Chapter 2
Consequence

Abstract Belief revision theory assumes that the agents reason according to a logic. As an abstraction that encompasses many different logics, we can consider a logic as a pair $\langle \mathscr{L}, Cn \rangle$ such that $\mathscr{L}$ is the *language* of the logic and Cn is the *consequence operation* $Cn : 2^{\mathscr{L}} \to 2^{\mathscr{L}}$ that gives the consequences of a set of sentences. We are particularly interested in Tarskian logics and certain properties that they may satisfy e.g., compactness, decomposability, distribuitivity, etc. In this chapter, Tarskian logics, some logical properties, and relations between these properties will be presented.

Keywords Consequence operator · Consequence relation · Tarskian logics · Lattices

As usual, the symbol $2^{\mathscr{L}}$ represent the set of subsets of $\mathscr{L}$. We represent sets of sentences (subsets of $\mathscr{L}$) using upper case letters $A, B, C, \ldots$. Sentences of the language are represented with lower case Greek letters $\alpha, \beta, \ldots$.

Given a logic $\langle \mathscr{L}, Cn \rangle$, consider a sentence $\alpha \in \mathscr{L}$ and two sets of sentences $A, B \in 2^{\mathscr{L}}$:

1. α is a *consequence* of B iff $\alpha \in Cn(A)$.
2. A is a *consequence* of B iff every element of A is a consequence of B i.e., $A \subseteq Cn(B)$.
3. A and B are *equivalent* iff $Cn(A) = Cn(B)$.
4. A is *trivial* iff $Cn(A) = \mathscr{L}$.

Following the above definitions, a set A is *not* a consequence of B iff $A \nsubseteq Cn(B)$ i.e., there is at least one sentence $\alpha \in A$ which is not a consequence of B. An alternative definition would impose that A is not a consequence of B iff $B \cap Cn(A) = \emptyset$. We will use the former.

M. M. Ribeiro, *Belief Revision in Non-Classical Logics*,
SpringerBriefs in Computer Science,
DOI: 10.1007/978-1-4471-4186-0_2, © The Author(s) 2013

2.1 Tarskian Consequence Operator

Some very basic properties that a consequence operator may satisfy are the following:

monotonicity: if $A \subseteq B$ then $Cn(A) \subseteq Cn(B)$.
idempotence: $Cn(A) = Cn(Cn(A))$.
inclusion: $A \subseteq Cn(A)$.

A consequence operator that satisfies these properties is called *Tarskian*. In general, if a consequence operator of a logic satisfies certain property, we will simply say that the logic itself satisfies it. For example, a logic $\langle \mathscr{L}, Cn \rangle$ with a Tarskian consequence operator is also called *Tarskian*.

Although Tarskian logics do not encompass every logic in the literature (e.g., linear logic [Gir87] and non-monotonic logics [AW97] are not Tarskian), it encompasses enough logics so that we will only consider them in this book. Hence, from now on, whenever we write logic we mean Tarskian logic.

The following are simple lemmas about Tarskian consequence operator that will be used throughout the book without reference:

Lemma 2.1 *Let A and B be sets of sentences in a Tarskian logic $\langle \mathscr{L}, Cn \rangle$. Then the following equations hold*:

1. $Cn(Cn(A) \cup Cn(B)) = Cn(A \cup B)$
2. $Cn(Cn(A) \cap Cn(B)) = Cn(A) \cap Cn(B)$

Proof

1. By *inclusion* we have $A \subseteq Cn(A)$ and $B \subseteq Cn(B)$. It follows that $A \cup B \subseteq Cn(A) \cup Cn(B)$ and by *monotonicity* $Cn(A \cup B) \subseteq Cn(Cn(A) \cup Cn(B))$.
   By *monotonicity* both $Cn(A)$ and $Cn(B)$ are subsets of $Cn(A \cup B)$. It follows that $Cn(A) \cup Cn(B) \subseteq Cn(A \cup B)$ and by *idempotence* $Cn(Cn(A) \cup Cn(B)) \subseteq Cn(A \cup B)$.
2. Of course, $Cn(A) \cap Cn(B) \subseteq Cn(A)$ and by *idempotence* $Cn(Cn(A) \cap Cn(B)) \subseteq Cn(A)$. Analogously we have that $Cn(Cn(A) \cap Cn(B)) \subseteq Cn(B)$. It follows that $Cn(Cn(A) \cap Cn(B)) \subseteq Cn(B) \cap Cn(A)$ $Cn(A) \cap Cn(B) \subseteq Cn(Cn(A) \cap Cn(B))$ follows directly by *inclusion*. □

