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

Ringo Baumann
baumann@informatik.uni-leipzig.de

Hannes Strass
strass@informatik.uni-leipzig.de

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

Abstract

Two knowledge bases are strongly equivalent if and only if they are mutually inter-
changeable in arbitrary contexts. This notion is of high interest for any logical formalism
since it allows one to locally replace, and thus give rise for simplification, parts of a given
theory without changing the semantics of the latter. In contrast to classical logic where
strong equivalence coincides with standard equivalence (having the same models), it is pos-
sible 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, 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 equiv-
alent in the logic 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).

1. Introduction

Reusability of human-made artifacts is of paramount importance in computer science. To
assess the reusability of (parts of) knowledge bases in logic-based knowledge represen-
tation languages, 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 hav-
ing 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 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 (Gebser, Kaminski, Kaufmann, & Schaub, 2012). Alas,
for normal logic programs, having the same stable models does not amount to mutual re-
placeability. For this, a stronger property is needed: it is called \textit{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 \texttildelow, 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.

In a series of interesting developments, researchers have succeeded in precisely characterizing strong equivalence for several formalisms, among them normal logic programs under the stable model semantics (Lifschitz, Pearce, & Valverde, 2001; Turner, 2001a). What is more, it turned out that sometimes, to characterize strong equivalence in formalism $F$, we can use ordinary equivalence in formalism $F'$: 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 (Lifschitz et al., 2001). 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: \textit{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 \textit{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 (Lifschitz et al., 2001; Turner, 2001a; Truszczynski, 2007; Truszczynski & Woltran, 2008; Cabalar & Diéguez, 2014). 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 \textit{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. 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, Truszczynski (2006) presented a general, algebraic account of strong equivalence within approximation fixpoint theory (Denecker, Marek, & Truszczynski, 2000). 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, 1995), another important AI formalism whose strong equivalence has been studied in the recent past (Oikarinen & Woltran, 2011; Baumann, 2016). More precisely, while AFs with all their semantics can be captured by approximation fixpoint theory (Strass, 2013), Truszczynski’s notion of expanding an operator does not coincide with the corresponding notion of expanding AFs and his results are not
directly applicable. 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 \( P_1 = \{ a \leftarrow \neg b \} \); likewise, \( P_2 = \{ a \leftarrow \neg c \} \) corresponds to the AF where  \( c \) attacks  \( a \) (Osorio, Zepeda, Nieves, & Cortés, 2005; Wu, Caminada, & Gabbay, 2009; Strass, 2013). However, the AF where both  \( b \) and  \( c \) attack  \( a \) (the union of the two AFs above) corresponds to the logic program \( P_3 = \{ a \leftarrow \neg b, \neg c \} \), where obviously the three programs are not subset-related: \( P_1 \nsubseteq P_3 \) and \( P_2 \nsubseteq P_3 \).

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 (Oikarinen & Woltran, 2011).

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 \( L \), that is, non-empty sets of language elements. We make no assumption on the internal structure of pieces of knowledge \( f \in L \). 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 \( L \) uses a set \( I \) of interpretations and a model function \( \sigma : 2^L \to 2^I \) with the intuition that \( \sigma \) assigns each language subset \( T \subseteq L \), a theory, the set \( \sigma(T) \) 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,

---

1. As 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: \( (A_1, R_1) \cup (A_2, R_2) = (A_1 \cup A_2, R_1 \cup R_2) \).
but only to sets thereof. We illustrate our definitions so far by showing more precisely how existing formalisms can be embedded into our setting.

**Example 1.** Consider a set $A$ of propositional atoms.

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

Normal logic programs: The underlying language $L_{LP}$ is the set of all normal logic program rules $a_0 \leftarrow a_1, \ldots, a_m, \sim a_{m+1}, \ldots, \sim a_n$ with $0 \leq m \leq n$ and $a_0, a_1, \ldots, a_n \in A$. The set $I_{LP}$ of interpretations is then the set $I_{LP} = 2^A$ of all possible stable model candidates. Accordingly, $\sigma_{stb}(T)$ returns the set of stable models of the theory (normal logic program) $T \subseteq L_{LP}$ (Gelfond & Lifschitz, 1988); we could also define $\sigma_{sup}(T)$ returning the set of supported models of $T$ (Clark, 1978), or $\sigma_{mod}(T)$ the set of classical models of $T$, interpreting $\leftarrow$ as material implication and $\sim$ as classical negation (van Emde & Kowalski, 1976).

Propositional default logic (Reiter, 1980): The underlying language $L_{DL}$ is the set of all defaults over formulas over $A$, that is, $L_{DL} = \{(\varphi, \Psi, \xi) \mid \varphi, \xi \in L_{PL}, \Psi \subseteq L_{PL}\}$ where the triple $(\varphi, \{\psi_1, \ldots, \psi_n\}, \xi)$ represents the default $\varphi : \psi_1, \ldots, \psi_n / \xi$. The possible interpretations are given by $I_{DL} = 2^L$, as each default theory is assigned a set of logical theories called extensions: $\sigma_{DL}(T) = \{E \subseteq L \mid E$ is an extension of $T\}$.

Abstract argumentation frameworks (Dung, 1995): The underlying language $L_{AF}$ contains the fundamental building blocks of AFs, that is, arguments and attacks:

$$L_{AF} = \{\{(a, \emptyset), (\{a, b\}, \{a, b\}\) \mid a, b \in A\}$$

Extension-based semantics can be incorporated by setting $I_{AF} = 2^A$ and, depending on the argumentation semantics $\rho$ 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 $T \subseteq L_{AF}$ (cf. Section 4.1 for more detailed information and discussion).

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 $L$ is a function $Cn : 2^L \rightarrow 2^L$ 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 $L$ be a language and $\sigma : 2^L \to 2^I$ be a model function. The *canonical consequence function* of $\sigma$ is defined as follows:

$$Cn^\sigma : 2^L \to 2^L, \quad T \mapsto \bigcup_{S \subseteq L, \sigma(T) \subseteq \sigma(S)} S$$

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 $L_{PL}$, 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.

**Definition 2.** Let $L$ be a language.

- A model function $\sigma : 2^L \to 2^I$ is *antimonotone* iff for all $T_1, T_2 \in 2^L$: $T_1 \subseteq T_2 \implies \sigma(T_2) \subseteq \sigma(T_1)$.
- A consequence operator $Cn : 2^L \to 2^L$ is *monotone* iff for all $T_1, T_2 \in 2^L$: $T_1 \subseteq T_2 \implies Cn(T_1) \subseteq Cn(T_2)$.
- A consequence operator $Cn : 2^L \to 2^L$ is *increasing* iff for all $T \in 2^L$, we find $T \subseteq Cn(T)$.
- A consequence operator $Cn : 2^L \to 2^L$ is *idempotent* iff for all $T \in 2^L$, we find $Cn(Cn(T)) \subseteq Cn(T)$.
- A consequence operator $Cn : 2^L \to 2^L$ 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 $(L, I, \sigma)$ consisting of a language $L$, an interpretation set $I$, and a model function $\sigma : 2^L \to 2^I$. We will sometimes associate the canonical consequence function of $\sigma$ 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 antimonotone if the considered model function is antimonotone.

---

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

1. \(Cn^\sigma\) is increasing.
2. If \(\sigma\) is antimonotone, then \(Cn^\sigma\) is monotone.

**Proof.**

1. Clearly for any \(T \subseteq \mathcal{L}\) we find \(\sigma(T) \subseteq \sigma(T)\) whence \(T \subseteq \bigcup_{S \subseteq \mathcal{L}, \sigma(T) \subseteq \sigma(S)} S = Cn^\sigma(T)\).

2. Let \(\sigma\) 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, \(Cn^\sigma\) is monotone. \(\square\)

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 \(I = \{1\}\) with \(\sigma(\emptyset) = \emptyset\) and \(\sigma(\{a\}) = \{1\}\). We get the following canonical consequence function:

\[
Cn^\sigma(\emptyset) = \bigcup_{S \subseteq \mathcal{L}, \sigma(\emptyset) \subseteq \sigma(S)} S = \bigcup_{S \subseteq \mathcal{L}, 0 \subseteq \sigma(S)} S = \bigcup_{S \subseteq \mathcal{L}} S = \mathcal{L} = \{a\}
\]

\[
Cn^\sigma(\{a\}) = \bigcup_{S \subseteq \mathcal{L}, \sigma(\{a\}) \subseteq \sigma(S)} S = \bigcup_{S \subseteq \mathcal{L}, \{1\} \subseteq \sigma(S)} S = \{a\}
\]

Thus \(Cn^\sigma(\emptyset) = \{a\} = Cn^\sigma(\{a\})\) and \(Cn^\sigma\) is monotone and increasing albeit \(\sigma\) 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 \((\mathcal{L}, I, \sigma)\) with language \(\mathcal{L} = \{a, b, c\}\), interpretation set \(I = \{1\}\) and model function \(\sigma : 2^\mathcal{L} \to 2^I\) given by

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

In addition to \(\sigma\) being antimonotone we have \(Cn^\sigma(\emptyset) = \{a, b\}\) and \(Cn^\sigma(\{a, b\}) = \{a, b, c\}\), whence

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

which shows that \(Cn^\sigma\) 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 $C : 2^\mathcal{L} \to 2^\mathcal{L}$ be a consequence function which is increasing and idempotent.

Then the model function $\sigma_C : 2^\mathcal{L} \to 2^\mathcal{L}$ with $T \mapsto \mathcal{L} \setminus C(T)$ is such that $C^{\sigma_C} = C$.\(^4\)

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

\[
C^{\sigma_C}(T) = \bigcup_{S \subseteq \mathcal{L}, \sigma_C(T) \subseteq \sigma_C(S)} S \quad \text{(Def. } C^{\sigma})
\]

\[
= \bigcup_{S \subseteq \mathcal{L}, \mathcal{L} \setminus C(T) \subseteq \mathcal{L} \setminus C(S)} S \quad \text{(Def. } \sigma_C)
\]

\[
= \bigcup_{S \subseteq \mathcal{L}, C(S) \subseteq C(T)} S \quad \text{(set algebra)}
\]

\[
= \bigcup_{S \subseteq \mathcal{L}, S \subseteq C(S) \subseteq C(T)} S \quad \text{(} C \text{ is increasing)}
\]

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

It is clear from Proposition 1 (Item 1) that no non-increasing consequence function $C$ can be mimicked by $C^{\sigma}$ for any $\sigma$. 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 $\forall U \subseteq \mathcal{L} : \sigma(T_1 \cup U) = \sigma(T_2 \cup U)$. \(\diamondsuit\)

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.

