in AF_{fin}, \\ F \dot{\subseteq} G}} \{H \mid H \equiv_s^\rho G\} && (AF(S) = G, AF(T) = H, \text{Theorem 25})
\end{aligned}$$

□

Recently, Baumann and Brewka introduced so-called *Dung-logics* to be able to perform AGM-style revision for Dung's abstract argumentation frameworks [36]. These Dung-logics are very similar yet different from the characterization logics presented in Proposition 28. The main difference is that theories in Dung-logics are sets of AFs in contrast to the newly presented characterization logic where theories correspond to single AFs. We mention that Dung-logics possess the intersection property (Definition 16) and furthermore, two AFs F and G are strongly equivalent with respect to an argumentation semantics ρ if and only if the singletons of F and G are ordinarily equivalent with respect to the semantics introduced by Baumann and Brewka [36, Definition 3].

We start with the formal definition of a Dung-logic in case of the most prominent argument-

ation semantics, namely stable semantics.¹²

Definition 16. The Dung-logic in case of stable semantics is a pair $(AF_{\text{fin}}, \delta)$ with

$$\delta : 2^{AF_{\text{fin}}} \rightarrow 2^{AF_{\text{fin}}} \quad \mathcal{U} \mapsto \bigcap_{F \in \mathcal{U}} \text{Mod}^{k(stb)}(F)$$

whereas $\text{Mod}^{k(stb)}(F) = \{G \in AF_{\text{fin}} \mid F^{k(stb)} \dot{\subseteq} G^{k(stb)}\}$ \diamond

In order to see the similarity we consider the above definition for singletons of AFs.

Observation 29. Let $stb : AF_{\text{fin}} \rightarrow 2^{AF_{\text{fin}}}$ be stable semantics and δ as in Definition 16. For any $F \in AF_{\text{fin}}$ we have:

$$\delta(\{F\}) = \bigcup_{\substack{G \in AF_{\text{fin}}, \\ F^{k(stb)} \dot{\subseteq} G^{k(stb)}}} \{H \mid H \equiv_s^{stb} G\}$$

The only difference in comparison to ρ' is that for δ , we include all equivalence classes of AFs G whose *kernels* are in superset relation with the kernel of F , instead of having the superset relation on the AFs themselves. A further analysis will be part of future work.

4.2. Normal Logic Programs

For normal logic programs under the stable model semantics as presented in Example 1, applying Theorem 22 yields:

Corollary 30. For the finite-theory version of the logic $(\mathcal{L}_{LP}, \sigma_{stb})$ of normal logic programs under stable model semantics, a finite-theory characterization logic is given by

$$\sigma'_{stb} : (2^{\mathcal{L}_{LP}})_{\text{fin}} \rightarrow 2^{(2^{\mathcal{L}_{LP}})_{\text{fin}}}, \quad T \mapsto \bigcup ([T]_s^{\sigma_{stb}})^{\uparrow}$$

An existing, well-known characterizing semantics is given by SE-models.

Definition 17. Let P be a normal logic program over \mathcal{A} and $X \subseteq Y \subseteq \mathcal{A}$. Define semantics $\sigma_{SE} : (2^{\mathcal{L}_{LP}})_{\text{fin}} \rightarrow 2^{\mathcal{A} \times \mathcal{A}}$ by

$$T \mapsto \{(X, Y) \mid X \subseteq Y \text{ and } X, Y \in \sigma_{mod}(T^Y)\}$$

where

$$\begin{aligned} T^Y &= \{a_0 \leftarrow a_1, \dots, a_m \mid a_0 \leftarrow a_1, \dots, a_m, \sim a_{m+1}, \dots, \sim a_n \in T, a_{m+1}, \dots, a_n \notin Y\} \\ \sigma_{mod}(T) &= \{M \subseteq \mathcal{A} \mid \forall a_0 \leftarrow a_1, \dots, a_m, \sim a_{m+1}, \dots, \sim a_n \in T : \\ &\quad (a_1, \dots, a_m \in M \wedge a_{m+1}, \dots, a_n \notin M) \implies a_0 \in M\} \end{aligned} \quad \diamond$$

SE-models characterize strong equivalence of stable models [3, Theorem 1]; SE-model semantics also has the intersection property [12, Lemma 3].

Proposition 31. $(\mathcal{L}_{LP}, \sigma_{SE})$ is a characterization logic for $(\mathcal{L}_{LP}, \sigma_{stb})$.

¹²For more details consider the original paper.

