1 Derivation Schemes in Twin Open Set Logic

Michael D. Westmoreland and Joan Krone

Several logic-like structures have been developed to analyze classical mechanical systems. These phase space logics are examples of a wider class of structures known as derived logics. Many of the derived logics for classical systems are non-Boolean so it is natural to ask about the existence of derivation schemes in these logics. The prime example of a derivation scheme is the one in classical logic given by entailment, implication, and modus ponens. This chapter presents the relation between these three aspects of classical logic and shows that these three exist only when the logic is Boolean. The chapter then considers some possible alternative derivation schemes for the particular derived logic known as twin open set logic. We show how twin open set logic describes a collision model of computation. We then consider this collision model to choose between alternatives for a derivation scheme for twin open set logic when that logic is applied to collision models.

1.1 Derived Logical Systems

The concept of derived logical systems was introduced in [16]. A logical system was said to be derived from a mathematical space and a class of distinguished substructures of the mathematical space. The substructures provide us with a family of propositions in the derived logic. A key point of [16] was to show that Birkhoff-von Neumann quantum logic is a derived logic. Thus, in that discussion the mathematical space represents a set of physical states and the distinguished substructures of the state space are identified by a theory of measurement on the space on physical states.

For example, in quantum logic the mathematical state space is a Hilbert space and von Neumann's theory of measurement of quantum states identifies the closed linear subspace of the Hilbert space as the propositions of quantum logic. This logic was first described by Birkhoff and von Neumann in [1].

Classical Boolean logic may be viewed as a derived logic. In classical logic the theory of measurement (which is rarely made explicit in classical mechanics) assumes that the state of a physical system can be determined with infinite precision; that is, the state of the system can be localized to a point in the system's phase space. We will consider another way of looking at classical Boolean logic as a derived logic in Sect. 1.3 below.
Twin open set phase space logic is suggested by the observation that there are two ways a measurement assigns a truth value to a proposition concerning a physical system. One of these ways is to use the measurement to verify the proposition. We say that a measurement $m$ verifies $P$ if $m \in \mathcal{P}_0$, where $\mathcal{P}_0$ is the canonical representative of $P$. The other way to use the measurement is to falsify the proposition. We say that a measurement $m$ falsifies $P$ if $m \in (\mathcal{P}_0)^c$, where $(\mathcal{P}_0)^c$ is the complement of the closure of the canonical representative of $P$. Note that the set $(\mathcal{P}_0)^c$ is an open set. These dual notions were used in [16] to describe the open phase space logic and its dual, the closed phase space logic. In twin open set phase space logic, these two notions are used simultaneously to assign truth values to a proposition.

As we show in the next section, twin open set logic is non-Boolean. We will show in Sect. 1.4 that a derivation scheme that is given by the lattice theoretic entailment, an implication $P \rightarrow Q$ that is equivalent to $\neg P \lor Q$ must be Boolean. Thus, the existence of possible analogous derivation schemes is an open question. We will present several alternative derivation schemes in Sect. 1.4.

Our goal is to consider these alternatives in the context of collision models of computation. It is well known that [13,4] that actual collision-based computers (e.g. billiard ball computers) are impractical. The causes of this impracticality should be noted:

1. collision computation is highly sensitive to initial conditions,
2. errors tend to propagate and accumulate,
3. the results of a computation (gate operation) must be determined by measurement.

Electronic computers in use today are extremely robust and so are insensitive to items 1 and 2 above. Hence in present-day computation one rarely worries about the problematic nature of measurement for models of computation. As computers become ever smaller and efficient, the quantum mechanical properties of computational components become relevant [12]. It is often argued that the most fundamental problem in quantum mechanics is the status of measurement [14,2,3]. The exact nature of the quantum mechanical measurement problem is well beyond the scope of this chapter.

Indeed, our purpose is to highlight a problematic feature of measurements performed on classical systems (and therefore those used by collision-based computers): measurement is inherently imprecise. One desire for derived logics is to have a logic or logics that are designed to reflect the various aspects of measurement. For example, twin open set logic is well-suited for analyzing the imprecision of measurement. One can then hope for some type of extension of twin open set logic (or some relative of it) that is suited for analyzing the non-commutativity of some quantum mechanical measurements [14] as well as the inherent imprecision of any measurement.

One possible application of twin open set logic to a collision-based computer is to provide a type of error detection. A key feature of twin open set
logic is that it is three-valued; it has an indeterminate truth value. We show in the truth tables in Sect. 1.3 below that the result of an operation is indeterminate if and only if one or more of the inputs is indeterminate. Hence, an output of indeterminate in the operation of a gate would indicate that the value of an input had drifted from the truth value it should represent. It is noted that some of the operations are robust to the presence of some indeterminacy; e.g., in the operation of disjunction if one of the inputs has the value “true” then the output will be “true”, regardless of the value of the other input.

The key consideration of this chapter is the possibility of derivation schemes in twin open set logic. By the term “derivation scheme” we mean a situation analogous to that found in classical logic where entailment, the implication operation, and \textit{modus ponens} allow one to say that from knowledge that $P$ and $P \rightarrow Q$ hold one to conclude that $Q$ holds as well. In this chapter, we reserve the term derivation scheme for such a three-part set-up. By the terms “derivation rule” or “derivation method”, we mean a rule such as \textit{modus ponens} or \textit{modus tollens}, or a rule analogous to them. An important fact presented here is that any DeMorgan algebra with a standard (in a sense to be made clear in Sect. 1.4) implication, entailment, and \textit{modus ponens} is a Boolean algebra. As twin open set logic is a non-Boolean DeMorgan algebra, any derivation scheme in it must have a nonstandard version of at least one of the three components. Our purpose here is to consider possibilities that involve modifying only one component and to choose one most amenable to the analysis of a billiard ball computer.

The chapter is structured as follows. Section 1.2 presents the basic properties of twin open set logic. Section 1.3 shows that Boolean logic can be considered as a special case of twin open set logic. Section 1.4 is the heart of the chapter showing that any logic with standard notions of implication, entailment and \textit{modus ponens} (i.e., a logic with a standard derivation scheme) must be Boolean. The remaining sections discuss alternative derivation schemes for twin open set logic. Section 1.5 shows that nontrivial tautologies are not possible in a general twin open set logic. Section 1.6 considers two alternatives for derivation schemes in twin open set logic in light of a collision model of computing. The chapter concludes with remarks given in Sect. 1.7.

### 1.2 Twin Open Set Logic

In order to consider simultaneously the verifiability and falsifiability of a proposition we will not consider single equivalence classes of subsets of the phase space; rather, we consider ordered pairs of certain equivalence classes of subsets of the phase space. That is, in twin open set phase space logic, we define propositions as follows:

**Definition 1.** We say that $P$ is a proposition in twin open set phase space logic if $P = ([V_0], [F_0])$, where $V_0$ and $F_0$ are disjoint open sets. Here, $[X]$
denotes the equivalence class of all sets that have the same interior as the set \( X \).