4. Please note that the assigned interpretation set $\mathcal{I}$ of $\sigma_C$ is just the language $\mathcal{L}$. 

7
Example 4. Given a logic \((\mathcal{L}, \mathcal{I}, \sigma)\), a theory \(S\) and a subtheory \(T_1\) of it, i.e. \(T_1 \subseteq S\). Now, we may replace \(T_1\) with any \(T_2\) being strongly equivalent to it without changing the semantics of \(S\). 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 assumed strong equivalence of \(T_1\) and \(T_2\) we obtain \(\sigma(T_1 \cup (S \setminus T_1)) = \sigma(T_1 \cup (S \setminus T_1))\). 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 (Lifschitz et al., 2001), default logic (Turner, 2001b), causal theories (Turner, 2004) and abstract argumentation (?; Baumann, 2016) 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 \(\sigma\) 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 \(\sigma\) given by

\[
\begin{align*}
\sigma(\emptyset) &= \{1, 2\} \\
\sigma(\{a\}) &= \{1, 2\} \\
\sigma(\{b\}) &= \{2\} \\
\sigma(\{a, b\}) &= \emptyset
\end{align*}
\]

It is easy to verify that the semantics \(\sigma\) is antimonotone. Therefore, by Proposition 1, its consequence function \(Cn^\sigma\) 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:

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

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

\[\diamondsuit\]
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 \(T \subseteq \mathcal{L}\):

\[
\sigma(T) = \bigcap_{F \in T} \sigma(\{F\}) \tag{\text{intersection}}
\]

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.\(^5\) Towards an explanation of Example 5 we can now remark that its model function \(\sigma\) 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 \(\sigma\) satisfies the intersection property, then standard equivalence coincides with strong equivalence.

**Proof.** Let \(\sigma\) 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

\[
\begin{align*}
\sigma(T_1 \cup U) &= \sigma(T_1) \cap \sigma(U) \quad \text{(intersection)} \\
&= \sigma(T_2) \cap \sigma(U) \quad \text{(presumption)} \\
&= \sigma(T_2 \cup U) \quad \text{(intersection)} \\
\end{align*}
\]

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 \(\sigma\) having the intersection property implies that \(\sigma\) is antimonotone (which in turn implies that \(\text{Cn}^\sigma\) is monotone).

**Proposition 4.** Let \((\mathcal{L}, \mathcal{I}, \sigma)\) be a logic where \(\sigma\) has the intersection property. Then \(\sigma\) 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 \(\sigma(T_2) = \sigma(T_1 \cup T_2) = \sigma(T_1) \cap \sigma(T_2) \subseteq \sigma(T_1)\).

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

\(^5\) We 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.
Example 6. Consider the language $L = \{ a \}$ and interpretation set $I = \{ 1 \}$. For semantics $\sigma$ 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, $\sigma$ is not antimonotone (as $\sigma(\{ a \}) = \{ 1 \} \not\subseteq \emptyset = \sigma(\emptyset)$) and does not have the intersection property:

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

It is easy to see that classical propositional logic $L_{PL}$ has the intersection property simply by definition: the standard model semantics is typically firstly defined for single formulas $\varphi \in L_{PL}$ 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 $(L, I, \sigma)$ be a logic that has the intersection property. Then for each $T \subseteq L$ we find that $\sigma(T) = \sigma(Cn^\sigma(T))$.

Proof.

$$
\sigma(Cn^\sigma(T)) = \sigma\left( \bigcup_{S \subseteq L, \sigma(T) \subseteq \sigma(S)} S \right) \quad \text{(Definition 1)}
$$

$$
= \bigcap_{S \subseteq L, \sigma(T) \subseteq \sigma(S)} \sigma(S) \quad \text{(intersection property)}
$$

$$
= \sigma(T)
$$

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.

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

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

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

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 $C : 2^L \to 2^L$ in the sense that $C$ has the replacement property if and only if for all theories $T_1, T_2 \subseteq L$, “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 \( \sigma \) and \( \tau \), we can define \( C_{\sigma, \tau}(T) = \tau(\sigma(T)) \). 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 \( \sigma \) and \( \tau \) are in Galois correspondence if and only if:

1. \( \sigma \) is antimonotone and \( \tau \) is antimonotone;
2. \( \sigma \circ \tau \) is increasing and \( \tau \circ \sigma \) is increasing. \( \Box \)

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

The first two properties together imply that the resulting consequence function is monotone; by the last property, it is also increasing. Galois correspondences have been studied in model theory (Goguen & Burstall, 1984), and in an even more abstract way in lattice theory (Ore, 1944; Birkhoff, 1973; Cohn, 1981; Davey & Priestley, 2002).

Indeed, there is also another (equivalent) formulation (Davey & Priestley, 2002).

**Proposition 7.** Let \((\mathcal{L}, \mathcal{I}, \sigma, \tau)\) be a logic. It holds that \( \sigma \) and \( \tau \) 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 $\sigma$ 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 $\tau$ is antimonotone.

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

It also follows that Galois correspondences induce closure operators, that is, operators that are monotone, increasing, and idempotent.

**Proposition 8.** Let $(\mathcal{L}, \mathcal{I}, \sigma, \tau)$ be a Galois logic. Then the operators

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

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 $\sigma$ and $\tau$ are antimonotone: if $K_1 \subseteq K_2$, then $\tau(K_2) \subseteq \tau(K_1)$ whence $\sigma(\tau(K_2)) \subseteq \sigma(\tau(K_1))$, 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 $\sigma$ is antimonotone, $\sigma(\tau(\sigma(\tau(K)))) \subseteq \sigma(\tau(K))$. \hfill \Box

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

$$\sigma\left(\bigcup_{T \in \mathcal{T}} T\right) = \bigcap_{T \in \mathcal{T}} \sigma(T)$$
Proof. if: Let \( T \subseteq \mathcal{L} \) and define \( \mathcal{T} = \{ \{ F \} \mid F \in T \} \). Clearly \( T = \bigcup_{F \in T} \{ F \} \), whence

\[
\sigma(T) = \sigma\left( \bigcup_{F \in T} \{ F \} \right) \\
= \sigma\left( \bigcup_{\{ F \} \in \mathcal{T}} \{ F \} \right) \\
= \bigcap_{\{ F \} \in \mathcal{T}} \sigma(\{ F \}) \\
= \bigcap_{F \in T} \sigma(\{ F \})
\]

only if: Let \( \mathcal{T} \subseteq 2^\mathcal{L} \) and define \( T = \bigcup_{U \in \mathcal{T}} U \). Now we have that

\[
\sigma\left( \bigcup_{U \in \mathcal{T}} U \right) = \sigma(T) \\
= \bigcap_{F \in T} \sigma(\{ F \}) \\
= \bigcap_{F \in \bigcup_{U \in \mathcal{T}} U} \sigma(\{ F \}) \\
= \bigcap_{U \in \mathcal{T}} \left( \bigcap_{F \in U} \sigma(\{ F \}) \right) \\
= \bigcap_{U \in \mathcal{T}} \sigma(U)
\]

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

We now conclude this section with its main result. It states that for any logic \((\mathcal{L}, \mathcal{I}, \sigma)\), the conditions “\( \sigma \) has the intersection property” and “\( \sigma \) is in a Galois correspondence with some \( \tau \)” are equivalent. The proof can be adapted from the literature (Davey & Priestley, 2002, 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 \( \tau : 2^\mathcal{I} \to 2^\mathcal{L} \) such that \( \sigma \) and \( \tau \) are in Galois correspondence, then \( \sigma \) has the intersection property.

2. If \( \sigma \) has the intersection property, then we can define a theory function \( \tau : 2^\mathcal{I} \to 2^\mathcal{L} \) with

\[
K \mapsto \bigcup_{T \subseteq \mathcal{L}, \ K \subseteq \sigma(T)} T
\]

such that \( \sigma \) and \( \tau \) are in Galois correspondence.
Proof. 1. Assume the presumption. We have to show that for any $T \subseteq 2^L$, we find that

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

Now denote $Z = \bigcup_{T \in \mathcal{T}} T$. We first show that $\sigma(Z)$ is a lower bound for the set $\sigma(T) = \{\sigma(T) \mid T \in \mathcal{T}\}$. Clearly, for each $T \in \mathcal{T}$ we have $T \subseteq Z$, whence by anti-monotonicity of $\sigma$ we get $\sigma(Z) \subseteq \sigma(T)$ for each $T \in \mathcal{T}$. Now let $Q \subseteq \mathcal{I}$ be any lower bound for $\sigma(T)$. Then $Q \subseteq \sigma(T)$ for all $T \in \mathcal{T}$. By Proposition 7, 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 7 again yields $Q \subseteq \sigma(Z)$ whence $\sigma(Z)$ is the greatest lower bound of $\sigma(T)$.

2. Let $\sigma$ have the intersection property. By Proposition 4, it follows that $\sigma$ is antimono-
tone. For antimonotonicity of $\tau$, consider $K_1 \subseteq K_2$. Then

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

Now for showing that $\sigma \circ \tau$ and $\tau \circ \sigma$ are increasing.

Let $K \subseteq \mathcal{I}$. We find

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

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

$$T \subseteq \bigcup_{S \subseteq L, \sigma(T) \subseteq \sigma(S)} S = \tau(\sigma(T))$$

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. 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 $(\mathcal{L}, \sigma)$, since concrete interpretations are immaterial for strong equivalence. Furthermore, we will distinguish between two important cases regarding the domain of $\sigma$. The first one is $dom(\sigma) = 2^L$ (called full logics) and the second one is $dom(\sigma) = (2^L)_{mf} = \{T \in 2^L \mid T \text{ is finite}\}$ (finite-theory logics), the restriction of $\mathcal{L}$ to finite knowledge bases.
Definition 6. Let \((\mathcal{L}, \sigma)\) be a logic. We define the binary relation \(\text{strong equivalence} \equiv_s \subseteq \text{dom}(\sigma) \times \text{dom}(\sigma)\) by \(T_1 \equiv_s T_2 \iff \forall U \in \text{dom}(\sigma) : \sigma(T_1 \cup U) = \sigma(T_2 \cup U)\). 