Lemma 2.2 *Let $A, B, K \in 2^{\mathscr{L}}$. If A and B are equivalent then $A \subseteq Cn(K)$ iff $B \subseteq Cn(K)$.*

Proof By *monotonicity* and *idempotence* $Cn(A) \subseteq Cn(K)$. Since by hypothesis $Cn(A) = Cn(B)$ then $Cn(B) \subseteq Cn(K)$ and by *inclusion* $B \subseteq Cn(K)$.

The converse is analogous. □

A set $K \subseteq \mathscr{L}$ is *closed* (*under* Cn) iff $K = Cn(K)$. We reserve the uppercase letter K to represent closed sets.

The class of all closed sets in a logic $\langle \mathscr{L}, Cn \rangle$ will be denoted $\mathbb{K}_{\mathscr{L}}$ or simply $\mathbb{K}$ when the context is clear. In symbols:

$$\begin{tikzcd} \mathscr{L} \arrow[d] \\ a \arrow[d] \\ \emptyset \end{tikzcd}$$

Fig. 2.1 This diagram represents the logic of example 2.3

$$\mathbb{K} = \{K \in 2^{\mathscr{L}} : K = Cn(K)\}$$

Of course, the relation of inclusion is a partial order over $\mathbb{K}$ i.e., for every $K_1, K_2, K_3 \in \mathbb{K}$ we have:

transitivity: if $K_1 \subseteq K_2$ and $K_2 \subseteq K_3$ then $K_1 \subseteq K_3$.
reflexivity: $K_1 \subseteq K_1$.
anti-symmetry: if $K_1 \subseteq K_2$ and $K_2 \subseteq K_1$ then $K_1 = K_2$.

For every $\Gamma \subseteq \mathbb{K}$, an *upper-bound* of Γ is any $K \in \mathbb{K}$ such that $K' \subseteq K$ for every $K' \in \Gamma$. The *supremum* or the *least upper-bound* of Γ (denoted $sup(\Gamma)$) is an upper-bound of Γ such that $sup(\Gamma) \subseteq K'$ for every upper-bound $K' \in \mathbb{K}$ of Γ. Notice that for every $\Gamma \in \mathbb{K}$, there is a unique $sup(\Gamma) \in \mathbb{K}$ defined by the formula:

$$sup(\Gamma) = Cn(\bigcup \Gamma)$$

The *infimum*, the greatest lower bound, of Γ in $\mathbb{K}$ is defined analogously. Again notice that $inf(\Gamma) = Cn(\bigcap \Gamma) \in \mathbb{K}$. Hence, $\langle \mathbb{K}, \subseteq \rangle$ form a *complete lattice* i.e., a partially ordered set such that every subsets has a supremum and an infimum.

Furthermore, $\mathbb{K}$ is closed under intersection i.e., for every $\Gamma \subseteq \mathbb{K}$ we have $\bigcap \Gamma \in \mathbb{K}$. In other words, $\mathbb{K}$ forms a *closure system*. Hence, we have that:

$$inf(\Gamma) = \bigcap \Gamma$$

Logics will be illustrated by means of diagrams like the one in Example 2.1. Each node in the diagram represent an element K of $\mathbb{K}$ and will be labeled by one set A such $Cn(A) = K$. The transitive closure of the arrows represent the consequence relation i.e., $Cn(K_j) \subseteq Cn(K_i)$ iff there is a path in the diagram from K_i to K_j.

Example 2.3 defines a simple logic[1] which is represented in the diagram of Fig. 2.1.

Example 2.3

$$\begin{aligned} \mathscr{L} &= \{a, b\} \\ Cn(\mathscr{L}) = Cn(\{b\}) &= \mathscr{L} \\ Cn(\{a\}) &= \{a\} \\ Cn(\emptyset) &= \emptyset \end{aligned}$$

---

[1] This logic was borrowed from [Flo06].