Given this notion of proposition, we now define the assignment of truth values as follows:

**Definition 2.** Let \( P = ([V_0],[F_0]) \) be a proposition in twin open set phase space logic and let \( m \) be a measurement. The proposition \( P \) will be assigned the truth value 1) **true** if \( m \subset V_0 \); 2) **false** if \( m \subset F_0 \); or 3) **indeterminate** otherwise.

We will refer to \([V_0]\) as the “verifiability class” of \( P \) and \([F_0]\) as the “falsifiability class” of \( P \). We will refer to \( V_0 \) as the “verifiability set of \( P \)” and to \( F_0 \) as the “falsifiability set of \( P \)”.

Note that the definitions of “true” and “false” are not identical to the notions of “verifiability” and “falsifiability” discussed above. If such were the case then propositions would always be of the form \( P = (\emptyset, \{\emptyset\}^c) \). That is, the falsifiability set would always be the interior of the complement of the closure of the verifiability set. Thus, propositions would be entirely characterized by their verifiability sets (or by their falsifiability sets) alone. The collection of propositions in the three-valued logic developed here is much larger; the only requirement for the canonical twin open sets is that they must be disjoint. This broader collection of propositions reflects that an experimenter’s resources for verifying a statement are often not identical to those resources that falsify the same statement.

Note also that a subset \( S \) is always an element of the verifiability class of some proposition (\( S \) is also the element of the falsifiability class of the negation of \( P \); a fact that will be explained later). We need only let \( V_0 = \text{int}(S) \) and then let \( F_0 \) be some open set disjoint with \( \text{int}(S) \). There will always be such an \( F_0 \); even in spaces with minimal separation properties (e.g. non-Hausdorff) we can at least take \( F_0 = \emptyset \). Thus, there will be no restriction to “proper propositions” as in the modified three-valued phase space logic.

We now define logical operators for any two propositions \( P \) and \( Q \). We will take

\[
P = ([P_0],[P_1])
\]
\[
Q = ([Q_0],[Q_1])
\]

where \( A_v \) is any representative of the verifiability class of the proposition \( A \) and \( A_f \) is any representative of the falsifiability class of the proposition \( A \).

**Definition 3.** The logical operators \( \neg \) (negation), \( \lor \) (disjunction), and \( \land \) (conjunction) are defined as follows:

\[
P \lor Q = ([\text{int}(P_0) \cup \text{int}(Q_0)], [\text{int}(P_1) \cap \text{int}(Q_1)])
\]
\[
P \land Q = ([\text{int}(P_0) \cap \text{int}(Q_0)], [\text{int}(P_1) \cup \text{int}(Q_1)])
\]
\[
\neg P = ([\text{int}(P_0)], [\text{int}(P_1)])
\]
We cast these definitions in terms of the open sets \( \text{int}(P_e), \text{int}(P_f) \), etc. because each equivalence class contains exactly one open set. This set can be taken as the canonical representative for the equivalence class. In each definition we apply set operations on the canonical representative verifiability and falsifiability sets for each proposition. The definitions are thus unambiguous. In the remainder of this presentation we shall take notational advantage of the fact that each equivalence class has one and only one open set. To this point we have followed the notation set down in [16 18,10] but hereafter we will denote propositions in twin open set logic as follows:

\[
P = [P_e, P_f];
\]
i.e., square brackets will denote that we are working with a proposition in twin open set logic. Any other grouping symbol (primarily parentheses) will denote a component of the ordered pair of open sets that is the proposition. Also, we will not explicitly work in the language of equivalence classes but will instead work with the disjoint open sets \( P_e \) and \( P_f \). Finally, any set of the form \((Y)_e\) denotes the verifiability set of the proposition \( Y \) and \((Y)_f\) denotes the falsifiability set. Hence, for example, \((P \land Q)_e\) denotes the verifiability set of the conjunction \( P \land Q \) (which is \( (P_e \cap Q_e) \)).

In terms of this more economical notation, Def. 3 is recast as follows:

**Definition 4.** The logical operators \( \neg \) (negation), \( \lor \) (disjunction), and \( \land \) (conjunction) are defined as follows:

\[
P \lor Q = [(P_e \cup Q_e), (P_f \cap Q_f)],
\]

\[
P \land Q = [(P_e \cap Q_e), (P_f \cup Q_f)],
\]

\[
\neg P = [P_f, P_e].
\]

In previous discussions of phase space logics [16,17], no attempt was made to define an implication operation. Indeed, as was pointed out in [17], the only derived logic for which a natural implication was known to exist was the logic referred to as the “logic derived from the unphysical theory of measurement”. (The “unphysical theory of measurement” is the one which allows for measurements of infinite precision. In this logic the lattice of propositions is the Boolean lattice of subsets of the phase space.) In contrast with the other logics derived from more realistic theories measurement, an implication operation will be defined for the twin open set phase space logic. The definition is as follows:

**Definition 5.** The logical operator \( \rightarrow \) (implication) is defined as follows:

\[
P \rightarrow Q = \neg P \lor Q = [(P_f \cup Q_e), (P_e \cap Q_f)]
\]
This operation also has a natural interpretation: a measurement $m$ will assign a value of “true” to the proposition $P \rightarrow Q$ if and only if $m$ assigns a value of “true” to either $\neg P$ or $Q$. That is, $m$ assigns a value of “true” to $P \rightarrow Q$ if and only if $m \subseteq (P_f \cup Q_c)$. Alternatively, $m$ will assign a value of “false” to the proposition $P \rightarrow Q$ if and only if $m$ assigns a value of “false” to both $\neg P$ and $Q$. That is, $m$ assigns a value of “false” to $P \rightarrow Q$ if and only if $m \subseteq (P_c \cap Q_f)$. $m$ will assign a value of “indeterminate” to the proposition $P \rightarrow Q$ if and only if $m \not\subseteq (P_f \cup Q_c)$ and $m \not\subseteq (P_c \cap Q_f)$.

We will now state some results about the operations in the twin open set phase space logic. The proofs can be found in [18].

**Theorem 1.** The operations $\vee$ and $\wedge$ are commutative and associative. $\vee$ distributes over $\wedge$ and vice versa.

**Proof.** In order to exhibit the character of the derivations, we will give only the proof of the distributivity of $\wedge$ over $\vee$. Other parts of the proof are equally straightforward.