It is straightforward to show that \(\equiv_s\) is an equivalence relation; we denote the equivalence class of a theory \(T \in \text{dom}(\sigma) \subseteq 2^\mathcal{L}\) by \([T]_s\). We recall and will pervasively use that for all theories \(T_1, T_2 \subseteq \mathcal{L}\), we have \(T_1 \in [T_2]_s^s\) if and only if \([T_1]_s^s = [T_2]_s^s\). Given an arbitrary logic \((\mathcal{L}, \sigma)\), we want to find a characterizing classical logic, that is, a semantics \(\sigma'\) that has the intersection property and whose ordinary equivalence coincides with strong \(\sigma\)-equivalence. Such logics get a name.

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

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

(characterization)

\[
\forall T \subseteq 2^\mathcal{L} : \sigma'\left( \bigcup_{T \in \mathcal{T}} T \right) = \bigcap_{T \in \mathcal{T}} \sigma'(T)
\]

(intersection)

Since characterization logics are the centerpiece of our study we would like to take a look to this central definition from an other angle, namely in terms of consequence functions. Note that the analysis of consequence relations have been very prominent in the formative years of nonmonotonic reasoning. We refer the interested reader to (Gabbay, 1985; Kraus, Lehmann, & Magidor, 1990; Makinson, 1994). Let us consider the canonical consequence functions of \(\sigma\) and \(\sigma'\) according to Definition 1. Doing so reveals that the characterizing and characterized consequence relations fit together appropriately as stated in the following assertion. It is part of future work to study further properties in terms of consequence relations.

Proposition 11. Given a logic \((\mathcal{L}, \sigma)\) and a characterization logic \((\mathcal{L}, \sigma')\) of it. Let \(\text{Cn}^\sigma\) and \(\text{Cn}^{\sigma'}\) be the canonical consequence functions of the corresponding logics. Given \(T \subseteq \mathcal{L}\).

1. \(\text{Cn}^{\sigma'}(T) \subseteq \text{Cn}^\sigma(T)\) (sublogic)

2. \(\text{Cn}^\sigma(T) = \text{Cn}^{\sigma'}(\text{Cn}^{\sigma'}(T))\) (left absorption)

3. \(\text{Cn}^\sigma(T) = \text{Cn}^\sigma(\text{Cn}^{\sigma'}(T))\) (right absorption)

Proof. In the following proofs we will often use that \((\mathcal{L}, \sigma)\) and \((\mathcal{L}, \sigma')\) are characterization logics implying that \(\sigma\) and \(\sigma'\) possess the intersection property (Definition 7) which in turn means that \(\text{Cn}^\sigma\) and \(\text{Cn}^{\sigma'}\) are closure operators (Proposition 6).

1. In order to show \(\text{Cn}^{\sigma'}(T) \subseteq \text{Cn}^\sigma(T)\) we first prove the subsequent property (*). For any \(U \subseteq \mathcal{L}\): If \(U \subseteq \text{Cn}^{\sigma'}(T)\), then \(\text{Cn}^{\sigma'}(T) = \text{Cn}^{\sigma'}(T \cup U)\). Since \(T \subseteq T \cup U\) we derive \(\text{Cn}^{\sigma'}(T) \subseteq \text{Cn}^{\sigma'}(T \cup U)\) (monotonicity). Moreover, we have \(T \subseteq \text{Cn}^{\sigma'}(T)\) (increasing) and \(U \subseteq \text{Cn}^{\sigma'}(T)\) (assumption) justifying \(T \cup U \subseteq \text{Cn}^{\sigma'}(T)\). Hence, \(\text{Cn}^{\sigma'}(T \cup U) \subseteq \text{Cn}^{\sigma'}(\text{Cn}^{\sigma'}(T))\) (monotonicity) and finally, \(\text{Cn}^\sigma(T \cup U) \subseteq \text{Cn}^{\sigma'}(T)\) (idempotency) concluding the proof for property (*).
Theorem 12. Let \( (\mathcal{L}, \sigma) \) be a full logic with characterization logic \( (\mathcal{L}, \sigma') \). The pair \( (\sigma'(2^\mathcal{L}), \subseteq) \) is a complete lattice where \( \text{glb} \wedge \) and \( \text{lub} \vee \) are given such that for all \( \mathcal{K} \subseteq \sigma'(2^\mathcal{L}) \),

\[
\bigwedge_{K \in \mathcal{K}} K = \bigcap_{K \in \mathcal{K}} K \quad \text{and} \quad \bigvee_{K \in \mathcal{K}} K = \bigvee_{L \in \mathcal{K}^u} L
\]

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 \( T \subseteq 2^\mathcal{L} \) that contains a \( T \subseteq \mathcal{T} \) with \( \sigma'(T) = K \) for each \( K \in \mathcal{K} \). Since \( \bigcup_{T \in \mathcal{T}} T \in 2^\mathcal{L} \) and \( \sigma' \) has the intersection property, \( \bigcap_{K \in \mathcal{K}} K = \bigcap_{T \in \mathcal{T}} \sigma'(T) = \sigma'\left( \bigcup_{T \in \mathcal{T}} T \right) \in \sigma'(2^\mathcal{L}) \).

Now consider \( \bigvee_{K \in \mathcal{K}} L \). We show that \( \bigwedge_{L \in \mathcal{K}^u} L \) is the least upper bound 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, \( \forall K \in \mathcal{K} : K \subseteq \bigcap_{L \in \mathcal{K}^u} L \) whence \( \bigcap_{L \in \mathcal{K}^u} L \in \mathcal{K}^u \). In particular, if \( M \in \mathcal{K}^u \) then \( \bigcap_{L \in \mathcal{K}^u} L \subseteq M \) and \( \bigcap_{L \in \mathcal{K}^u} L \) is the least upper bound of \( \mathcal{K} \).

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

---

6. The proof of Theorem 12 reveals that \( (\mathcal{L}, \sigma') \) does not necessarily has to be a characterization logic of \( (\mathcal{L}, \sigma) \). Indeed, the stated properties regarding the model theory hold for any logic possessing the intersection property.
Theorem 7. Consider \( \mathcal{L} = \{a, b, c\} \) and semantics \( \sigma \) with
\[
\begin{align*}
[\emptyset]^\sigma_s &= \{\emptyset\}, \\
[\{a\}]^\sigma_s &= \{\{a\}\}, \\
[\{b\}]^\sigma_s &= \{\{b\}\}, \\
[\{c\}]^\sigma_s &= \{\{c\}\}, \\
[\{a, b\}]^\sigma_s &= \{\{a, c\}\} = \{\{a, b, c\}\} = \{\{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}
\end{align*}
\]
Assume that \( (\mathcal{L}, \sigma') \) is a characterization logic for \( (\mathcal{L}, \sigma) \). We claim that\( \sigma'([b]) \cup \sigma'([c]) \notin \sigma'(2^\mathcal{L}) \)
whence \( \sigma'(2^\mathcal{L}) \) 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
\[
\begin{align*}
\sigma'(\{a\} \cup T) &= \sigma'([a]) \cap \sigma'(T) \\
&= \sigma'([a]) \cap (\sigma'([b]) \cup \sigma'([c])) \\
&= (\sigma'([a]) \cap \sigma'([b])) \cup (\sigma'([a]) \cap \sigma'([c])) \\
&= \sigma'([a, b]) \cup \sigma'([a, c]) \\
&= \sigma'([a, b]) = \sigma'([a, c]) = \sigma'([b, c]) = \sigma'([a, b, c])
\end{align*}
\]
Thus \( T = \{b\} \) or \( T = \{c\} \) or \( T = [\{a, b, c\}]^\sigma_s \). Since \( \sigma' \) has the intersection property, it is in particular antimonotone (Proposition 4). Hence for \( T \in [\{a, b, c\}]^\sigma_s \), we get \( \sigma'(T) \subseteq \sigma'([b]) \) and since \([\{b\}]^\sigma_s \neq [\{a, b, c\}]^\sigma_s \) even \( \sigma'(T) \subseteq \sigma'([b]) \). Substituting \( c \) for \( b \) in the argument above, we get \( \sigma'(T) \subseteq \sigma'([c]) \) in the same fashion. In combination \( \sigma'(T) \subseteq \sigma'([b]) \cup \sigma'([c]) \), contradiction. Thus \( T = \{b\} \) or \( T = \{c\} \). Clearly \([\{b\}]^\sigma_s \neq [\{b, c\}]^\sigma_s \) and \([\{c\}]^\sigma_s \neq [\{b, c\}]^\sigma_s \) imply that
\[
\begin{align*}
\sigma'([b]) \cap \sigma'([c]) &= \sigma'([b]) \\
\sigma'([b]) \cap \sigma'([c]) &= \sigma'([b]) \subseteq \sigma'([b])
\end{align*}
\]
Thus \( \sigma'([b]) \) \( \sigma'([c]) \) \( \neq \emptyset \) and \( \sigma'([c]) \) \( \neq \emptyset \). Therefore
\[
\begin{align*}
\sigma'([b]) \cup \sigma'([c]) &= \sigma'(T)
\end{align*}
\]
as well as
\[
\begin{align*}
\sigma'([c]) \cup \sigma'([c]) &= \sigma'(T).
\end{align*}
\]
Contradiction. Thus \( T \) does not exist and \( \sigma'([b]) \cup \sigma'([c]) \notin \sigma'(2^\mathcal{L}). \)

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

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

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

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, \(\phi(\sigma'(T)) = \phi(\sigma''(T))\).

- \(\phi\) 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 \(\sigma''\) has the characterization property, we get \([T_1]^\phi = [T_2]^\phi\). Since \(\sigma'\) has the characterization property, we get \(K_1 = \sigma'(T_1) = \sigma'(T_2) = K_2\).

- \(\phi\) 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 \(\phi(\sigma'(T)) = \sigma''(T) = M\).