Since finite-theory characterization logics are unique up to isomorphism (Theorem 20), there is a one-to-one-correspondence between the model sets given by Corollary 30 (as well as any other model set of a certain characterization logic) and sets of SE-models. More precisely, for any two logic programs $T_1, T_2 \in \mathcal{L}_{LP}$, we find $\sigma'_{stb}(T_1) \subseteq \sigma'_{stb}(T_2)$ if and only if $\sigma_{SE}(T_1) \subseteq \sigma_{SE}(T_2)$. However, we note that the set of SE-models of a finite logic program is finite, while σ'_{stb} maps logic programs to infinite model sets in general. So in the concrete case of logic programs, SE-models are much easier to work with.

5. Conclusions and Future Work

We presented a general framework for analyzing strong equivalence of knowledge representation formalisms. The framework abstracts away from all language specifics other than that knowledge bases be expressible as sets of atomic language elements. For two classes of formalisms, covered and finite-theory logics, we showed that they always possess a classical characterization logic. We called characterization logics classical because they have the intersection property (that is, the semantics of theories can always be obtained by considering the semantics of its members independently). We called characterization logics characterizing because their standard equivalence coincides with strong equivalence in the characterized formalism. As an application of our results, we obtained a first characterization logic for abstract argumentation where single AFs are interpreted as theories. This new logic complements the already existing Dung-logics which consider single AFs as building blocks and hence, theories as sets of AFs [36].

Most previous work on characterizing strong equivalence in KR that we know of focused on specific formalisms or on a handful of related formalisms, such as work on strong equivalence in logic programs under stable models [2, 3] and supported models [8], that also give rise to similar developments in default logic [3, 7] and autoepistemic logic [8]. By considering and exploiting formalism specifics, more fine-grained views on classical, strong and intermediate equivalence notions are possible [4, 7, 6]. Such notions are at present not “visible” in our setting, but could be incorporated by restricting the set of theories that are allowed for expansion.

We have chosen to consider as “classical” all logics whose model function possesses the intersection property. Other characterizations might be possible when choosing consequence functions instead of model functions as starting point. For those, we considered closure operators as a special class (implying, for example, cumulativity). Other properties for future consideration come to mind – for example compactness, which is independent of the closure property and relevant to proof theory.

Our approach could be further generalized by abstracting away even from knowledge bases as sets and knowledge base expansion as set union. We could assume a language as equipped with an expansion operator \oplus under which the language is closed and then derive our results completely algebraically. This would enable us to treat, for example, abstract dialectical frameworks [41, 42, 43], a quite recent non-classical KR formalism encompassing both argumentation frameworks and logic programs [17], for which strong equivalence has not been studied yet.

References

- [1] M. Gebser, R. Kaminski, B. Kaufmann, T. Schaub, Answer set solving in practice, Synthesis Lectures on Artificial Intelligence and Machine Learning 6 (3) (2012) 1–238. [doi:10.2200/S00457ED1V01Y201211AIM019](https://doi.org/10.2200/S00457ED1V01Y201211AIM019).
- [2] V. Lifschitz, D. Pearce, A. Valverde, Strongly equivalent logic programs, ACM Transactions on Computational Logic 2 (4) (2001) 526–541. [doi:10.1145/502166.502170](https://doi.org/10.1145/502166.502170).