Suppose $A$, $B$, and $C$ are propositions in the twin open set phase space logic. Then

\[
A \wedge (B \vee C) = [A_v, A_f] \wedge [(B_v \cup C_v), (B_f \cap C_f)] \\
= [(A_v \cap (B_v \cup C_v)), (A_f \cup (B_f \cap C_f))].
\]

The union of open sets is open, so $int(int(V) \cup int(W)) = (int(V) \cup int(W))$. Also, the interior of an intersection of sets is equal to the intersection of their interiors, so $int(int(V) \cap int(W)) = (int(V) \cap int(W))$. Thus

\[
A \wedge (B \vee C) = [(A_v \cap B_v) \cup (A_v \cap C_v), (A_f \cup B_f) \cap (A_f \cup C_f)] \\
= [(A_v \cap B_v), (A_f \cup B_f)] \vee [(A_v \cap C_v), (A_f \cup C_f)] \\
= (A \wedge B) \vee (A \wedge C).
\]

Therefore $\wedge$ distributes over $\vee$ in the twin open set phase space logic. The proof that $\vee$ distributes over $\wedge$ is the dual of the proof given: simply replace $\vee$ with $\wedge$, $\cap$ with $\cup$ and vice versa.

The next several theorems highlight the main difference between the twin open set phase space logic and the phase space logics that have been described previously: negation in the twin open set phase space logic has many of the properties of the negation in standard Boolean logic. In particular, negation is an involutive operation; i.e., $\neg \neg P = P$ for all propositions $P$. Also, versions of the Law of Non-contradiction and tertium non datur hold in twin open set phase space logic.

**Theorem 2.** In twin open set phase space logic, $\neg(\neg P) = P$ for all propositions $P$. 

Proof. Let \( P = [P_e, P_f] \) be a proposition in twin open set logic. By the
definition of negation we have
\[
\neg(\neg P) = \neg[P_f, P_e] \\
= [P_e, P_f] \\
= P.
\]

The following theorem can be thought of as a version of the Law of Non-
contradiction. In standard Boolean logic \( P \land \neg P \) is always false. In the twin
open set logic this proposition is never true; i.e. for different measurements
and propositions it may receive a value of indeterminate or a value of false.

**Theorem 3.** In the twin open set phase space logic, for any proposition \( P \),
\( P \land \neg P = [\emptyset, U] \). That is, \( P \land \neg P \) is a proposition which is never assigned
the value of true (\( U \) is some open subset of the topological space \( X \)).

Proof. Let \( P \) be any proposition in the twin open set phase space logic. We
then have
\[
P \land \neg P = [P_e, P_f] \land [P_f, P_e] \\
= [(P_e \cap P_f), (P_f \cup P_e)] \\
= [\emptyset, U],
\]
where \( U = P_f \cup P_e \), an open subset of \( X \).

By a dual argument, we also have a version of the Law of the Excluded
Middle (*tertium non datur*). As a middle truth value is allowed, we might
wish for a more appropriate name.

**Theorem 4.** In the twin open set phase space logic, for any proposition \( P \),
\( P \lor \neg P = ([U], [\emptyset]) \).

That is, \( P \lor \neg P \) is a never false proposition (again, \( U \) is an open subset
of the topological space \( X \).

DeMorgan’s laws also hold in twin open set phase space logic.

**Theorem 5.** For propositions \( P \) and \( Q \) in twin open set phase space logic,
\[
\neg(\neg P \land Q) = \neg P \lor \neg Q, \\
\neg(\neg P \lor Q) = \neg P \land \neg Q.
\]

Proof. Let \( P = [P_e, P_f] \) and \( Q = [Q_e, Q_f] \) be propositions in twin open set
phase space logic. We then have
\[
\neg(P \land Q) = \neg[(P_e \cap Q_e), (P_f \cup Q_f)] \\
= [(P_f \cup Q_f), (P_e \cap Q_e)] \\
= [P_f, P_e] \lor [Q_f, Q_e] \\
= \neg P \lor \neg Q.
\]

By a dual argument, we see that DeMorgan’s other law holds as well.
As Theorems 3 and 4 illustrate, there are many propositions that are never false and there are many propositions that are never true. Among such propositions, there are two worthy of special mention: The proposition \(1 = [X, \emptyset]\) is the always true, never false proposition. Dually, the proposition \(0 = [\emptyset, X]\) is the always false, never true proposition. In the following theorem, we see that 0 and 1 act as identity elements for \(\vee\) and \(\wedge\) respectively.

**Theorem 6.** *For each proposition \(P\) in the twin open set phase space logic we have*

\[
\begin{align*}
P \lor 0 &= P \\
P \land 1 &= P
\end{align*}
\]

**Proof.** Let \(P = [P_e, P_f]\) be any proposition, we have

\[
\begin{align*}
P \lor 0 &= [P_e, P_f] \lor [\emptyset, X] \\
&= [(P_e \cup \emptyset), (P_f \cap X)] \\
&= [P_e, P_f] \\
&= P.
\end{align*}
\]

The proof that \(P \land 1 = P\) is equally straightforward.

We see by the theorems presented above that the propositions in the twin open set logic, under the operations of \(\lor\) and \(\land\), satisfy many of the properties of a Boolean algebra. Recall [8] that an algebra \(\langle L, \oplus, \odot, 0, 1 \rangle\) is said to be Boolean if it satisfies the following axioms:

1. \(\oplus\) and \(\odot\) are associative binary operators,
2. \(\oplus\) and \(\odot\) are symmetric,
3. 0 and 1 are the identities of \(\oplus\) and \(\odot\) respectively,
4. the unary operator \(\neg\) satisfies \(b \oplus \neg b = 1\) and \(b \odot \neg b = 0\), and
5. \(\oplus\) distributes over \(\odot\) and vice-versa.

We see that (4) is the only Boolean algebra axiom that twin open set phase space logic fails. As this logic satisfies weakened forms of the Law of Non-contradiction and *tertium non datur*, it fails to be Boolean in general. It does satisfy the conditions for being a DeMorgan algebra. It should be noted that if \(X\) is given the trivial (or indiscrete) topology, then the derived twin open set phase space logic is Boolean. It is also interesting to note that the twin open set phase space logic derived from such a topology is effectively bivalent, in that no measurement will ever assign a value of indeterminate to a proposition.
1.3 Twin Open Set Logic and Classical Logic

As promised in Sect. 1.1, we again consider the status of classical Boolean logic as a derived logic. In previous discussions of this issue ([16, 18]) only the relationship between classical Boolean logic and other two-valued derived logics was considered. Here we wish to show that classical Boolean logic can be considered as a sublogic of a variety of twin open set logic; that is, there is sublattice of a twin open set logic that can be interpreted as classical Boolean logic.

The twin open set logic we wish to consider is the one derived from the unphysical measurement theory: the theory that allows for measurements of infinite precision. This implies that the topology for this twin open set logic is the discrete topology. For this reason (and in the interest of readability) we will refer to this variant as the twin discrete logic. In the discrete topology, every subset of the state space $X$ is an open set. Thus, in the full discrete twin open set logic propositions are ordered pairs of disjoint subsets of $X$.