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

\[
\phi \left( \bigwedge_{K \in \mathcal{K}} K \right) = \phi \left( \bigcap_{K \in \mathcal{K}} K \right) = \phi \left( \bigcap_{T \in T} \sigma'(T) \right) = \phi \left( \sigma' \left( \bigcup_{T \in T} T \right) \right)
\]

\[
= \sigma'' \left( \bigcup_{T \in T} T \right) = \bigcap_{T \in T} \sigma''(T) = \bigcap_{K \in \mathcal{K}} \phi(K) = \bigvee_{K \in \mathcal{K}} \phi(K)
\]

Since \(\bigvee\) can be defined in terms of \(\bigwedge\), it follows that

\[
\phi \left( \bigvee_{K \in \mathcal{K}} K \right) = \phi \left( \bigwedge_{K \in \mathcal{K}^\omega} K \right) = \bigwedge_{K \in \mathcal{K}^\omega} \phi(K) = \bigvee_{K \in \mathcal{K}} \phi(K)
\]

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 \(I \neq \emptyset\) arbitrary. We define the semantics \(\sigma : 2^\mathcal{L} \to 2^I\) such that

\[
\sigma(T) = \begin{cases} 
\emptyset & \text{if } T \text{ is finite}, \\
I & \text{otherwise}. 
\end{cases}
\]
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 \((\mathcal{L}, \sigma')\) is a characterization logic for \((\mathcal{L}, \sigma)\).

By the model intersection property, we get

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

in contradiction to the characterization property. \hfill \qed

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 \((\mathcal{L}, \sigma)\) 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:

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

**Proof.**

1. \[
[T_1]_s^\sigma = [T_2]_s^\sigma
\]

   \[
   \iff \forall U \subseteq \mathcal{L} : \sigma(T_1 \cup U) = \sigma(T_2 \cup U) \quad \text{(Def. } \equiv_s^\sigma)\]

   \[
   \iff \forall U' \subseteq \mathcal{L} : \sigma(T_1 \cup U) = \sigma(T_2 \cup U' \cup U'') \quad \text{(write } U' \cup U'')\]

   \[
   \iff \forall U' \subseteq \mathcal{L} : \sigma(T_1 \cup U') = \sigma(T_2 \cup U') \quad \text{(associativity \cup)}\]

   \[
   \iff \forall U' \subseteq \mathcal{L} : [T_1 \cup U']_s^\sigma = [T_2 \cup U']_s^\sigma \quad \text{(Def. } \equiv_s^\sigma)\]

   \[
   \iff \forall U \subseteq \mathcal{L} : [T_1 \cup U]_s^\sigma = [T_2 \cup U]_s^\sigma \quad \text{(rename } U' = U)\]

2. \[
T_1, T_2 \in [T]_s^\sigma
\]

   \[
   \iff [T_1]_s^\sigma = [T_2]_s^\sigma = [T]_s^\sigma
   \]

   \[
   \implies [T_1 \cup T_2]_s^\sigma = [T_2 \cup T_2]_s^\sigma = [T_2]_s^\sigma = [T]_s^\sigma
   \]

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

19
3.

\[ T_1, T_3 \in [T]_s^\sigma \land T_1 \subseteq T_2 \subseteq T_3 \]
\[ \implies (T_1 \cup T_2 \in [T]_s^\sigma \iff T_3 \cup T_2 \in [T]_s^\sigma) \land T_1 \subseteq T_2 \subseteq T_3 \]
\[ \implies (T_2 \in [T]_s^\sigma \iff T_3 \in [T]_s^\sigma) \]
\[ \implies T_2 \in [T]_s^\sigma \]

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 \( \subseteq \) on strong equivalence classes where two classes are in \( \subseteq \)-relation if and only if there are \( \subseteq \)-comparable representatives of the classes.

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

\[ [S]_s^\sigma \subseteq [T]_s^\sigma \iff \exists S' \in [S]_s^\sigma : \exists T' \in [T]_s^\sigma : S' \subseteq T' \]

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

**Lemma 15.** The relation \( \subseteq \) is a partial order, that is, reflexive, antisymmetric and transitive.

**Proof.** Reflexivity is clear.

antisymmetric: Let \([S]_s^\sigma \subseteq [T]_s^\sigma\) and \([T]_s^\sigma \subseteq [S]_s^\sigma\). 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

\[ [S]_s^\sigma = [S_1 \cup S_2]_s^\sigma \quad (S_1, S_2 \in [S]_s^\sigma; \text{Lemma 14, Item 2}) \]
\[ = [T_2 \cup S_1 \cup S_2]_s^\sigma \quad (T_2 \subseteq S_2) \]
\[ = [T_1 \cup T_2 \cup S_1 \cup S_2]_s^\sigma \quad ([T_2]_s^\sigma = [T_1 \cup T_2]_s^\sigma) \]
\[ = [T_1 \cup T_2 \cup S_1]_s^\sigma \quad ([S_1 \cup S_2]_s^\sigma = [S_1]_s^\sigma) \]
\[ = [T_1 \cup T_2]_s^\sigma \quad (S_1 \subseteq T_1) \]
\[ = [T]_s^\sigma \quad (T_1, T_2 \in [T]_s^\sigma; \text{Lemma 14, Item 2}) \]

transitive: Let \([T_1]_s^\sigma \subseteq [T_2]_s^\sigma\) and \([T_2]_s^\sigma \subseteq [T_3]_s^\sigma\). Then there exist \(T'_1 \in [T_1]_s^\sigma\), \(T'_2, T'_3 \in [T_2]_s^\sigma\) and \(T'_3 \in [T_3]_s^\sigma\) with \(T'_1 \subseteq T'_2 \subseteq T'_3\). Since \([T_2]_s^\sigma = [T'_2]_s^\sigma\), we can conclude that \([T'_2 \cup T'_3] = [T'_2 \cup T'_3] = [T'_3]_s^\sigma\). 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 \([T_1]_s^\sigma \subseteq [T_3]_s^\sigma\).

It follows from Lemma 14 in particular that in the case of logics \((\mathcal{L}, \sigma)\) 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 \((\mathcal{L}, \sigma)\) be a logic with strong equivalence relation \(\equiv_s\). For an equivalence class \([T]_s^\sigma \in (2^\mathcal{L})_{\equiv_s} = \{[T]_s^\sigma \mid T \subseteq \mathcal{L}\}\), we define its *cover* to be the set

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

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

Roughly, the existence of greatest elements in equivalence classes guarantees that these classes are closed under arbitrary set unions.\(^7\) 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 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 \((\mathcal{L}, \sigma)\) be a covered logic with strong equivalence relation \(\equiv_s\). The pair \((2^\mathcal{L})_{\equiv_s}, \sqsubseteq\) is a complete lattice with operations

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

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

- \(D = \bigcup_{C \in \mathcal{C}} C\) is the least upper bound of \(\mathcal{C}\):
  
  For any \(C \in \mathcal{C}\), we get \(\widehat{C} \subseteq \bigcup_{B \in \mathcal{B}} \widehat{B}\) immediately and \(\widehat{C} \in \mathcal{C}\) since \((\mathcal{L}, \sigma)\) is covered. The fact that \(\bigcup_{B \in \mathcal{B}} \widehat{B} \in [\bigcup_{B \in \mathcal{B}} \widehat{B}]_s^\sigma = D\) is also immediate, whence \(C \subseteq D\) 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 \(C \subseteq E\), there are \(T_C \in C\) and \(T'_C \in E\) such that \(T_C \subseteq T'_C\). Since \((\mathcal{L}, \sigma)\) is covered, \([T_C]_s^\sigma = C = [\widehat{C}]_s^\sigma\) whence 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 \(\forall C \in \mathcal{C} : \exists T'_C \in E : \widehat{C} \cup T'_C \subseteq E\). 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 \(\widehat{B} \cup T'_B \subseteq \widehat{E}\). 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 \((\mathcal{L}, \sigma)\) is covered. By

\(^7\) Strong equivalence classes are always closed under set unions with non-empty, finite index sets (Lemma 14).
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 \( D \subseteq E \) and \( D \) is the least upper bound of \( \mathcal{C} \).

- \( D = \bigcap_{C \in \mathcal{C}} C \) is the greatest lower bound of \( \mathcal{C} \):

As above, \( C \in \mathcal{C} \) implies \( \bigcap_{B \in \mathcal{B}} \widehat{B} \subseteq \widehat{C} \), whence by \( \widehat{C} \in \mathcal{C} \) and \( \bigcap_{B \in \mathcal{B}} \widehat{B} \in D \) we get \( D \subseteq C \). 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 \( |T_C|^s = |T'_B|^s = E \) whence \( |T'_C \cup T_B|^s = |T'_B \cup T'_C|^s = |T'_B|^s \) by strong equivalence and since \( T'_B \subseteq T_B \). Therefore, \( T'_C \cup T_B \in E \), that is, \( T'_C \subseteq T'_C \cup T_B \subseteq B \) with \( \widehat{B} \in B \) since \((\mathcal{L}, \sigma)\) was arbitrarily chosen, we get the following: for every \( B \in \mathcal{C} \), we have that \( T'_C \subseteq \widehat{B} \). Consequently \( T'_C \subseteq \bigcap_{B \in \mathcal{B}} \widehat{B} \) with \( T'_C \in E \) and \( \bigcap_{B \in \mathcal{B}} \widehat{B} \in D \). Thus \( E \subseteq D \) and \( D \) is the greatest lower bound of \( \mathcal{C} \).

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

\[
\bigcup_{C \in \emptyset} C = \left[ \bigcup_{C \in \emptyset} \widehat{C} \right]^\sigma_s = [\emptyset]^\sigma_s \quad \text{and} \quad \bigcap_{C \in \emptyset} C = \left[ \bigcap_{C \in \emptyset} \widehat{C} \right]^\sigma_s = [\mathcal{L}]^\sigma_s.
\]

As we show next, it follows that the mapping \( |^\sigma_s : 2^\mathcal{L} \to (2^\mathcal{L})_{/\equiv_s} \) assigning a theory \( T \) its equivalence class \( |T|^s \) 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 one of sets).