Since we will present several examples of logics like the above, some conventions are useful. We will use $Cn(a)$ as an abbreviation for $Cn(\{a\})$. Furthermore, we will use a instead of $\{a\}$ to label the diagram. The value of $Cn(\mathscr{L})$ will be omitted in future presentation of logics, since it is always equal to $\mathscr{L}$ in Tarskian logics.

2.2 Consequence Relation

Another way to present Tarskian logics is via a consequence relation $R \subseteq 2^{\mathscr{L}} \times \mathscr{L}$. In this case a logic is a pair $\langle \mathscr{L}, R\rangle$ and $(A, \alpha) \in R$ means that α is a consequence of A. Tarskian consequence relation satisfies the following properties:

1. if $\alpha \in A$ then $(A, \alpha) \in R$,
2. if $(A, \alpha) \in R$ and $(B, \beta) \in R$ for every $\beta \in A$ then $(B, \alpha) \in R$ and
3. if $(A, \alpha) \in R$ and $A \subseteq B$ then $(B, \alpha) \in R$.

A Tarskian relation R induces the following consequence operator:

$$Cn(A) = \{\alpha \in \mathscr{L} : (A, \alpha) \in R\}$$

We call Cn the consequence operator that is induced by R. As we should expect, the consequence operator Cn induced by a Tarskian relation R is Tarskian:

Proposition 2.4 *If R is a Tarskian relation then the consequence operator Cn induced by R is Tarskian.*

Proof The proof is simple and follows like this:

Inclusion: If $\alpha \in A$ then $(A, \alpha) \in R$ and by definition $\alpha \in Cn(A)$.
Idempotence: Let $\alpha \in Cn(Cn(A))$. For every $\beta \in Cn(A)$, by definition, we have that $(A, \beta) \in R$. Since $(Cn(A), \alpha) \in R$ then $(A, \alpha) \in R$. Hence, $\alpha \in Cn(A)$.
Monotonicity: Let $A \subseteq B$ and $\alpha \in Cn(A)$. By definition $(A, \alpha) \in R$ which implies $(B, \alpha) \in R$. It follows that $\alpha \in Cn(B)$. □

Moreover, a Tarskian consequence operator Cn induces a relation:

$$(A, \alpha) \in R \text{ iff } \alpha \in Cn(A)$$

Proposition 2.5 *If Cn is a Tarskian operator then the consequence relation R induced by Cn is Tarskian.*

Proof This proof is omitted since it is trivial and very similar to the previous one. □

2.2.1 Properties of the Consequence Operator

In this section, we present a list of properties which are not as basic as the ones presented in previous section. These properties would not be generally assumed, but they will be useful throughout the book. The list of properties that will be considered include *compactness*, *finiteness*, *Descending Chain Condition (DCC)*, *closure under complement*, *distribuitivity* and *decomposability*.

Compactness guaranties that any consequence of a set of sentences is a consequence of a finite subset of it:

compactness: A logic $\langle \mathscr{L}, Cn \rangle$ is *compact* iff for all $\alpha \in \mathscr{L}$ and all $A \subseteq \mathscr{L}$, if $\alpha \in Cn(A)$ then there is a finite $A' \subseteq A$ such that $\alpha \in Cn(A')$.

A consequence relation is *compact* if for every $\alpha \in \mathscr{L}$ and $A \subseteq \mathscr{L}$ we have that $(A, \alpha) \in R$ there is a finite $A' \subseteq A$ such that $(A', \alpha) \in R$. It is trivial to verify that a compact relation induces a compact consequence operator and vice-versa.

Compact logics are sometimes called finitary. We would not name them finitary to avoid confusion with what we call finite logics:

finiteness: A logic $\langle \mathscr{L}, Cn \rangle$ is *finite* iff there is only a finite number of distinct belief sets i.e., $\mathbb{K}_{\mathscr{L}}$ is finite.

Of course, finite logics are compact, but the converse is not true in general.

Another logical properties related to compactness and finiteness are the chain conditions. A sequence of sets of sentences $A_0, A_1 \ldots$ is an *descending chain* iff $Cn(A_0) \supset Cn(A_1) \supset \ldots$. The Descending Chain Conditions (DCC) states that any chain in $\langle \mathscr{L}, Cn \rangle$ is finite.

descending chain condition: A logic $\langle \mathscr{L}, Cn \rangle$ satisfies the *descending chain condition* iff every descending chain in $\langle \mathscr{L}, Cn \rangle$ has a minimal element i.e., a A_j such that for every A_i, we have that $Cn(A_j) \subseteq Cn(A_i)$.