We now restrict the set of propositions to ordered pairs of the form $[A, A']$ where $A$ is any subset of $X$ and $A'$ is the complement of $A$. For our Boolean sublogic we also require measurements to be singleton points of $X$ (in the discrete topology, this equivalent to requiring a measurement to be a connected open subset of the space $X$). Even though measurements in the full twin discrete logic may assign a value of indeterminate to a general proposition, no measurement in the twin discrete logic can assign a value of indeterminate to a proposition in the restricted class of propositions of the form $[A, A']$. This follows because measurements are points and the indeterminacy set for each of our restricted propositions is the empty set. Thus, any measurement in the twin discrete logic must assign a value of either “true” or “false” to any proposition in the restricted set.

The restricted set of propositions is isomorphic as a lattice to the power set of $X$ (denoted as $P(X)$); the mapping from the restricted set to $P(X)$ given by $[A, A'] \to A$ where $A$ is any subset of $X$. Hence we can think of discrete twin logic as a three-valued extension of classical Boolean two-valued logic.

One reason classical Boolean logic can be effectively described by collision models at least theoretically (indeed why it can be modeled by any physical system such as voltage, as a practical matter) is that classical Boolean logic is truth functional. Thus, a physical computation system need not be concerned with the physical particulars of why a proposition has a certain truth value, the system only needs to be able to model the truth values.

Another way of saying that a logic is truth functional is to say that truth tables are well-defined. That the truth tables for classical Boolean logic are well-defined is well known. The truth tables for the three-valued full discrete twin logic are also well-defined, which we now show. These truth tables correspond to Kleene three-valued logic [9].
The truth table for disjunction is:

<table>
<thead>
<tr>
<th>$P$</th>
<th>$Q$</th>
<th>$P \lor Q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td>$T$</td>
<td>$I$</td>
<td>$T$</td>
</tr>
<tr>
<td>$T$</td>
<td>$F$</td>
<td>$T$</td>
</tr>
<tr>
<td>$I$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td>$I$</td>
<td>$I$</td>
<td>$I$</td>
</tr>
<tr>
<td>$I$</td>
<td>$F$</td>
<td>$I$</td>
</tr>
<tr>
<td>$F$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td>$F$</td>
<td>$I$</td>
<td>$I$</td>
</tr>
<tr>
<td>$F$</td>
<td>$F$</td>
<td>$F$</td>
</tr>
</tbody>
</table>

Line five takes a truth value of “$T$” because the measurement $m$ (a point in $X$) falls in the indeterminacy set of $P$ and in the indeterminacy set of $Q$. Hence $m \notin P \cup Q$, and $m \notin P \cap Q$, so $m$ is in the indeterminacy set of $P \lor Q$. Line six takes a truth value of “$T$” because $m$ is the indeterminacy set of $P$ and the falsifiability set of $Q$. Hence $m \notin P \cup Q$, and $m \notin P \cap Q$, since $m \notin P_f$, so $m$ is in the indeterminacy set of $P \lor Q$. As $\lor$ is commutative in any twin open set logic, we see that line eight takes the value “$T$” as well.

The truth table for negation is:

<table>
<thead>
<tr>
<th>$P$</th>
<th>$\neg P$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>$F$</td>
</tr>
<tr>
<td>$I$</td>
<td>$I$</td>
</tr>
<tr>
<td>$F$</td>
<td>$T$</td>
</tr>
</tbody>
</table>

We emphasize that negation is well-defined (truth functional) for any twin open set logic. This feature is not dependent on the assumption that we are working in the discrete twin logic. It should be noted that this corresponds to the diametrical negation described by Reichenbach in [15].

The truth table for conjunction is:

<table>
<thead>
<tr>
<th>$P$</th>
<th>$Q$</th>
<th>$P \land Q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td>$T$</td>
<td>$I$</td>
<td>$I$</td>
</tr>
<tr>
<td>$T$</td>
<td>$F$</td>
<td>$F$</td>
</tr>
<tr>
<td>$I$</td>
<td>$T$</td>
<td>$I$</td>
</tr>
<tr>
<td>$I$</td>
<td>$I$</td>
<td>$I$</td>
</tr>
<tr>
<td>$I$</td>
<td>$F$</td>
<td>$F$</td>
</tr>
<tr>
<td>$F$</td>
<td>$T$</td>
<td>$F$</td>
</tr>
<tr>
<td>$F$</td>
<td>$I$</td>
<td>$F$</td>
</tr>
<tr>
<td>$F$</td>
<td>$F$</td>
<td>$F$</td>
</tr>
</tbody>
</table>
As we showed in Sect. 1.2 above, the DeMorgan's Laws hold in any twin open set logic. The values assigned to conjunction follow from the truth tables for disjunction, negation and DeMorgan's Laws.

Given the truth tables for negation and disjunction, we see that implication in discrete open logic is also truth functional. The truth table for implication is:

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>P → Q = ¬P ∨ Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>I</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>I</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>I</td>
<td>F</td>
<td>I</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>I</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

We note that even though the negation for twin open set logic (and so for the discrete twin logic considered here) was described by Reichenbach in [15], the standard implication described by him is not equivalent to that displayed above. The two implications differ in line five; where we assign a value of "T", Reichenbach assigns the value "T".

We have now established that the standard logical operators in discrete twin logic are truth functional. We have also seen that the discrete twin logic has a two valued Boolean sublogic. Recall that this results from the assumption that measurements in the discrete twin logic are infinitely precise. As pointed out above, the truth functionality of a logic makes possible, at least theoretically, its implementation by a physical computation model. The chief reason that collision models of computation are not practically possible is exactly because measurements are not infinitely precise. Hence we are led to ask how the truth tables change when the measurement theory reflects the inherent imprecision of measurement. This places us squarely in the context of standard twin open set logic where the topology is the standard topology on $\mathbb{R}^n$ where $n$ is the number of degrees of freedom in the physical model. In the rest of the chapter, we will refer to any such standard twin open set logic as simply "twin open set logic" unless explicitly noted.

As pointed out above, negation in any twin open set logic is truth functional and has the truth table as presented above. In contrast, the disjunction operation has the following truth table in twin open set logic:
$|P\lor Q|$

<table>
<thead>
<tr>
<th>$P$</th>
<th>$Q$</th>
<th>$P \lor Q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td>$T$</td>
<td>$I$</td>
<td>$T$</td>
</tr>
<tr>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td>$T$</td>
<td>$I$</td>
<td>$T$</td>
</tr>
<tr>
<td>$F$</td>
<td>$I$</td>
<td>$I$</td>
</tr>
<tr>
<td>$F$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td>$F$</td>
<td>$I$</td>
<td>$I$</td>
</tr>
<tr>
<td>$F$</td>
<td>$F$</td>
<td>$F$</td>
</tr>
</tbody>
</table>

We see that line six is unchanged as follows: Assume that a measurement $m$ assigns a truth value of “$I$” to $P$ and “$F$” to $Q$. It is clear that $m$ will not assign a value of “$T$” to $P \lor Q$. But $m$ cannot assign a value of “$F$” to $P \lor Q$ for otherwise we would have $m \in P_f \cap Q_f$ and so $m \in P_f$, contradicting the assumption that $m$ assigns “$I$” to $P$. Thus $m$ must assign a value of “$I$” to $P \lor Q$ in twin open set logic. It is now clear that line eight takes the value “$I$” as well.