- [3] H. Turner, Strong equivalence for logic programs and default theories (made easy), in: LPNMR, 2001, pp. 81–92.
- [4] T. Eiter, M. Fink, Uniform equivalence of logic programs under the stable model semantics, in: C. Palamidessi (Ed.), Logic Programming, 19th International Conference, ICLP 2003, Mumbai, India, December 9–13, 2003, Proceedings, Vol. 2916 of Lecture Notes in Computer Science, Springer, 2003, pp. 224–238. doi:10.1007/978-3-540-24599-5_16.
- [5] E. Oikarinen, T. Janhunen, Achieving compositionality of the stable model semantics for smodels programs, Theory Pract. Log. Program. 8 (5-6) (2008) 717–761. doi:10.1017/S147106840800358X.
- [6] S. Woltran, A common view on strong, uniform, and other notions of equivalence in answer-set programming, TPLP 8 (2) (2008) 217–234. doi:10.1017/S1471068407003250.
- [7] M. Truszczyński, The modal logic S4F, the default logic, and the logic here-and-there, in: Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22–26, 2007, Vancouver, British Columbia, Canada, AAAI Press, 2007, pp. 508–514. URL <http://www.aaai.org/Library/AAAI/2007/aaai07-080.php>
- [8] M. Truszczyński, S. Woltran, Hyperequivalence of logic programs with respect to supported models, in: D. Fox, C. P. Gomes (Eds.), Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13–17, 2008, AAAI Press, 2008, pp. 560–565. URL <http://www.aaai.org/Library/AAAI/2008/aaai08-089.php>
- [9] P. Cabalar, M. Diéguez, Strong equivalence of non-monotonic temporal theories, in: C. Baral, G. D. Giacomo, T. Eiter (Eds.), Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014, Vienna, Austria, July 20–24, 2014, AAAI Press, 2014, pp. 598–601. URL <http://www.aaai.org/ocs/index.php/KR/KR14/paper/view/7954>
- [10] J. A. Goguen, R. M. Burstall, Introducing institutions, in: Logics of Programs, Springer, 1984, pp. 221–256.
- [11] L. Straßburger, What is a logic, and what is a proof?, in: J.-Y. Beziau (Ed.), Logica Universalis, Birkhäuser, 2005, pp. 135–145, updated version at <http://www.lix.polytechnique.fr/~lutz/papers/WhatLogicProof.pdf>.
- [12] M. Truszczyński, Strong and uniform equivalence of nonmonotonic theories – an algebraic approach, Annals of Mathematics and Artificial Intelligence 48 (3–4) (2006) 245–265. doi:10.1007/s10472-007-9049-2.
- [13] M. Denecker, V. Marek, M. Truszczyński, Approximations, Stable Operators, Well-Founded Fixpoints and Applications in Nonmonotonic Reasoning, in: Logic-Based Artificial Intelligence, Kluwer Academic Publishers, 2000, pp. 127–144.
- [14] P. M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n -person games, Artificial Intelligence 77 (2) (1995) 321–357.
- [15] E. Oikarinen, S. Woltran, Characterizing strong equivalence for argumentation frameworks, Artificial Intelligence 175 (14–15) (2011) 1985–2009. doi:10.1016/j.artint.2011.06.003.

- [16] R. Baumann, [Characterizing equivalence notions for labelling-based semantics](#), in: Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa, April 25-29, 2016., 2016, pp. 22–32. URL <http://www.aaai.org/ocs/index.php/KR/KR16/paper/view/12836>
- [17] H. Strass, Approximating operators and semantics for abstract dialectical frameworks, *Artificial Intelligence* 205 (2013) 39–70.
- [18] M. Osorio, C. Zepeda, J. C. Nieves, U. Cortés, Inferring acceptable arguments with answer set programming, in: Proceedings of the Sixth Mexican International Conference on Computer Science (ENC), 2005, pp. 198–205.
- [19] Y. Wu, M. Caminada, D. M. Gabbay, Complete Extensions in Argumentation Coincide with 3-Valued Stable Models in Logic Programming, *Studia Logica* 93 (2–3) (2009) 383–403.
- [20] A. J. Harrison, V. Lifschitz, F. Yang, [The semantics of gringo and infinitary propositional formulas](#), in: C. Baral, G. D. Giacomo, T. Eiter (Eds.), Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014, Vienna, Austria, July 20-24, 2014, AAAI Press, 2014, pp. 32–41. URL <http://www.aaai.org/ocs/index.php/KR/KR14/paper/view/7966>
- [21] A. Harrison, V. Lifschitz, M. Truszczynski, On equivalence of infinitary formulas under the stable model semantics, *Theory Pract. Log. Program.* 15 (1) (2015) 18–34. doi:10.1017/S1471068414000088.
- [22] A. Harrison, V. Lifschitz, D. Pearce, A. Valverde, Infinitary equilibrium logic and strongly equivalent logic programs, *Artif. Intell.* 246 (2017) 22–33. doi:10.1016/j.artint.2017.02.002.
- [23] M. Gelfond, V. Lifschitz, The stable model semantics for logic programming, in: ICLP, 1988, pp. 1070–1080.
- [24] K. L. Clark, Negation as failure, in: H. Gallaire, J. Minker (Eds.), *Logic and Data Bases*, Plenum Press, 1978, pp. 293–322.
- [25] M. H. van Emden, R. A. Kowalski, The Semantics of Predicate Logic as a Programming Language, *Journal of the ACM* 23 (4) (1976) 733–742.
- [26] R. Reiter, A Logic for Default Reasoning, *Artificial Intelligence* 13 (1980) 81–132.
- [27] H. Turner, Strong equivalence for logic programs and default theories (made easy), in: T. Eiter, W. Faber, M. Truszczyński (Eds.), *Logic Programming and Nonmonotonic Reasoning*, 6th International Conference, LPNMR 2001, Vienna, Austria, September 17-19, 2001, Proceedings, Vol. 2173 of Lecture Notes in Computer Science, Springer, 2001, pp. 81–92. doi:10.1007/3-540-45402-0_6.
- [28] H. Turner, Strong equivalence for causal theories, in: *Logic Programming and Nonmonotonic Reasoning*, 7th International Conference, LPNMR 2004, Fort Lauderdale, FL, USA, January 6-8, 2004, Proceedings, 2004, pp. 289–301. doi:10.1007/978-3-540-24609-1_25.
- [29] O. Ore, [Galois connexions](#), *Transactions of the American Mathematical Society* 55 (3) (1944) 493–513. URL <http://www.jstor.org/stable/1990305>