It is trivial to notice that finite logics satisfies DCC.

The *complement* of a set $A \subseteq \mathscr{L}$, if it exists, is a set $A' \subseteq \mathscr{L}$ such that:

- $Cn(A \cup A') = \mathscr{L}$
- $Cn(A) \cap Cn(A') = Cn(\emptyset)$

A set $A \subseteq \mathscr{L}$ is *finitely representable* iff there is finite A' such that $Cn(A) = Cn(A')$.

closure under complement: A logic $\langle \mathscr{L}, Cn \rangle$ is closed under complement or simply *complemented* iff every finitely representable $A \subseteq \mathscr{L}$ has a complement $A' \subseteq \mathscr{L}$.

Dropping the restriction to finitely representable would narrow too much the scope of complemented logics. If this restriction was dropped, even CPL would not be complemented.[2]

Notice that the complement of a set $A \subseteq \mathscr{L}$ may not be unique. In the logic of Example 2.6 and Fig. 2.2 the set $\{a\}$ has two distinct complements $\{b\}$ and $\{c\}$.

[2] This was noted by Flouris in personal communication.

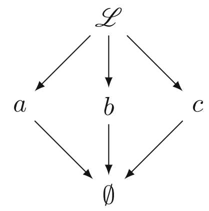

Fig. 2.2 Logic of Example 2.6

Example 2.6

$$
\begin{aligned}
\mathscr{L} &= \{a, b, c\} \\
Cn(\{a, b\}) = Cn(\{b, c\}) = Cn(\{a, c\}) &= \mathscr{L} \\
Cn(a) &= \{a\} \\
Cn(b) &= \{b\} \\
Cn(c) &= \{c\} \\
Cn(\emptyset) &= \emptyset
\end{aligned}
$$

The following property guaranties the uniqueness of complement:

distribuitivity: A logic $\langle \mathscr{L}, Cn \rangle$ is distributive iff for all $A, B, C \in 2^{\mathscr{L}}$ we have:

$$Cn(A \cup B) \cap Cn(A \cup C) \subseteq Cn(A \cup (Cn(B) \cap Cn(C)))$$

By monotonicity we have that the converse of this property holds. Hence, Tarskian distributive logics satisfy the following property for every set A and every finitely representable sets B and C :

$$Cn(A \cup (Cn(B) \cap Cn(C))) = Cn(A \cup B) \cap Cn(A \cup C)$$

For now on when we mention distributive logics we mean Tarskian distributive logics.

A logic is called *Boolean* iff it is distributive and complemented. In Boolean logics, every sentence has an unique complement modulo equivalences:

Proposition 2.7 *Let $\langle \mathscr{L}, Cn \rangle$ be a Boolean logic and consider $A \subseteq \mathscr{L}$. If A' and A'' are two complements of A then $Cn(A') = Cn(A'')$.*

Proof

$$
\begin{aligned}
Cn(A') &= Cn(A' \cup Cn(\emptyset)) \\
&= Cn(A' \cup (Cn(A) \cap Cn(A''))) \\
&= Cn(A' \cup A) \cap Cn(A' \cup A'') \qquad \text{by distribuitivity} \\
&= \mathscr{L} \cap Cn(A' \cup A'') \\
&= Cn(A' \cup A'')
\end{aligned}
$$

Using an analogous argument we prove that $Cn(A'') = Cn(A' \cup A'')$. Hence, $Cn(A') = Cn(A'')$. □

Furthermore, if two sets of sentences are equivalent in a Boolean logic then their complements are also equivalent:

Proposition 2.8 *Let $\langle \mathscr{L}, Cn \rangle$ be a Boolean logic and consider $A, B \in 2^{\mathscr{L}}$ and let A' be a complement of A and B' be a complement of B then $Cn(A) = Cn(B)$ if and only if $Cn(A') = Cn(B')$.*

Proof

$$\begin{aligned} Cn(A') &= Cn(A') \cup Cn(\emptyset) \\ &= Cn(A') \cup (Cn(B) \cap Cn(B')) \\ &= Cn(A') \cup (Cn(A) \cap Cn(B')) \\ &= Cn(A' \cup A) \cap Cn(A' \cup B') \qquad \text{by distribuitivity} \\ &= \mathscr{L} \cap Cn(A' \cup B') \\ &= Cn(A' \cup B') \end{aligned}$$

The proof that $Cn(B') = Cn(A' \cup B')$ is analogous and we conclude that $Cn(A') = Cn(B')$ □

In Boolean logics, the closure of the complement of A is denoted $\neg A$. $\neg A$ is well defined precisely because it is unique i.e., $\neg A = Cn(A')$ for A' a complement of A.