To see that passing from discrete twin logic to twin open set logic destroys the truth functionality of disjunction, we must show why the truth table above is not well-defined in line five. We do this by citing two simple examples. Let us assume that our physical system has one degree of freedom; for example we are trying to locate a billiard ball within the length of an infinitely long tube. Thus our topological space $X$ is simply $\mathbb{R}$ in the standard topology. In this case, propositions are ordered pairs of disjoint open intervals and a measurement is simply an open interval. For both examples, let

$$P = (\{-2, 2\}, (5, 9])$$

$$Q = ([1, 3], (8, 11])$$

so

$$P \lor Q = ([{-1, 3}], (8, 9])$$.

For the first example, let $m = (0, 2.5)$. This measurement assigns a truth value of “$T$” to each of $P$ and $Q$ but it assigns a truth value of “$T$” to $P \lor Q$ since $m \subseteq P_f \cup Q_f$. For the second example, let $m = (0, 4)$. This measurement assigns a truth value of “$T$” to $P \lor Q$ as well as to $P$ and to $Q$ since $m \not\subseteq P_f \cup Q_e$. Thus we see that $\lor$ is not well-defined in twin open set logic.

By the duality we have that the truth table for conjunction in twin open set logic is the following:
\[
\begin{array}{c|c|c}
P & Q & P \land Q \\
\hline
T & T & T \\
T & I & I \\
T & F & F \\
I & T & I \\
I & I & I or I \\
I & F & F \\
F & T & F \\
F & I & F \\
F & F & F \\
\end{array}
\]

The examples given above to demonstrate the lack of truth functionality for disjunction also serve that purpose for conjunction if we replace the propositions \(P\) and \(Q\) with their respective negations.

The truth table for the standard implication in twin open set logic, defined in Sect. 1.2 is:

\[
\begin{array}{c|c|c}
P & Q & P \rightarrow Q \\
\hline
T & T & T \\
T & I & I \\
T & F & F \\
I & T & T \\
I & I & I or I \\
I & F & I \\
F & T & T \\
F & I & T \\
F & F & T \\
\end{array}
\]

It is the ambiguity in line five of this truth table that motivates this chapter. The lack of truth functionality for implication suggests that another operation that is truth functional should be used as a replacement so that the derivation scheme of \textit{modus ponens} and entailment are preserved. One might hope to find an implication that is \textit{given by the lattice structure of the propositions} (this notion will be made precise in the next section) that also respects \textit{modus ponens} and entailment. A key result of the next section (Theorem 7) shows that this can only be realized by a Boolean algebra. The remainder of Sect. 1.4 looks at various alternative derivation schemes for twin open set logic.

### 1.4 Derivation Schemes in Twin Open Set Logic

One of the problematic features of von Neumann-Birkhoff quantum logic is its lack of a natural implication operator; see [5] for an informative discussion
of this issue. In this section we will discuss implication type operators in twin
open set phase space logic.

By the term “implication type operator” we mean a binary operation
that relates to derivations in a way analogous to the relation between classical
implication and modus ponens. It is noted in [5] that any implication operator
must be directly related to the relation of semantic implication. The direct
relation posited by Hardegree [5] is: if $P$ semantically entails $Q$ then $P \rightarrow Q$ is
valid.

If the set of propositions forms a lattice, then we can phrase the relation
between entailment and implication in lattice theoretic terms: If $P \leq Q$ then
$P \rightarrow Q$ is valid, where “$\leq$” is the lattice ordering. It is clear that, in classical
logic, semantic entailment does provide an ordering on the set of propositions.

Thus we must address at least three issues:

1. What is a proper ordering for the propositions in twin open set logic?
2. What is a proper implication operator in twin open set logic?
3. What derivation method can be implemented given the answers to 1) and
   2)?

In addressing these issues, we should be aware of the following:

**Theorem 7.** Let $(A, \land, \lor, \neg)$ be a DeMorgan algebra. If we define an ordering “$\preceq$” on the algebra by

$$P \preceq Q \Leftrightarrow P \lor Q = Q,$$

then $P \land (\neg P \lor Q) \preceq Q$ if and only if $(A, \land, \lor, \neg)$ is a Boolean algebra.

**Remark 1.** This theorem implies that any DeMorgan algebra in which

1. entailment is given by $\land$ (or $\lor$),
2. the implication operator is given by $\neg P \lor Q$, and
3. that satisfies the rule of modus ponens,

must be a Boolean algebra.

**Proof.** We will only need to show that $(A, \land, \lor, \neg)$ satisfies the Law on Non-contradiction, as any DeMorgan algebra that satisfies that rule is, in fact, a
Boolean algebra. In any DeMorgan algebra, we have that $0 \leq P \land \neg P$. If we
substitute $Q = 0$ in the generalized modus ponens scheme in the hypothesis,
we have $P \land (\neg P \lor 0) \leq 0$. As distributivity holds in a DeMorgan algebra,
this gives

$$(P \land \neg P) \lor (P \land 0) \leq 0.$$

As $P \land 0 = 0$ and $Q \lor 0 = Q$, we have that $P \land \neg P \leq 0$. As $\leq$ is antisymmetric
we have $P \land \neg P = 0$ and so $(A, \land, \lor, \neg)$ is Boolean.
This proposition implies that if we wish to maintain the connection between some lattice theoretic entailment and the implication operation in a non-Boolean logic, then at least one of the three conditions in the intuitive description above must fail. In each of the following subsections we will modify each of these conditions from its analogue to classical logic. We will refer to the modified construct as “nonstandard”.

1.4.1 Nonstandard Derivation Methods (or what to do when you can’t do modus ponens)

The ordering that is given by the operation “∧” in twin open set logic is the following:

\[ P \leq Q \iff P_v \subseteq Q_v \text{ and } Q_f \subseteq P_f. \]

We say that this is the ordering given by “∧” since

\[ P \wedge Q = P \iff P_v \subseteq Q_v \text{ and } Q_f \subseteq P_f. \]

(We could also say that this ordering is given by “∨” since

\[ P \vee Q = Q \iff P_v \subseteq Q_v \text{ and } Q_f \subseteq P_f. \]

Intuitively we can think of this as the ordering “Q is more readily verified and less easily falsified than P”.