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

\[
\left[ \bigcup_{T \in \mathcal{T}} T \right]^\sigma_s = \bigcup_{T \in \mathcal{T}} |T|^s_s
\]

Proof. Let \( \mathcal{T} \subseteq 2^\mathcal{L} \) and denote \( \mathcal{C} = \{|T|^s_s \mid T \in \mathcal{T}\} \). Clearly \( \bigcup_{T \in \mathcal{T}} |T|^s_s = \bigcup \mathcal{C} \). Consequently, it suffices to show that

\[
\left[ \bigcup_{T \in \mathcal{T}} T \right]^\sigma_s = \bigcup \mathcal{C}.
\]

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

Let \( C \in \mathcal{C} \). Then there is a \( T \in \mathcal{T} \) with \( C = |T|^s_s \). Clearly \( T \in \mathcal{C} \) and

\[
T \subseteq \bigcup_{S \in \mathcal{T}} S \subseteq \bigcup_{S \in \mathcal{T}} |S|^s_s = \widehat{D}
\]

22
Example 9. Let \( \mathcal{L} \) be an arbitrary \( \mathcal{C} \) covered. Thus \( \mathcal{C} \subseteq \mathcal{D} \). Since \( \mathcal{C} \in \mathcal{C} \) was arbitrarily chosen, \( \mathcal{D} \) is an upper bound of \( \mathcal{C} \).

Now let \( \mathcal{E} \) be any upper bound of \( \mathcal{C} \). Consider an arbitrary \( \mathcal{C} \in \mathcal{C} \). There clearly is a \( \mathcal{T} \in \mathcal{T} \) with \( \mathcal{C} = [T]_S^T \) and since \( \mathcal{E} \) is an upper bound for \( \mathcal{C} \) there also are \( \mathcal{T}_C \in \mathcal{C} \) and \( \mathcal{T}_C \subseteq \mathcal{E} \) with \( \mathcal{T}_C \subseteq \mathcal{E} \).

We have \( \mathcal{E} \in \mathcal{T}_C \) and \( \mathcal{T}_C \subseteq \mathcal{E} \). Then clearly \( \bigcup_{\mathcal{T} \subseteq \mathcal{T}_C} \mathcal{T} \subseteq \bigcup_{\mathcal{T} \subseteq \mathcal{T}_C} (\mathcal{T} \cup \mathcal{T}_C) \subseteq \mathcal{E} \), and by \( \bigcup_{\mathcal{T} \subseteq \mathcal{T}_C} \mathcal{T} \subseteq \mathcal{D} \) and \( \mathcal{E} \subseteq \mathcal{E} \) (the logic \( (\mathcal{L}, \sigma) \) is covered) we get \( \mathcal{D} \subseteq \mathcal{E} \). Since \( \mathcal{E} \) was an arbitrarily chosen upper bound of \( \mathcal{C} \), \( \mathcal{D} \) is the least upper bound of \( \mathcal{C} \).

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 \( \bigcap_{\mathcal{T} \in \mathcal{T}} [\mathcal{T}]_S^\mathcal{T} \subseteq \bigcap_{\mathcal{T} \in \mathcal{T}} [\mathcal{T}]_S^\mathcal{T} \), as

\[
\bigcap_{\mathcal{T} \in \mathcal{T}} \mathcal{T} \subseteq \bigcap_{\mathcal{T} \in \mathcal{T}} \mathcal{T}\mathcal{S}^\mathcal{T} = \left( \bigcap_{\mathcal{T} \in \mathcal{T}} \mathcal{T}\mathcal{S}^\mathcal{T} \right) \mathcal{S}^\mathcal{T} = \bigcap_{\mathcal{T} \in \mathcal{T}} [\mathcal{T}]_S^\mathcal{T}.
\]

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

Example 9. Let \( \mathcal{L} = \{a, b\} \) and \( \equiv_s \) such that \( \emptyset \neq \equiv_s \{a\} \equiv_s \{b\} \equiv_s \{a, b\} \). Then we get:

\[
\begin{align*}
\{\{a\} \cap \{b\}\}^S_s &= \emptyset^S_s = \emptyset \\
\{\{a\}\}^S_s \cap \{\{b\}\}^S_s &= \{\{a, b\}\}^S_s = \{\{a\}, \{b\}, \{a, b\}\}^S_s = \{\{a, b\}\}^S_s = \{\{a\}, \{b\}, \{a, b\}\} \neq \emptyset
\end{align*}
\]

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 \( \mathcal{T} \) of the original language to the union of all strong equivalence classes \( [S]_S^\mathcal{T} \) that are in \( \equiv_s \)-relation to the class \( [T]_S^\mathcal{T} \) of the input theory.

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

\[
\sigma' : 2^{2^\mathcal{L}} \to 2^{2^\mathcal{L}}, \quad \mathcal{T} \mapsto \bigcup_{S \subseteq 2^\mathcal{L}, \mathcal{S} \subseteq [S]_S^\mathcal{T}} [S]_S^\mathcal{T}
\]

Proof. characterization: We have to show \( \forall \mathcal{T_1}, \mathcal{T_2} \subseteq \mathcal{L} : \sigma'(\mathcal{T_1}) = \sigma'(\mathcal{T_2}) \iff [\mathcal{T_1}]_S^\mathcal{T} = [\mathcal{T_2}]_S^\mathcal{T} \).

Let \( \mathcal{T_1}, \mathcal{T_2} \subseteq \mathcal{L} \).

\( \Rightarrow \) : Let \( \sigma'(\mathcal{T_1}) = \sigma'(\mathcal{T_2}) \). Then by definition of \( \sigma' \) we get

\[
\bigcup_{S \subseteq 2^\mathcal{L}, [\mathcal{T_1}]_S^\mathcal{T} \subseteq [S]_S^\mathcal{T}} [S]_S^\mathcal{T} = \bigcup_{S \subseteq 2^\mathcal{L}, [\mathcal{T_2}]_S^\mathcal{T} \subseteq [S]_S^\mathcal{T}} [S]_S^\mathcal{T}
\]

23
that is, for any \( S \in 2^L \) we find that \([T_1]_s^\sigma \subseteq [S]_s^\sigma\) iff \([T_2]_s^\sigma \subseteq [S]_s^\sigma\). (Clearly, the set \( \{ [S]_s^\sigma \mid S \in 2^L \} \) corresponds to a partition of \( 2^L \); thus for any element \( T \in \bigcup_{S \in 2^L} [T_1]_s^\sigma \subseteq [S]_s^\sigma \) we have \([T_1]_s^\sigma \subseteq [T]_s^\sigma\), and if also \( T \in \bigcup_{S \in 2^L} [T_2]_s^\sigma \subseteq [S]_s^\sigma \), then \([T_2]_s^\sigma \subseteq [T]_s^\sigma\) as well. Symmetry applies to obtain the above conclusion.) In particular, \( T_1, T_2 \in 2^L \) whence \([T_1]_s^\sigma \subseteq [T_2]_s^\sigma\) iff \([T_2]_s^\sigma \subseteq [T_1]_s^\sigma\). This shows \([T_1]_s^\sigma \subseteq [T_2]_s^\sigma\) and \([T_2]_s^\sigma \subseteq [T_1]_s^\sigma\), that is, \([T_1]_s^\sigma = [T_2]_s^\sigma\).

\[\text{Let } [T_1]_s^\sigma = [T_2]_s^\sigma. \text{ Then immediately } \]

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

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

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

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

Using this relationship, we get that

\[\sigma'(\bigcup_{T \in \mathcal{T}} T) = \bigcup_{S \in 2^L, [\bigcup_{T \in \mathcal{T}} T]_s^\sigma \subseteq [S]_s^\sigma} [S]_s^\sigma \]

\[= \bigcup_{S \in 2^L, \forall T \in \mathcal{T}, [T]_s^\sigma \subseteq [S]_s^\sigma} [S]_s^\sigma \]

\[= \bigcup_{T \in \mathcal{T}} \left( \bigcup_{S \in 2^L, [T]_s^\sigma \subseteq [S]_s^\sigma} [S]_s^\sigma \right) \]

\[= \bigcap_{T \in \mathcal{T}} \sigma'(T) \ \square \]

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

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

\([\{ a, b \}]_s^\sigma = \{ \{ a, c \} \}_s^\sigma = \{ \{ b, c \} \}_s^\sigma = \{ \{ a, b, c \} \}_s^\sigma = \{ \{ a, b \}, \{ a, c \}, \{ b, c \}, \{ a, b, c \} \} \).

The resulting lattice of strong equivalence classes is depicted below.
According to Theorem 18, the characterization logic $\sigma' : 2^L \rightarrow 2^{2^L}$ assigns as follows:

\[
\sigma'(\emptyset) = \bigcup_{S \in 2^L, \emptyset \subseteq S} S_s^\sigma = \bigcup_{S \in 2^L} S_s^\sigma = 2^L
\]

\[
\sigma'(\{a\}) = \bigcup_{S \in 2^L, \{a\} \subseteq S} S_s^\sigma = \{\{a\}\}^{\sigma'} \cup \{\{a, b, c\}\}^{\sigma'} = \{\{a\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}
\]

\[
\sigma'(\{b\}) = \bigcup_{S \in 2^L, \{b\} \subseteq S} S_s^\sigma = \{\{b\}\}^{\sigma'} \cup \{\{a, b, c\}\}^{\sigma'} = \{\{b\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}
\]

\[
\sigma'(\{c\}) = \bigcup_{S \in 2^L, \{c\} \subseteq S} S_s^\sigma = \{\{c\}\}^{\sigma'} \cup \{\{a, b, c\}\}^{\sigma'} = \{\{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}
\]

\[
\sigma'(\{a, b\}) = \bigcup_{S \in 2^L, \{a, b\} \subseteq S} S_s^\sigma = \{\{a, b\}\}^{\sigma'} = \{\{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}
\]

\[
\sigma'(\{a, b, c\}) = \bigcup_{S \in 2^L, \{a, b, c\} \subseteq S} S_s^\sigma = \{\{a, b, c\}\}^{\sigma'} = \{\{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}
\]

Now it holds for example that

\[
\sigma'(\{a\}) \cap \sigma'(\{b\}) = \{\{a\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\} \cap \{\{b\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\} = \{\{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}
\]

\[
\sigma'(\{a\} \cup \{b\}) = \sigma'(\{a, b\}) \quad \diamondsuit
\]

### 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 \((\mathcal{L}, \sigma)\), the finite-theory version \((\mathcal{L}, \sigma_{\text{fin}})\) of \((\mathcal{L}, \sigma)\) 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 \((2^\mathcal{L})_{\text{fin}} = \{ T \in 2^\mathcal{L} \mid T \text{ is finite} \} \).