Let $A, K \in 2^{\mathscr{L}}$ such that $K = Cn(K)$ and $Cn(\emptyset) \subset A \subset K$, the *complement of A relative to K* (denoted $K^{-}(A)$) is the class of all sets K' such that:

- $Cn(K') \subset Cn(K)$
- $Cn(K' \cup A) = Cn(K)$

decomposability [FPA04]: A logic $\langle \mathscr{L}, Cn \rangle$ is *decomposable* iff for every $A, K \in 2^{\mathscr{L}}$ such that $K = Cn(K)$, $Cn(\emptyset) \subset Cn(A) \subset K$ and A is finitely representable we have that $K^{-}(A) \neq \emptyset$ i.e., there is $K' \subseteq \mathscr{L}$ such that $Cn(K') \subset Cn(K)$ and $Cn(K' \cup A) = Cn(K)$.

Decomposability was first introduced in [FPA04] and it is intimately related with AGM theory as will be showed in Chap. 4.

The following is a sufficient condition for a logic not to be decomposable:

Lemma 2.9 [FPA05] *Consider a logic $\langle \mathscr{L}, Cn \rangle$. If for some $K, K' \in 2^{\mathscr{L}}$ such that $K' = Cn(\{\beta \in \mathscr{L} : Cn(\beta) \subset K\})$ we have that $Cn(\emptyset) \subset K' \subset Cn(K)$ then $\langle \mathscr{L}, Cn \rangle$ is not decomposable.*

Proof For any $A \subseteq \mathscr{L}$, if $Cn(A) \subset Cn(K)$ then $Cn(A) \subseteq K'$. It follows that $Cn(K' \cup A) = Cn(K') \neq Cn(K)$. □

Now let us present some relations between decomposability and the other logical properties presented so far.

Proposition 2.10 *Boolean logics are decomposable.*

Proof Consider a Boolean logic $\langle \mathscr{L}, Cn\rangle$ and two sets of sentences $A, K \subseteq \mathscr{L}$ such that A is finitely representable and $Cn(\emptyset) \subset Cn(A) \subset Cn(K)$. We will prove that then $Cn(K) \cap Cn(A') \in K^-(A)$ for some complement A' of A (which must exist because $\langle \mathscr{L}, Cn\rangle$ is complemented).
Let $B = Cn(K) \cap Cn(A')$, then $Cn(A \cup B) = Cn(A \cup (Cn(K) \cap Cn(A')))$. By distributivity this is equal to $Cn(A \cup K) \cap Cn(A \cup A') = Cn(K)$.

If $A \subseteq B$ then $Cn(K) \cap Cn(A') = B = Cn(B) = Cn(B \cup A) = Cn(K)$, so $Cn(A') \supseteq Cn(K) \supset Cn(A)$. In this case, $Cn(A) = Cn(\emptyset)$ which is a contradiction. It follows that $A \nsubseteq B$. Hence, $B \in K^-(A)$. □

Proposition 2.11 *If a logic is decomposable and satisfy the descending chain condition then it is complemented.*

Proof Let $A \subseteq \mathscr{L}$. By decomposability we have that $\mathscr{L}^-(A) \neq \emptyset$. By the descending chain condition, there is an $X' \in \mathscr{L}^-(A)$ such that $X' \subseteq X$ and there is no X'' such that $Cn(X'') \subset Cn(X')$. We will show that X' is a complement of A.

Since $X' \in \mathscr{L}^-(A)$ then $Cn(X' \cup A) = \mathscr{L}$. Now suppose that $Cn(X') \cap Cn(A) \neq Cn(\emptyset)$, by decomposability, there is $Y \in X'^-(Cn(X') \cap Cn(A))$.