In Sect. 1.2 we defined implication in twin open set logic in a manner analogous to one of the characterizations of classical implication. That is, we use the following definition:

\[ P \rightarrow Q \equiv_{def} \neg P \vee Q. \]

Thus, in terms of the canonical verifiability and falsifiability sets for P and Q we have

\[ (P \rightarrow Q) = (\neg P \vee Q) \]

\[ = [(P_f \cup Q_f), (P_v \cap Q_f)]. \]

From Theorem 7 we know that the lattice theoretic modus ponens will not hold for this implication. It is instructive to see why it fails.

**Proposition 1.** In general, with the ordering given by “∧”, it is not the case that \( P \wedge (P \rightarrow Q) \leq Q \).

**Proof.**

\[
P \wedge (P \rightarrow Q) = P \wedge (\neg P \vee Q) \\
= (P \wedge \neg P) \vee (P \wedge Q) \\
= [[P_v \cap P_f], (P_v \cup P_f)] \lor [(P_v \cap Q_v), (P_f \cup Q_f)] \\
= [\emptyset, (P_v \cup P_f)] \lor [(Q_v \cap P_v), (Q_f \cup P_f)] \\
= [(Q_v \cap P_v), ((P_v \cup P_f) \cap (Q_f \cup P_f))] \\
= [(Q_v \cap P_v), ((P_v \cap Q_f) \cup P_f)].
\]
In order for $P \land (P \rightarrow Q) \subseteq Q$ to hold in general, then $Q_f \subseteq (P_e \cap Q_f) \cup P_f$ would have to be true in general. In any situation where $P_e \cup P_f \neq X$ this set containment would fail. Thus, in any nondiscrete topology we will have disjoint open sets $P_f$ and $P_e$ such that $P_e \cup P_f \neq X$ so the claim is established.

Any derivation method in twin open set logic that is implemented in close analogy to classical entailment and implication must be weaker than *modus ponens*. As the proof of Prop. 1 shows, we must find a proposition that is guaranteed to contain $[(Q_e \cap P_e), ((P_e \cap Q_f) \cup P_f)]$ in general. Furthermore, we see that the verifiability set of this proposition is not the problem, it is always contained in $Q_e$ so the problem is reduced to asking: What open set or open sets can will be contained in $((P_e \cap Q_f) \cup P_f)$ and still be disjoint from $Q_e \cap P_e$? The proposition $[(Q_e \cap P_e), []]$ satisfies these conditions.

What does this derivation method mean? The method would operate as follows: If $P \rightarrow Q$ and $P$ are each true then the proposition $[Q_e, []]$ is true. Thus, if a measurement $m$ verifies $P \rightarrow Q$ and $P$ then it must be the case that $m$ lies entirely in the set $Q_e$, so that the proposition $[Q_e, []]$ is verified. But so is any proposition of the form $(Q_e, Y)$ where $Y$ is any open subset of $\text{int}(Q_e)$. Thus, other choices for the derived proposition are possible in addition to the just described $[Q_e, []]$.

In the quest for a reasonable proposition to derive from the premises $P$ and $P \rightarrow Q$ we consider the derivation method *modus tollens*. This is the derivation method that allows us to derive $\neg P$ from the premises $\neg Q$ and $P \rightarrow Q$. In classical propositional logic *modus tollens* is simply a special case of *modus ponens*. This follows from the equivalence of an implication $P \rightarrow Q$ to its contrapositive: $\neg Q \rightarrow \neg P$. This equivalence also holds in twin open set logic:

\[
P \rightarrow Q \equiv_{def} \neg P \lor Q
\]

\[
= \neg(\neg Q) \lor \neg P
\]

\[
=_{def} \neg Q \rightarrow \neg P.
\]

In the present setting we cannot claim that *modus tollens* is a special case of *modus ponens*; after all, since we are assuming a standard implication and entailment we cannot have *modus ponens*.

Nonetheless we shall consider the proposition $\neg Q \land (P \rightarrow Q)$; i.e., the premises for *modus tollens*:

\[
\neg Q \land (P \rightarrow Q) = \neg Q \land (\neg P \lor Q)
\]

\[
= (\neg Q \land \neg P) \lor (\neg Q \land Q)
\]

\[
= [(Q_f \cap P_f), (Q_e \cup P_e)] \lor [(Q_f \cap Q_e), (Q_e \cup Q_f)]
\]

\[
= [(Q_f \cap P_f), (Q_e \cup P_e)] \lor [(Q_f \cap Q_e), (Q_e \cup Q_f)]
\]

\[
= [(Q_f \cap P_f), (Q_e \cup (P_e \cap Q_f))]
\]

\[
= [(Q_f \cap P_f), (Q_e \cup (P_e \cap Q_f))].
\]
We see that our previous derivation rule of "\( P \) and \( P \rightarrow Q \) yields \( [Q_e, \emptyset] \)" has an analogue suggested by *modus tollens* of "\( \neg Q \) and \( P \rightarrow Q \) yields \( [P_f, \emptyset] \)". But another derived proposition is suggested by the two derivations: *modus ponens* is replaced by "the premises \( P \) and \( P \rightarrow Q \) yield \( [Q_e, P_e \cap Q_f] \simdef Q_P \)" and *modus tollens* is replaced by "the premises \( \neg Q \) and \( P \rightarrow Q \) yield \( [P_f, P_e \cap Q_f] \simdef P_Q \)". For at least some applications this latter derivation rule is more desirable than the rule yielding propositions of the form \( [X, \emptyset] \). For example, a string of applications of our derivations could represent a potentially falsifiable line of reasoning. In the first alternative, there is no way to falsify the results of an application of *modus ponens*. But if the falsifiability set is \( P_e \cap Q_f \) then there are measurements that will yield the derived proposition as false. It is interesting to note that the set \( P_e \cap Q_f \) corresponds to the falsifiability set for \( P \rightarrow Q \) in the two-valued logic that is the Boolean restriction of the discrete twin logic described in Sect. 1.3 above.

Thus, we have a candidate to replace *modus ponens* that will respect the entailment given by \( \land \) and the standard implication. That is, we have

\[
P \land (P \rightarrow Q) \subseteq Q_P
\]

since (from the last line of the proof of Prop. 1 above)

\[
Q_e \cap P_e \subseteq Q_e
\]

and

\[
Q_f \cap P_e \subseteq (Q_f \cap P_e) \cup P_f.
\]

### 1.4.2 Derivation Schemes under Nonstandard Entailment

We now consider possible alternatives to the entailment ordering given by the conjunction operator \( \land \) in twin open set logic. Several alternative relations on the set of propositions are possible. We first consider the relation \( P \rho Q \) given by

\[
P \rho Q \equivdef P_e \subseteq Q_e.
\]

Note that this relation is not a partial order as it is not antisymmetric. This is seen by noting that any two propositions \( P \) and \( Q \) such that \( P_e = Q_e \) will satisfy the condition that \( P \rho Q \) and \( Q \rho P \) without it necessarily following that \( P = Q \). This relation \( \rho \) is reflexive and transitive; recall that relations that are reflexive and transitive are called *quasi-ordering relations* [7].