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

Definition 11. Let \((\mathcal{L}, \sigma)\) 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 \((\mathcal{L}, \sigma)\) if and only if:

1. \(\forall T_1, T_2 \in (2^\mathcal{L})_{\text{fin}} : \sigma'_{\text{fin}}(T_1) = \sigma'_{\text{fin}}(T_2) \iff [T_1]_{\text{fin}} = [T_2]_{\text{fin}} \quad \text{(finite-theory characterization)}\)

2. \(\forall T_1, T_2 \in (2^\mathcal{L})_{\text{fin}} : \sigma'_{\text{fin}}(T_1 \cup T_2) = \sigma'_{\text{fin}}(T_1) \cap \sigma'_{\text{fin}}(T_2) \quad \text{(finite-theory intersection)}\)

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 \(K_1, K_2 \in \mathcal{K}\):

\[
K_1 \land K_2 = K_1 \cap K_2 \quad \text{and} \quad K_1 \lor K_2 = \nbigvee \{ K_1, K_2 \}^u
\]

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 \(\sigma_{\text{fin}}(T_2) = K_2\).

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

\[
K_1 \land K_2 = \sigma_{\text{fin}}(T_1) \land \sigma_{\text{fin}}(T_2) = \sigma_{\text{fin}}(T_1) \cap \sigma_{\text{fin}}(T_2) = \sigma_{\text{fin}}(T_1 \cup T_2) \in \mathcal{K}
\]
It follows from the finite intersection property that $\sigma'_\text{in}$ is antimonotone. Now $\emptyset \subseteq T_1$ implies that $K_1 = \sigma'_\text{in}(T_1) \subseteq \sigma'_\text{in}(\emptyset)$ and likewise for $T_2$. Thus $\sigma'_\text{in}(\emptyset)$ is an upper bound of $K_1$ and $K_2$, whence the set $\{K_1, K_2\}^u$ is non-empty. 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, $\sigma'_\text{in}$ being antimonotone means that $\emptyset \subseteq T_1' \subseteq T_1$ implies $\sigma'_\text{in}(T_1) \subseteq \sigma'_\text{in}(T_1') \subseteq \sigma'_\text{in}(\emptyset)$. Thus both $K_1$ and $K_2$ have only finitely many supersets in $\mathcal{K}$, that is, the sets $K_1^\dagger$ and $K_2^\dagger$ are finite, whence $\{K_1, K_2\}^u \subseteq K_1^\dagger \cup K_2^\dagger$ is finite. Consequently, $\{K_1, K_2\}^u$ is a finite, non-empty subset of $\mathcal{K}$. It therefore possesses a greatest lower bound $K_i = \bigwedge \{K_1, K_2\}^u \in \mathcal{K}$. Since $\{K_1, K_2\}^u$ is closed under intersection we have that $K_i = \bigwedge \{K_1, K_2\}^u = \bigcap \{K_1, K_2\}^u \in \{K_1, K_2\}^u$ 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.

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

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

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

$\phi$ is bijective: The proof is as in the general case.

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

\[
\begin{align*}
\phi(K_1 \land K_2) &= \phi(K_1 \cap K_2) \\
&= \phi(\sigma'\text{in}(T_1) \cap \sigma'\text{in}(T_2)) \quad (\text{Def. } \land) \\
&= \phi(\sigma'\text{in}(T_1 \cup T_2)) \quad (\text{intersection } \sigma'\text{in}) \\
&= \sigma''\text{in}(T_1 \cup T_2) \quad (\text{Def. } \phi) \\
&= \sigma''\text{in}(T_1) \cap \sigma''\text{in}(T_2) \quad (\text{intersection } \sigma''\text{in}) \\
&= \phi(\sigma'\text{in}(T_1)) \cap \phi(\sigma'\text{in}(T_2)) \quad (\text{Def. } \phi) \\
&= \phi(K_1) \cap \phi(K_2) \quad (\text{assumption}) \\
&= \phi(K_1) \land \phi(K_2) \quad (\text{Def. } \land)
\end{align*}
\]
With reasonable effort, we can also show that

\[
\phi(K_1 \lor K_2) \\
= \phi(\bigwedge \{K_1, K_2\}^u) \\
\text{(Def. } \lor) \\
= \bigwedge \phi(\{K_1, K_2\}^u) \\
\text{ (} \phi \text{ preserves } \land) \\
= \bigwedge \phi(\{K \in \mathcal{K} \mid K_1 \subseteq K, K_2 \subseteq K\}) \\
\text{(Def. } .^u) \\
= \bigwedge \{\phi(K) \mid K \in \mathcal{K}', K_1 \subseteq K, K_2 \subseteq K\} \\
\text{(notation)} \\
= \bigwedge \{\phi(\sigma'_\text{fin}(T)) \mid T \in (2^C)^\text{fin}, \sigma'_\text{fin}(T_1) \subseteq \sigma'_\text{fin}(T), \sigma'_\text{fin}(T_2) \subseteq \sigma'_\text{fin}(T)\} \\
\text{ (} \sigma'_\text{fin} \text{ is onto } \mathcal{K}'\) \\
= \bigwedge \{\phi(\sigma''\text{fin}(T)) \mid T \in (2^C)^\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)\} \\
\text{(elementary)} \\
= \bigwedge \{\phi(\sigma''\text{fin}(T)) \mid T \in (2^C)^\text{fin}, [T_1 \cup T]^g = [T_1]^g, [T_2 \cup T]^g = [T_2]^g\} \\
\text{(characterization } \sigma''\text{fin}) \\
= \bigwedge \{\sigma''\text{fin}(T) \mid T \in (2^C)^\text{fin}, \sigma''\text{fin}(T_1 \cap T) = \sigma''\text{fin}(T_1), \sigma''\text{fin}(T_2 \cap T) = \sigma''\text{fin}(T_2)\} \\
\text{(intersection } \sigma''\text{fin}) \\
= \bigwedge \{\sigma''\text{fin}(T) \mid T \in (2^C)^\text{fin}, \sigma''\text{fin}(T_1) \subseteq \sigma''\text{fin}(T), \sigma''\text{fin}(T_2) \subseteq \sigma''\text{fin}(T)\} \\
\text{(elementary)} \\
= \bigwedge \{K \mid K \in \mathcal{K}'', \sigma''\text{fin}(T_1) \subseteq K, \sigma''\text{fin}(T_2) \subseteq K\} \\
\text{ (} \sigma''\text{fin} \text{ is onto } \mathcal{K}''\) \\
= \bigwedge \{\sigma''\text{fin}(T_1), \sigma''\text{fin}(T_2)\}^u \\
\text{(Def. } .^u) \\
= \sigma''\text{fin}(T_1) \lor \sigma''\text{fin}(T_2) \\
\text{(Def. } \lor) \\
= \phi(\sigma''\text{fin}(T_1)) \lor \phi(\sigma''\text{fin}(T_2)) \\
\text{(Def. } \phi) \\
= \phi(K_1) \lor \phi(K_2) \\
\text{(assumption)}
\]
Thus $\phi$ is a structure-preserving bijection from $K'$ to $K''$, and the two lattices are isomorphic.

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.

**Theorem 21.** Let $(\mathcal{L}, \sigma)$ be a full logic. Then a finite-theory characterization logic for $(\mathcal{L}, \sigma)$ is given by $(\mathcal{L}, \sigma'_\text{fin})$ with

$$
\sigma'_\text{fin} : (2^\mathcal{L})_{\text{fin}} \to 2^\mathcal{L}, \quad T \mapsto \bigcup_{S \in (2^\mathcal{L})_{\text{fin}}} [S]_{\text{fin}}^T
$$

**Proof.** finite intersection: We show that for all $T \in (2^\mathcal{L})_{\text{fin}}$, we find $\sigma'_\text{fin}(T) = \bigcap_{t \in T} \sigma'_\text{fin}(\{t\})$.

Consider $T \in (2^\mathcal{L})_{\text{fin}}$.

$\subseteq$: Let $U \in \sigma'_\text{fin}(T)$. Hence, there is an $S \in (2^\mathcal{L})_{\text{fin}}$ such that $U \in [S]_{\text{fin}}^T$ and $T \subseteq S$. Consequently, for any $t \in T$, we find $\{t\} \subseteq T \subseteq S$ and thus $[S]_{\text{fin}}^T \subseteq \sigma'_\text{fin}(\{t\})$, showing $U \in \bigcap_{t \in T} \sigma'_\text{fin}(\{t\})$.

$\supseteq$: Consider now $U \in \bigcap_{t \in T} \sigma'_\text{fin}(\{t\})$. This means that for any $t \in T$ there exists an $S_t \in (2^\mathcal{L})_{\text{fin}}$ such that $U \in [S_t]_{\text{fin}}^T$ (that is, $[U]_{\text{fin}}^T = [S_t]_{\text{fin}}^T$) and $\{t\} \subseteq S_t$. It follows that for any $t, t' \in T$, we find $[S_t]_{\text{fin}}^T = [U]_{\text{fin}}^T = [S_{t'}]_{\text{fin}}^T$. Let us fix a certain $\bar{t} \in T$. In particular, $U \in [S_{\bar{t}}]_{\text{fin}}^T$. Now consider the set $\bigcup_{t \in T} S_t$, which is finite since $T$ is finite and each $S_t$ is finite. Furthermore, $\bigcup_{t \in T} S_t \in [S_{\bar{t}}]_{\text{fin}}^T$, that is, $[\bigcup_{t \in T} S_t]_{\text{fin}}^T = [S_{\bar{t}}]_{\text{fin}}^T$ by Lemma 14. Since for all $t \in T$ we have $t \in S_t$, we conclude that $T = \bigcup_{t \in T} \{t\} \subseteq \bigcup_{t \in T} S_t$. In turn, this yields $U \in \sigma'_\text{fin}(T)$.

finite-theory characterization: We show that for all $T, T' \in (2^\mathcal{L})_{\text{fin}}$, we find $[T_1]_{\text{fin}}^T = [T_2]_{\text{fin}}^T$ if and only if $\sigma'_\text{fin}(T_1) = \sigma'_\text{fin}(T_2)$.

if: Let $T_1, T_2 \in (2^\mathcal{L})_{\text{fin}}$ with $\sigma'_\text{fin}(T_1) = \sigma'_\text{fin}(T_2)$. Firstly, the definition of $\sigma'_\text{fin}$ yields that for each $S \in (2^\mathcal{L})_{\text{fin}}$ we have $[S]_{\text{fin}}^T \subseteq \sigma'_\text{fin}(S)$. In combination with the presumption, this means that $[T_1]_{\text{fin}}^T \subseteq \sigma'_\text{fin}(T_1) = \sigma'_\text{fin}(T_2)$ and $[T_2]_{\text{fin}}^T \subseteq \sigma'_\text{fin}(T_2) = \sigma'_\text{fin}(T_1)$. Hence, there is a set $S_1 \in [T_1]_{\text{fin}}^T$ with $T_1 \subseteq S_1$, whence $[S_1]_{\text{fin}}^T \subseteq [T_1]_{\text{fin}}^T$.