In this case, we have that $X' = Cn(Y \cup (Cn(A) \cap Cn(X'))) \subseteq Cn(Y \cup A)$. However, since $A \subseteq Cn(Y \cup A)$, we have that $Cn(X' \cup A) \subseteq Cn(Y \cup A)$. Hence, $Cn(Y \cup A) = \mathscr{L}$.
In this case, $Cn(Y) \subset Cn(X')$ and $Cn(Y \cup A) = \mathscr{L}$. It follows that X' is not the minimal which contradicts the definition. We conclude that $Cn(X') \cap Cn(A) = Cn(\emptyset)$. □

As a corollary of this result we have that finite decomposable logics are complemented.

2.3 Standard Languages

So far, no assumptions was made over the structure of the language $\mathscr{L}$. In this section, the standard language for propositional logics will be presented together with logical properties that depend on it.

A language $\mathscr{L}$ is *closed under an n-ary connective N* iff for every $\alpha_1, \ldots, \alpha_n \in \mathscr{L}$ we have that $N(\alpha_1, \ldots, \alpha_n) \in \mathscr{L}$. We will use standard infix notation for binary connectives i.e., we will write $\alpha N \beta$ instead of $N(\alpha, \beta)$.

A language is *standard* if it is closed under the standard connectives $\wedge$ (conjunction), $\vee$ (disjunction), $\rightarrow$ (implication) and $\neg$ (negation) (of course only the last connective is unary while the others are binary).

Consider a language $\mathscr{L}$ closed under negation. We say the negation $\neg$ in $\mathscr{L}$ is classical iff $\langle \mathscr{L}, Cn\rangle$ satisfies the following properties for every $\alpha \in \mathscr{L}$:

1. $Cn(\alpha) \cap Cn(\neg\alpha) = Cn(\emptyset)$
2. $Cn(\{\alpha, \neg\alpha\}) = \mathscr{L}$

If a set contains both $\alpha, \neg\alpha \in Cn(A)$ then we say that A is *contradictory*. Using this terminology we say that the second statement above guarantees that if $\langle \mathscr{L}, Cn\rangle$ is closed under classical negation then if A is contradictory it must be trivial. Of course, if a logic is closed under classical negation then it is complemented.

The following property of negation states that the negation of a sentence never "helps" to prove the sentence [Was00]:

α-local non contravention: A logic $\langle \mathscr{L}, Cn\rangle$ closed under negation satisfies *α-local non contravention* iff for every $\alpha \in \mathscr{L}$ and every $A \subseteq \mathscr{L}$ we have that if $\alpha \notin Cn(A)$ then $\alpha \notin Cn(A \cup \{\neg\alpha\})$

Proposition 2.12 *If $\langle \mathscr{L}, Cn\rangle$ is distributive and closed under classical negation then $\langle \mathscr{L}, Cn\rangle$ satisfies α-local non contravention.*

Proof Let $\alpha \in Cn(A \cup \{\neg\alpha\})$. Since $\alpha \in Cn(A \cup \{\neg\alpha\}) \cap Cn(A \cup \{\alpha\})$, by distributivity, $\alpha \in Cn(A \cup (Cn(\alpha) \cap Cn(\neg\alpha)))$. Since $\neg$ is classical, $Cn(\alpha) \cap Cn(\neg\alpha) = Cn(\emptyset)$. It follows that $\alpha \in Cn(A)$. □

In Chap. 4 classical AGM theory will be presented. This theory makes certain assumptions about the underlying logic $\langle \mathscr{L}, Cn\rangle$. First, it assumes that the language is a standard language i.e., it is closed under the standard connectives. The other two assumptions are presented below:

deduction: A logic $\langle \mathscr{L}, Cn\rangle$ closed under implication satisfies *deduction* iff for every $\alpha \in \mathscr{L}$ and every $A \subseteq \mathscr{L}$ we have that $\alpha \in Cn(A \cup \{\beta\})$ iff $\beta \to \alpha \in Cn(A)$.
supraclassicality: A logic $\langle \mathscr{L}, Cn\rangle$ is *supraclassical* iff for every $\alpha \in \mathscr{L}$ and every $A \subseteq \mathscr{L}$ we have that if $\alpha \in C_{CPL}(A)$ then $\alpha \in Cn(A)$ i.e., if α is a classical consequence of A then $\alpha \in Cn(A)$. Classical consequence will be formalized in the next section.