We see if we replace entailment by \( \rho \) but maintain the standard implication then *modus ponens* holds relative to \( \rho \); i.e., the following holds:

**Proposition 2.** If "\( \rho \)" is defined as above then \( P \land (P \rightarrow Q) \rho Q \) in twin open set logic.
Proof.

\[ P \land (P \to Q) = P \land (\neg P \lor Q) \]
\[ = (P \land \neg P) \lor (P \land Q) \]
\[ = [(P_v \land P_f), (P_v \lor P_f)] \lor [(P_v \land Q_v), (P_f \lor Q_f)] \]
\[ = [\emptyset, (P_v \cup P_f)] \lor [(Q_v \cap P_v), (Q_f \cup P_f)] \]
\[ = ([Q_v \cap P_v], ((P_v \cup P_f) \cap (Q_f \cup P_f))] \]
\[ = [(Q_v \cap P_v), ((P_v \cap Q_f) \cup P_f)]. \]

Since \( Q_v \cap P_v \subseteq Q_v \), the result holds.

One can think of this “entailment” as given by the verifiability properties of the propositions. It is interesting that the corresponding “entailment” given by the quasi-ordering \( \eta \) derived from

\[ P_\eta Q \equiv_{def} Q_f \subseteq P_f, \]

does not provide a valid *modus ponens*. Indeed, it is precisely because

\[ Q_f \nsubseteq ((P_v \cap Q_f) \cup P_f) \]

in general that the entailment given by \( \land \) does not give a valid *modus ponens*.

The use of the relation \( \rho \) is unsatisfying since the falsifiability of propositions plays no role in this entailment. Thus we are led to ask: Are there any relations on the set of propositions in twin open set logic that involve the falsifiability properties of the propositions?

There is such a quasi-ordering given by:

\[ P \leq Q \equiv_{def} \begin{cases} 
P_v \subseteq Q_v, \text{ and} \\
Q_f \cap P_f = Q_f \cap P_f. 
\end{cases} \]

Recall that \( P_S = P_v \cup P_f \). An alternative phrasing of this is

\[ P \leq Q \equiv_{def} \begin{cases} 
P_v \subseteq Q_v, \text{ and} \\
Q_f \cap P_f = \emptyset. 
\end{cases} \]

That this relation is reflexive is straightforward. The transitivity of \( \leq \) defined here is demonstrated as follows: Assume that \( P \leq Q \) and \( Q \leq R \) then we have that \( P_v \subseteq R_v \) by the transitivity of set containment. We also have that \( R_f \cap Q_v = \emptyset \) and since \( P_v \subseteq Q_v \) it follows that \( R_f \cap R_v = \emptyset \) so that \( P \leq R \).

The quasi-ordering \( \leq \) also gives a valid *modus ponens*, i.e., we have the following:
Proposition 3. If \( \leq \) is defined as above then \( P \land (P \rightarrow Q) \leq Q \) in twin open set logic.

Proof. From the proof of Prop. 2 above we have

\[
P \land (P \rightarrow Q) = (Q_e \land P_e, (P_e \land Q_f) \cup P_f).
\]

Again it is clear that \( (Q_e \land P_e) \subseteq Q \), but we need to check that the second condition for the relation \( \leq \) is satisfied. The falsifiability set for \( P \land (P \rightarrow Q) \) is \((P_e \land Q_f) \cup P_f\). We see that

\[
((P_e \land Q_f) \cup P_f) \cap Q_f = (P_e \land Q_f) \cap (P_f \land Q_f)
= (P_e \land Q_f) \cup (P_f \land Q_f)
= (P_e \cup P_f) \land Q_f
= P_s \land Q_f.
\]

The set \( (P \land (P \rightarrow Q)) \) is

\[
(Q_e \land P_e) \cup ((P_e \land Q_f) \cup P_f) = (Q_e \land P_e) \cup ((P_e \cup P_f) \land (Q_f \cup P_f))
= (Q_e \land P_e) \cup (P_s \land (Q_f \cup P_f))
= (Q_e \land P_e) \cup ((Q_s \land Q_f) \cup (Q_f \cup P_f))
= (Q_e \land P_e) \cup ((Q_s \land Q_f) \cup (Q_s \land P_f))
= (Q_e \land P_e) \cup ((Q_s \land Q_f) \cup (Q_s \land P_f))
= (Q_e \land P_e) \cup ((Q_s \land Q_f) \cup (Q_s \land P_f))
= (Q_e \land P_e) \cup (Q_s \land (Q_f \cup P_f))
= (Q_e \land P_e) \cup (Q_s \land (Q_f \cup P_f))
= (Q_e \land P_e) \cup (Q_s \land (Q_f \cup P_f))
= (Q_e \land P_e) \cup (P_s \land (Q_f \cup P_f)).
\]

This still unwieldy last line allows us to conclude that

\[
(P \land (P \rightarrow Q)) \land Q_f = ((Q_e \land P_f) \cup ((P_e \land Q_f) \cup P_f)) \land Q_f
= (Q_e \land P_f) \land Q_f \cup ((P_e \land Q_f) \cup P_f) \land Q_f
= (P_e \land Q_f) \cup (P_f \land Q_f)
= (P_e \land Q_f) \cup (P_f \land Q_f)
= (P_e \land Q_f) \cup (P_f \land Q_f)
= P_s \land Q_f.
\]

Hence, we conclude that

\[
(P \land (P \rightarrow Q)) \land Q_f = (P \land (P \rightarrow Q)) \land Q_f
\]

thus establishing that \( (P \land (P \rightarrow Q)) \leq Q \). That is, we have that \textit{modus ponens} is valid in the quasi-ordering \( \leq \).
1.4.3 Nonstandard Implication

If we maintain the ordering given by $\land$ and the close analogue with *modus ponens* then we must consider other characterizations of implication other than $\neg P \lor Q$. As noted in [5], $\text{sup}(X | P \land X \leq Q)$ is a well-defined operation in any orthomodular lattice. Recall that an algebra $(L, \land, \lor)$ is a lattice if and only if $L$ is a nonempty set, $\land$ and $\lor$ are binary operations on $L$, both $\land$ and $\lor$ are idempotent, associative, commutative, and they satisfy the following two absorption identities [7]:

\[
\begin{align*}
P \lor (P \land Q) &= P \\
P \land (P \lor Q) &= P.
\end{align*}
\]

In the context of twin open set logic, $L$ is the set of ordered pairs $[A, B]$ where $A$ and $B$ are open subsets of a topological space $X$ such that $A \cap B = \emptyset$. In Sect. 1.2 above we showed that $\land$ and $\lor$ are associative and commutative; it is clear that they are idempotent as well. We verify the first absorption identity:

\[
\begin{align*}
P \lor (P \land Q) &= [P_e, P_f] \lor ([P_e, P_f] \land [Q_e, Q_f]) \\
&= [P_e, P_f] \lor ([P_e \land Q_e], (P_f \lor Q_f)] \\
&= [(P_e \lor (P_e \land Q_e)), (P_f \land (P_f \lor Q_f))] \\
&= [P_e, P_f] \\
&= P.
\end{align*}
\]

The justification for the step from line three to line four is that the set operations $\cap$ and $\lor$ satisfy the absorption identities. The proof that twin open set logic satisfies the second absorption identity is equally straightforward.