Likewise, there is a set $S_2 \in [T_1]_{\text{fin}}^T$ with $T_2 \subseteq S_2$, whence also $[S_2]_{\text{fin}}^T \subseteq [T_1]_{\text{fin}}^T$. By Lemma 15 (saying that $\subseteq$ is antisymmetric), we get $[T_1]_{\text{fin}}^T = [T_2]_{\text{fin}}^T$.

only if: Let $T_1, T_2 \in (2^\mathcal{L})_{\text{fin}}$ with $[T_1]_{\text{fin}}^T = [T_2]_{\text{fin}}^T$ and consider any $T \in \sigma'_\text{fin}(T_1)$. Then there is an $S \in (2^\mathcal{L})_{\text{fin}}$ such that $T \in [S]_{\text{fin}}^T$ and $T_1 \subseteq S$. Clearly there is a $U \in (2^\mathcal{L})_{\text{fin}}$ with $T_1 \cup U = S$. Since $[T_1]_{\text{fin}}^T = [T_2]_{\text{fin}}^T$, by presumption, we can apply Lemma 14, which yields $T \in [S]_{\text{fin}}^T = [T_1 \cup U]_{\text{fin}}^T = [T_2 \cup U]_{\text{fin}}^T$. Obviously, we have $T_2 \cup U \in (2^\mathcal{L})_{\text{fin}}$ with $T_2 \subseteq T_2 \cup U$ and the definition of $\sigma'_\text{fin}$ yields $T \in \sigma'_\text{fin}(T_2)$. This shows $\sigma'_\text{fin}(T_1) \subseteq \sigma'_\text{fin}(T_2)$; the reverse inclusion $\sigma'_\text{fin}(T_2) \subseteq \sigma'_\text{fin}(T_1)$ holds by symmetry.

$\square$
Intuitively, in this canonical construction of a characterization semantics \( \sigma' \) (akin to Herbrand interpretations in first-order logic), the model set of a theory \( T \) is the set of all theories that are strongly equivalent to some supertheory of \( T \).

4. Applying Canonical Constructions to Nonmonotonic Formalisms

In the previous section we have seen that (under certain condition) 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 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 (Dung, 1995). Surprisingly, we get a meaningful result very similar to the recently introduced Dung logics (Baumann & Brewka, 2015).

4.1 Abstract Argumentation Theory

We start with a brief introduction to Dung’s argumentation theory (cf. (Baroni, Caminada, & Giacomin, 2018) 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 (Baumann & Spanring, 2015, 2017) it is common to assume that \( A \), the set of arguments, is a finite subset of a fixed infinite background set \( U \). Let us denote the class of all finite AFs by \( AF_{\text{fin}} \). An (extension-based) argumentation semantics is a function \( \rho : AF_{\text{fin}} \rightarrow 2^2U \) where elements of \( \rho(F) \) are called \( \rho \)-extensions of \( F \). The most prominent one is stable semantics (abbreviated by \( \text{stb} \)) which was already defined by Dung in 1995. A set \( E \) is a \( \text{stb} \)-extension of \( F \) if 1. there are no \( a, b \in A \), s.t. \( (a, b) \in R \) (conflict-freeness) and 2. for any \( c \in A \setminus E \), there is an \( a \in A \), s.t. \( (a, c) \in R \) (full range).

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

**Definition 12.** Given an argumentation semantics \( \rho \). Two AFs \( F \) and \( G \) are strongly \( \rho \)-equivalent if for any \( H \in AF_{\text{fin}} \), \( \rho(F \cup H) = \rho(G \cup H) \). For short, \( F \equiv^\rho G \). 

The first work regarding characterizing strong equivalence for AFs was (Oikarinen & Woltran, 2011). 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. Consider the following definition and characterization theorem. An exhaustive overview on well-known semantics, further equivalence notions and their characterization can be found in (Baumann, 2018).
**Definition 13.** Given an AF $F = (A, R)$. The $\text{stb}$-kernel $F^{k(\text{stb})} = (A, R^{k(\text{stb})})$ is defined with $R^{k(\text{stb})} = R \setminus \{(a, b) \mid a \neq b \land (a, a) \in R\}$.

**Theorem 22 ((Oikarinen & Woltran, 2011)).** For two AFs $F, G$ we have:

$$F \equiv^{\text{stb}}_s G \iff F^{k(\text{stb})} = G^{k(\text{stb})}.$$  

Let us illustrate the introduced concepts with an example.

**Example 11.** Consider the following six AFs.

\[
\begin{align*}
F &: a \quad b \quad c \quad d \\
F^{k(\text{stb})} &: a \quad b \quad c \quad d \\
F \cup H &: a \quad b \quad c \quad d \quad e \\
G &: b \quad c \quad d \\
G^{k(\text{stb})} &: b \quad c \quad d \\
G \cup H &: b \quad c \quad d \quad e \\
\end{align*}
\]

The AFs $F$ and $G$ possess the same stable extensions, namely $\text{stb}(F) = \text{stb}(G) = \{\{b, d\}\}$. However, both frameworks are not strongly $\text{stb}$-equivalent since $F^{k(\text{stb})} \neq G^{k(\text{stb})}$. A witnessing framework is given by $H$. Indeed $\text{stb}(F \cup H) = \{\{a, d, e\}\} \neq \emptyset = \text{stb}(G \cup H)$.

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 $\rho$-logic which formally captures a specific argumentation semantics $\rho$ on the level of theories.

**Definition 14.** Let $\mathcal{U}$ be a background set of arguments and $\rho$ be an AF semantics. A $\rho$-logic is a triple $(\mathcal{L}_{\text{AF}}, \mathcal{I}, \sigma_\rho)$ where $\mathcal{L}_{\text{AF}} = \{\{\{a\}\}, \{\{a, b\}\}, \{\{a, b\}\} \mid a, b \in \mathcal{U}\}$, $\mathcal{I} = 2^\mathcal{U}$ and $\sigma_\rho: 2^\mathcal{L}_{\text{AF}} \to 2^\mathcal{I}$ with $\sigma_\rho(T) = \rho\left(\bigcup_{t \in T} t\right)$.  

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 $T = \{(b,c), (c,b), \} \cup \{ (c,d), (e,d), \} \cup \{ (c,c) \}$ and

$S = T \cup \{(a,0), (b,0), (c,0)\}$. Please observe that $\bigcup_{t \in T} t = \bigcup_{s \in S} s = G$ as depicted in Example 11. Moreover by definition we have $\sigma_{stb}(T) = \sigma_{stb}(S) = stb(G) = \{ b, d \}$. \hfill \diamond

The following functions (restricted to the finite case) will be frequently used. First, we define the associated $AF$ of a given theory via $AF : (2^{\mathcal{L}_{AF}})^{fin} \rightarrow AF_{fin}$, where $T \mapsto \bigcup_{t \in T} t$. 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 $C : AF_{fin} \rightarrow (2^{\mathcal{L}_{AF}})^{fin}$ 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 $AF$s can be reflected by considering the finite-theory versions of $\rho$-logics. Before applying our canonical construction presented in Theorem 21 we have to ensure that a constructed $\rho$-logic correctly reflects $AF$s under semantics $\rho$. 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 $\sigma_{\rho}$.

Proposition 23. Let $(\mathcal{L}_{AF}, I, \sigma_{\rho})$ be a $\rho$-logic and consider any theories $S, T \subseteq \mathcal{L}_{AF}$.

1. $AF(S \cup T) = AF(S) \cup AF(T)$ and

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

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

$$
AF(S \cup T) = \bigcup_{u \in S \cup T} u \quad \text{(Definition } AF)\\
= \bigcup_{s \in S} s \cup \bigcup_{t \in T} t \quad \text{(associativity } \bigcup)\\
= AF(S) \cup AF(T) \quad \text{(Definition } AF)
$$

2.

$$
\sigma_{\rho}(S \cup T) = \rho\left( \bigcup_{u \in S \cup T} u \right) \quad \text{(Definition } \sigma_{\rho})\\
= \rho(AF(S \cup T)) \quad \text{(Definition } AF)\\
= \rho(AF(S) \cup AF(T)) \quad \text{(Item 1)} \quad \square
$$

The following theorem shows that two $\mathcal{L}_{AF}$-theories $S$ and $T$ are strongly equivalent under $\sigma_{\rho}$ if and only if the $AF$s $AF(S)$ and $AF(T)$ are strongly equivalent under $\rho$ (denoted by $AF(S) \equiv_{\rho} AF(T)$). We mention that the theorem does not require finiteness and is thus valid for arbitrary cardinalities of theories as well as $AF$s.
Theorem 24. Let \((\mathcal{L}_AF, \mathcal{I}, \sigma, \rho)\) be a \(\rho\)-logic. For \(S, T \subseteq \mathcal{L}_AF\) we have

\[
AF(S) \equiv_s \rho AF(T) \iff [S]_s^{\sigma, \rho} = [T]_s^{\sigma, \rho}.
\]

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. \(\sigma(\rho(S \cup V)) = \sigma(\rho(T \cup V))\).

\[
\sigma(\rho(S \cup V)) = \rho(AF(S) \cup AF(V)) \quad \text{(Proposition 23)}
\]

\[
= \rho(AF(T) \cup AF(V)) \quad \text{(assumption \(AF(S) \equiv_s \rho AF(T)\))}
\]

\[
= \sigma(\rho(T \cup V)) \quad \text{(Proposition 23)}
\]

“\(\impliedby\)”: 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. \(\rho(AF(S) \cup H) = \rho(AF(T) \cup H)\).

\[
\rho(AF(S) \cup H) = \rho(AF(S) \cup AF(C(H))) \quad \text{(AF(C(H)) = H)}
\]

\[
= \sigma(\rho(S \cup C(H))) \quad \text{(Proposition 23)}
\]

\[
= \sigma(\rho(T \cup C(H))) \quad \text{(assumption \([S]_s^{\sigma, \rho} = [T]_s^{\sigma, \rho}\))}
\]

\[
= \rho(AF(T) \cup AF(C(H))) \quad \text{(Proposition 23)}
\]

\[
= \rho(AF(T) \cup H) \quad \text{(AF(C(H)) = H)} \quad \square
\]

Due to Theorem 21 we are able to present finite-theory characterization logics for any \(\rho\)-logic.