A consequence relation R satisfies deduction if the following holds:

$$(A \cup \{\beta\}, \alpha) \in R \quad \text{iff} \quad (A, \beta \to \alpha) \in R$$

If R satisfies deduction then the consequence operator induced by R also satisfies deduction and conversely.

We say that a logic satisfies the *AGM-assumptions* if it satisfies four requirements: closure under standard language, deduction, supraclassicality, and compactness. We will sometimes call *well behaved* a logic that satisfies AGM assumptions.

Well-behaved logics are Boolean.

Lemma 2.13 *If $\langle \mathscr{L}, Cn\rangle$ satisfy the AGM-assumption then $Cn(A \cup \{\alpha_1\}) \cap Cn(A \cup \{\alpha_2\}) \subseteq Cn(A \cup \{\alpha_1 \vee \alpha_2\})$.*

Proof By deduction we have that $\alpha_1 \to \beta, \alpha_2 \to \beta \in Cn(A)$. Since $(\alpha_1 \vee \alpha_2) \to \beta \in C_{CPL}(\{\alpha_1 \to \beta, \alpha_2 \to \beta\})$, by supraclassicality, monotonicity and idempotence $(\alpha_1 \vee \alpha_2) \to \beta \in Cn(A)$. Finally, by deduction, $\beta \in Cn(A \cup \{\alpha_1 \vee \alpha_2)$. □

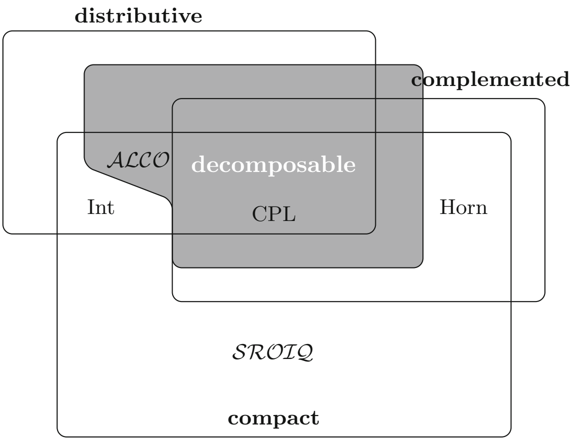

Fig. 2.3 Relation between logical properties

Proposition 2.14 *If a logic $\langle \mathcal{L}, Cn\rangle$ satisfies the AGM-assumptions then $\langle \mathcal{L}, Cn\rangle$ is Boolean.*

Proof Let $\beta \in Cn(A \cup B) \cap Cn(A \cup C)$. By compactness there are $A_1 \subseteq A \cup B$ and $A_2 \subseteq A \cup C$ both finite and such that $\beta \in Cn(A_1) \cap Cn(A_2)$. Let $A' = A_1 \cup A_2,\ B' = A_1 \cap A,\ C' = A_2 \cap C$. It is easy to verify that $A',\ B'$ and C' are finite and that $\beta \in Cn(A' \cup B') \cap Cn(A' \cup C')$. Let $B' = \{\beta_0, \ldots, \beta_n\},\ C' = \{\beta_{n+1}, \ldots \beta_{n+m}\},\ \gamma_1 = \beta_0 \wedge \ldots \wedge \beta_n$ and $\gamma_2 = \beta_{n+1} \wedge \ldots \wedge \beta_{n+m}$. By supraclassicality we have that $Cn(A' \cup B') \cap Cn(A' \cup C') = Cn(A' \cup \{\gamma_1\}) \cap Cn(A' \cup \{\gamma_2\})$. Using Lemma 2.13 $Cn(A' \cup \{\gamma_1\}) \cap Cn(A' \cup \{\gamma_2\}) \subseteq Cn(A' \cup \{\gamma_1 \vee \gamma_2\})$. It follows that:

$$\begin{aligned}
\beta \in\ & Cn(A' \cup B') \cap Cn(A' \cup C') \\
=\ & Cn(A' \cup \{\gamma_1\}) \cap Cn(A' \cup \{\gamma_2\}) \\
\subseteq\ & Cn(A' \cup \{\gamma_1 \vee \gamma_2\}) \\
\subseteq\ & Cn(A' \cup (Cn(B') \cap Cn(C'))) \\
\subseteq\ & Cn(A \cup (Cn(B) \cap Cn(C)))
\end{aligned}$$

We conclude that $\beta \in Cn(A \cup (Cn(B) \cap Cn(C)))$ and, hence, $Cn(A \cup B) \cap Cn(A \cup C) \subseteq Cn(A \cup (Cn(B) \cap Cn(C)))$.