- [30] G. Birkhoff, *Lattice Theory*, 3rd Edition, Vol. 25 of American Mathematical Society Colloquium Publications, American Mathematical Society, Providence, Rhode Island, 1973.
- [31] P. M. Cohn, *Universal algebra*, Mathematics and Its Applications, Springer, Dordrecht, 1981.
- [32] B. Davey, H. Priestley, *Introduction to Lattices and Order*, 2nd Edition, Cambridge University Press, 2002.
- [33] D. M. Gabbay, Theoretical foundations for non-monotonic reasoning in expert systems, in: K. R. Apt (Ed.), *Logics and Models of Concurrent Systems*, Springer Berlin Heidelberg, Berlin, Heidelberg, 1985, pp. 439–457.
- [34] S. Kraus, D. Lehmann, M. Magidor, Nonmonotonic reasoning, preferential models and cumulative logics, *Artificial Intelligence* 44 (1) (1990) 167 – 207. doi:[https://doi.org/10.1016/0004-3702\(90\)90101-5](https://doi.org/10.1016/0004-3702(90)90101-5).
- [35] D. Makinson, [General patterns in nonmonotonic reasoning](#), in: D. M. Gabbay, C. J. Hogger, J. A. Robinson (Eds.), *Handbook of Logic in Artificial Intelligence and Logic Programming* (Vol. 3), Oxford University Press, Inc., New York, NY, USA, 1994, pp. 35–110. URL <http://dl.acm.org/citation.cfm?id=186124.186126>
- [36] R. Baumann, G. Brewka, [AGM meets abstract argumentation: Expansion and revision for Dung frameworks](#), in: Q. Yang, M. Wooldridge (Eds.), *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015*, Buenos Aires, Argentina, July 25-31, 2015, AAAI Press, 2015, pp. 2734–2740. URL <http://ijcai.org/papers15/Abstracts/IJCAI15-387.html>
- [37] P. Baroni, M. Caminada, M. Giacomin, Abstract argumentation frameworks and their semantics, in: P. Baroni, D. Gabbay, M. Giacomin, L. van der Torre (Eds.), *Handbook of Formal Argumentation*, College Publications, 2018, Ch. 4, pp. 159–236.
- [38] R. Baumann, C. Spanring, Infinite argumentation frameworks – on the existence and uniqueness of extensions, in: T. Eiter, H. Strass, M. Truszczynski, S. Woltran (Eds.), *Advances in Knowledge Representation, Logic Programming, and Abstract Argumentation – Essays Dedicated to Gerhard Brewka on the Occasion of His 60th Birthday*, Vol. 9060 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 281–295. doi:[10.1007/978-3-319-14726-0_19](https://doi.org/10.1007/978-3-319-14726-0_19).
- [39] R. Baumann, C. Spanring, A study of unrestricted abstract argumentation frameworks, in: *IJCAI, Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2017, pp. 807–813.
- [40] R. Baumann, On the nature of argumentation semantics: Existence and uniqueness, expressibility, and replaceability, in: P. Baroni, D. Gabbay, M. Giacomin, L. van der Torre (Eds.), *Handbook of Formal Argumentation*, College Publications, 2018, Ch. 18, pp. 839–936.
- [41] G. Brewka, S. Woltran, [Abstract dialectical frameworks](#), in: *Proceedings of the Twelfth International Conference on the Principles of Knowledge Representation and Reasoning (KR)*, 2010, pp. 102–111. URL <http://aaai.org/ocs/index.php/KR/KR2010/paper/view/1294>
- [42] G. Brewka, S. Ellmauthaler, H. Strass, J. P. Wallner, S. Woltran, Abstract dialectical frameworks revisited, in: *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI), IJCAI/AAAI*, 2013, pp. 803–809.

- [43] G. Brewka, S. Ellmauthaler, H. Strass, J. P. Wallner, S. Woltran, Abstract dialectical frameworks, in: P. Baroni, D. Gabbay, M. Giacomin, L. van der Torre (Eds.), *Handbook of Formal Argumentation*, College Publications, 2018, Ch. 5, pp. 237–286.