Corollary 25. Let \((\mathcal{L}_AF, \mathcal{I}, \sigma, \rho)\) be a \(\rho\)-logic. The following logic \((\mathcal{L}_AF, \kappa)\) is a finite-theory characterization logic of \((\mathcal{L}_AF, \mathcal{I}, \sigma, \rho)\):

\[
\kappa : (2^{\mathcal{L}_AF})_{fin} \to 2^{2^{\mathcal{L}_AF}}, \quad T \mapsto \bigcup_{S \subseteq (2^{\mathcal{L}_AF})_{fin}} [S]_{(\sigma, \rho)_{fin}}^{(\sigma, \rho)}.
\]

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 \(AF : 2^{2^{\mathcal{L}_AF}} \to 2^{AF_{fin}}\) where \(T \mapsto \{AF(T) \mid T \in T\}\).\(^8\) Consider the following definition. We will see that all crucial properties of \(\kappa\) transfer to \(\rho'\), that is, \(\rho'\) satisfies finite intersection and furthermore, it characterizes strong equivalence under \(\rho\).

\(^8\) We do not introduce a new symbol for the new function. Which function is meant will be clear from the context.
Definition 15. Given an argumentation semantics \( \rho : AF_{in} \to 2^{2^{AF_{in}}} \). We define the function \( \rho' : AF_{in} \to 2^{AF_{in}} \) with \( F \mapsto AF(\kappa(C(F))) \).

\[
\rho'(F) = \bigcup_{G \in AF_{in}, F \subseteq G} \{ H | H \equiv_s G \}
\]

Proposition 26. For any argumentation semantics \( \rho \) and semantics \( \rho' \) as defined above we have:

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

Proof. finite-theory characterization: “\( \leftarrow \)”: Let \( F \equiv_s G \). Hence, \( AF(C(F)) \equiv_s AF(C(G)) \). Consequently, \( [C(F)]^{\rho}_{\sigma} = [C(G)]^{\rho}_{\sigma} \) (Theorem 24). Since \( (\mathcal{L}_{AF}, \kappa) \) is a finite-theory characterization logic of \( (\mathcal{L}_{AF}, I, \sigma_{\rho}) \) (Corollary 25) 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 \( F \not\equiv_s G \). This means, \( AF(C(F)) \not\equiv_s AF(C(G)) \) and we obtain \( [C(F)]^{\rho}_{\sigma} \not\equiv \ [C(G)]^{\rho}_{\sigma} \) (Theorem 24). Consequently, \( \kappa(C(F)) \not\equiv \kappa(C(G)) \) since \( \kappa \) characterizes strong equivalence under \( \sigma_{\rho} \) (Corollary 25). Since equivalence classes are disjoint we deduce the existence of a theory \( U \), such that (without loss of generality) \( [U]^{\rho}_{\sigma} \subseteq \kappa(C(F)) \setminus \kappa(C(G)) \). 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 \( U' \in [U]^{\rho}_{\sigma} \) (Theorem 24). Hence, \( AF(\kappa(C(F))) \not\equiv AF(\kappa(C(G))) \) which means \( \rho'(F) \not\equiv \rho'(G) \) (Definition 15).

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

Proposition 27. Let \( \rho : AF_{in} \to 2^{2^{AF_{in}}} \) be a semantics and \( \rho' \) as in Definition 15. For any \( F \in AF_{in} \) we have:

\[
\rho'(F) = \bigcup_{G \in AF_{in}, F \subseteq G} \{ H | H \equiv_s G \}
\]
Proof.

\[ \rho'(F) = AF(\kappa(C(F))) \]  

(Definition 15)

\[ = AF \left( \bigcup_{S \in (2^{\mathcal{L}}\mathcal{AF})_{\mathrm{lin}}, C(F) \subseteq S} [S]_{s}^{(\sigma_{\rho})_{\mathrm{fin}}} \right) \]  

(Corollary 25)

\[ = AF \left( \bigcup_{S \in (2^{\mathcal{L}}\mathcal{AF})_{\mathrm{lin}}, C(F) \subseteq S} [S]_{s}^{(\sigma_{\rho})_{\mathrm{fin}}} \right) \]  

(Definition \(AF(\cdot)\))

\[ = \bigcup_{S \in (2^{\mathcal{L}}\mathcal{AF})_{\mathrm{lin}}, C(F) \subseteq S} \{ AF(T) \mid T \in [S]_{s}^{(\sigma_{\rho})_{\mathrm{fin}}} \} \]  

(Definition \(AF(\cdot)\))

\[ = \bigcup_{S \in (2^{\mathcal{L}}\mathcal{AF})_{\mathrm{lin}}, C(F) \subseteq S} \{ AF(T) \mid [T]_{s}^{(\sigma_{\rho})_{\mathrm{fin}}} = [S]_{s}^{(\sigma_{\rho})_{\mathrm{fin}}} \} \]  

(equivalence relation)

\[ = \bigcup_{G \in \mathcal{AF}_{\mathrm{lin}}, F \subseteq G} \{ H \mid H \equiv_{s} G \} \]  

\[ (AF(S) = G, AF(T) = H, \text{Theorem 24}) \]

\[ \square \]

Recently, Baumann and Brewka introduced so-called Dung-logics to be able to perform AGM-style revision for Dung’s abstract argumentation frameworks (Baumann & Brewka, 2015). These Dung-logics are very similar yet different from the characterization logics presented in Proposition 27. 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 \(\rho\) if and only if the singletons of \(F\) and \(G\) are ordinarily equivalent with respect to the semantics defined by Theorem 3 of Baumann and Brewka (2015).

We start with the formal definition of a Dung-logic in case of the most prominent argumentation semantics, namely stable semantics.\(^9\)

\(^9\) For more details consider the original paper.
Definition 16. The Dung-logic in case of stable semantics is a pair \((AF_{\text{fin}}, \delta)\) with
\[
\delta : 2^{AF_{\text{fin}}} \rightarrow 2^{AF_{\text{fin}}} \quad U \mapsto \bigcap_{F \in U} \text{Mod}^{k(stb)}(F)
\]
whereas \(\text{Mod}^{k(stb)}(F) = \{ G \in AF_{\text{fin}} \mid F^{k(stb)} \subseteq G^{k(stb)} \}\)

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

Observation 28. Let \(stb : AF_{\text{fin}} \rightarrow 2^{AF_{\text{fin}}}\) be stable semantics and \(\delta\) as in Definition 16. For any \(F \in AF_{\text{fin}}\) we have:
\[
\delta(\{F\}) = \bigcup_{G \in AF_{\text{fin}}, \ F^{k(stb)} \subseteq G^{k(stb)}} \{ H \mid H \equiv_{stb} G \}
\]

The only difference in comparison to \(\rho'\) is that for \(\delta\), 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 21 yields:

Corollary 29. For the finite-theory version of logic \((\mathcal{L}_{LP}, \sigma_{stb})\) of normal logic programs under stable model semantics, a finite-theory characterization logic is given by
\[
\sigma'_{stb} : (2^{\mathcal{L}_{LP}})_{\text{fin}} \rightarrow 2^{(2^{\mathcal{L}_{LP}})_{\text{fin}}}, \quad T \mapsto \bigcup_{S \in (2^{\mathcal{L}_{LP}})_{\text{fin}}, T \subseteq S} [S]_{\sigma_{stb}}^{\mathcal{L}_{LP}}
\]

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

Definition 17. Let \(P\) be a normal logic program over \(A\) and \(X \subseteq Y \subseteq A\). Define semantics \(\sigma_{SE} : (2^{\mathcal{L}_{LP}})_{\text{fin}} \rightarrow 2^{A \times A}\) by
\[
T \mapsto \{ (X, Y) \mid X \subseteq Y \text{ and } X, Y \in \sigma_{mod}(T^Y) \}
\]
where
\[
T^Y = \{ a_0 \leftarrow a_1, \ldots, a_m \mid a_0 \leftarrow a_1, \ldots, a_m, \sim a_{m+1}, \ldots, a_n \in T, a_{m+1}, \ldots, a_n \notin Y \}
\]
\[
\sigma_{mod}(T) = \{ M \subseteq A \mid \forall a_0 \leftarrow a_1, \ldots, a_m, \sim a_{m+1}, \ldots, a_n \in T : (a_1, \ldots, a_m, a_{m+1}, \ldots, a_n \notin M) \Rightarrow a_0 \in M \}
\]

SE-models characterize strong equivalence of stable models (Turner, 2001a, Theorem 1): SE-model semantics also has the intersection property (Truszczynski, 2006, Lemma 3).

Proposition 30. \((\mathcal{L}_{LP}, \sigma_{SE})\) is a characterization logic for \((\mathcal{L}_{LP}, \sigma_{stb})\).
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 29 (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, please note that the set of SE-models of a finite logic program is finite, while \( \sigma'_{stb} \) 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. Discussion

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 (Baumann & Brewka, 2015).

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 (Lifschitz et al., 2001; Turner, 2001a) and supported models (Truszczyński & Woltran, 2008), that also give rise to similar developments in default logic (Turner, 2001a; Truszczyński, 2007) and autoepistemic logic (Truszczyński & Woltran, 2008). By considering and exploiting formalism specifics, more fine-grained views on classical, strong and intermediate equivalence notions are possible (Eiter & Fink, 2003; Truszczyński, 2007; Woltran, 2008). 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 (Brewka & Woltran, 2010; Brewka, Ellmauthaler, Strass, Walner, & Woltran, 2013), a quite recent non-classical KR formalism encompassing both...
Baumann and Strass

argumentation frameworks and logic programs (Strass, 2013), for which strong equivalence has not been studied yet.

Acknowledgments

The authors would like to thank the reviewers of KR 2016 for helpful suggestions for improvement of an earlier version of this paper.

References