Now consider a set $X \subseteq \mathcal{L}$. If X is finitely representable then there is a finite X' such that $Cn(X) = Cn(X')$. Let $X' = \{\beta_0, \ldots, \beta_n\}$. Since $\langle \mathcal{L}, Cn\rangle$ is supraclassical we have that $Cn(\beta_0, \ldots, \beta_n) = Cn(\beta_0 \wedge \ldots \wedge \beta_n)$. We will show that $\neg\alpha$ is the complement of $\alpha = \beta_0 \wedge \ldots \wedge \beta_n$.

Since $\beta \in C_{CPL}(\alpha, \neg\alpha)$ (see Sect. 3.1) for every $\beta \in \mathcal{L}$ we have $\beta \in Cn(\alpha, \neg\alpha)$, by supraclassicality. Furthermore, let $\beta \in Cn(\alpha) \cap Cn(\neg\alpha)$. By deduction it holds that $\alpha \to \beta, \neg\alpha \to \beta \in Cn(\emptyset)$. Since $\alpha \vee \neg\alpha \in C_{CPL}(\emptyset)$ (see Sect. 3.1), by supraclassicality, we have that $\beta \in Cn(\emptyset)$. Hence, $Cn(\alpha) \cap Cn(\neg\alpha) \subseteq Cn(\emptyset)$. □

A trivial corollary of this result states that if a logic $\langle \mathcal{L}, Cn\rangle$ satisfies the AGM-assumptions then it is decomposable.

2.4 Conclusion

In this chapter, a very general framework for logics was presented. A logic was defined as a language together with a consequence operator. In this book, we are interested only in Tarskian logics i.e., in logics with a consequence operator that satisfies monotonicity, inclusion, and idempotence. Not every logic in the literature is Tarskian, non-monotonic logics [AW97] and linear logics [Gir87] are examples of non-Tarskian logics. These logics would not be considered in this book.

One nice property of Tarskian logics is that there is a bijection between them and complete lattices. For this reason, it is possible to visually illustrate certain toy logics. We will use lattice many times throughout the book as an heuristic to generate examples of logics with certain properties. For a more complete presentation of consequence operator see [Wój88].

Tarskian logics may satisfy certain properties such as distributivity, decomposability etc. These properties are not independent. Some combinations of properties may imply other properties. Several such relations between properties were proved in this chapter. Diagram in Fig. 2.3 sums up the these results. The points in the diagram correspond to logics that will be presented in the following chapter.

References

[AW97] G. Antoniou and M.A. Williams. *Nonmonotonic reasoning*. Artificial intelligence. MIT Press, 1997.

[Flo06] Giorgos Flouris. *On Belief Change and Ontology Evolution*. PhD thesis, University of Crete, 2006.

[FPA04] Giorgos Flouris, Dimitris Plexousakis, and Grigoris Antoniou. Generalizing the AGM postulates: preliminary results and applications. In James P. Delgrande and Torsten Schaub, editors, *Proceedings of the 10th International Workshop on Non-Monotonic Reasoning 2004 (NMR-04)*, pages 171–179, Whistler BC, Canada, June 6-8 2004.

[FPA05] Giorgos Flouris, Dimitris Plexousakis, and Grigoris Antoniou. On applying the AGM theory to DLs and OWL. In Enrico Motta Yolanda Gil, V. Richard Benjamins, and Mark A. Musen, editors, *Proceedings of the 4th International Semantic Web Conference (ISWC 2005)*, pages 216–231, Galway, Ireland, November, 6-10 2005. Springer.

[Gir87] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[Was00] Renata Wassermann. *Resource Bounded Belief Revision*. PhD thesis, Universiteit van Amsterdan, Janeiro 2000.

[Wój88] R. Wójcicki. *Theory of logical calculi: basic theory of consequence operations*. Synthese library. Kluwer Academic Publishers, 1988.

http://www.springer.com/978-1-4471-4185-3

Belief Revision in Non-Classical Logics
Ribeiro, M.M.
2013, XI, 120 p. 5 illus., Softcover
ISBN: 978-1-4471-4185-3