Thus, the set of propositions in twin open set logic forms a lattice. As noted in Sect. 1.4 above, twin open set logic is *not* an orthomodular lattice. The desired supremum is guaranteed to exist only in orthomodular lattices. We now show that the desired supremum does not exist in general twin open set logics. In terms of the components of $\text{sup}(X | P \land X \leq Q)$ we want to characterize the proposition $X = [X_e, X_f]$ so that $X_e = \text{sup}(Y | P_e \cap Y \leq Q_e)$ and $X_f = \text{inf}(Y | Q_f \subseteq P_f \cup Y)$. We recall in the derivation of Prop. 1 above that the verifiability set of $P \land (\neg P \lor Q)$ did not block the desired entailment. That is, $(P_e \cap (P_f \lor Q_f)) \subseteq Q_e$ for any two propositions $P$ and $Q$. Hence for the desired supremum under $\leq$ we require that $P_f \lor Q_f \subseteq X_e$. Once this requirement is imposed, the existence of $X_f = \text{inf}(Y | Q_f \subseteq P_f \lor Y)$ is blocked in twin open set logic, since we also require $X_e \cap X_f = \emptyset$. The natural open set disjoint from $X_e$ to consider is $X_f = (Q_f \setminus \overline{P_f})$, where $\overline{A}$ means the closure of $A$. We must use the closure of $P_f$ to ensure that $X_f$ is an open set. Thus, we consider replacing implication operator $P \Rightarrow Q \triangleq \neg P \lor Q$ with the operation $P \Rightarrow Q \triangleq [P_f \lor Q_e, Q_f \setminus \overline{P_f}]$. We now show that this does not suffice in preserving *modus ponens* and the entailment given by $\land$. 


Proposition 4. With "\( \rightarrow \)" defined as above we have

\[ P \land (P \rightarrow Q) \not\models Q. \]

Proof.

\[ P \land (P \rightarrow Q) =_{def} [P_e, P_f] \land [P_f \cup Q_e, Q_f \setminus P_f] \]
\[ = \quad [P_e \cap (P_f \cup Q_e), P_f \cup (Q_f \setminus P_f)] \]
\[ = \quad [\emptyset \cup (P_e \cap Q_e), P_f \cup (Q_f \setminus P_f)] \]
\[ = \quad [(P_e \cap Q_e), (Q_f) \cap (P_f \setminus P_f)] \]
\[ \not\models \quad [Q_e, Q_f] = Q. \square \]

We see that if \( bd(P_f) \neq \emptyset \) there is no possibility that the desired entailment will hold.

One might assert that the requirement that \( P_f \cup Q_e \subseteq X_e \) is too stringent; that is, it may be possible to find a substitute implication operator that preserves \textit{modus ponens} and the standard entailment. While we cannot, at this point, rule out such a possibility, it should be pointed out that this choice for \( X_e \) is desirable on at least two grounds. First, in the limiting case given by the Boolean restriction to the discrete twin logic (i.e., classical Boolean logic) this is the form of the implication operator. Second, this is the most natural interpretation of implication when we know that a measurement verifies \( P \) and \( P \rightarrow Q \), whatever form the replacement implication operator, \( \rightarrow \) may take.

1.5 Tautologies in Twin Open Set Logics

From Sect. 1.4 we see that we have several alternatives for derivation schemes in twin open set logics. We also know from Theorem 7 in Sect. 1.4 that we cannot import the standard implication scheme from classical logic; i.e., we cannot have the standard versions of entailment, implication and \textit{modus ponens} since the algebra of propositions in twin open set logic is not Boolean.

What of other possibilities? For example, perhaps by making minor modifications to entailment and implication together, \textit{modus ponens} would continue to hold with the overall scheme being in some (as yet undescribed) sense more appealing that any of the those listed here.

For example one might note that in classical Boolean logic the utility of \textit{modus ponens} when applied to implication is closely tied to the tautological nature of the proposition \((P \land (P \rightarrow Q)) \rightarrow Q\). One might wonder if there could be some derivation scheme that would maintain an analogous tautology in twin open set logic. We address this issue in the present section.
The proposition \( (P \land (P \rightarrow Q)) \rightarrow Q \) is not a tautology in twin open set logic, as can be seen by the following derivation (here we let \( X_S \) denote a set of the form \( X_v \cup X_f \); i.e., it is the union of \( X \)'s verifiability and falsifiability sets):

\[
(P \land (P \rightarrow Q)) \rightarrow Q = \\
= \neg((P \land \neg P) \lor (P \lor Q)) \lor Q \\
= \neg((P \land \neg P) \lor (P \lor Q)) \lor Q \\
= ((\neg P \lor P) \land (\neg P \land \neg Q)) \lor Q \\
= ((\neg P \lor P) \lor (\neg P \land \neg Q)) \land Q \\
= [(P_S \cup Q_e), \emptyset \in (P_f \cup Q_s), (P_e \cap \emptyset)] \\
= [(P_S \cap Q_e) \cup (P_f \cap Q_s), (\emptyset)] \\
= [(P_f \cap Q_e) \cup ((P_S \cap Q_s) \cup (Q_e \cap Q_s)), (\emptyset)] \\
= [(P_f \cup (Q_e \cap P_f)) \in ((P_S \cup Q_s) \cup (Q_e \cup Q_s)), (\emptyset)] \\
= [(P_f \cup Q_e) \cup ((P_S \cap Q_s) \cup (Q_e \cap Q_s)), (\emptyset)] \\
= [(P_f \cup Q_e) \cup ((P_S \cap Q_s) \cup Q_e), (\emptyset)] \\
= [(P_f \cup (P_S \cap Q_s)) \cup Q_e), (\emptyset)]
\]

Note that this is the always true proposition \( (T = [X], (\emptyset)) \) only if \( P_f = X \), or \( Q_e = X \), or \( Q_s = X = P_s \). Thus, in any topology with three pairwise disjoint nonempty open sets, this sentential version of *modus ponens* is not a tautology.

Is there any derivation scheme that will yield a tautological sentential version in twin open set logic? The answer to this question is: no. This is because there are no nontrivial tautologies in any twin open set logic derived from a topology with three pairwise disjoint nonempty open sets. The term "nontrivial" used here deserves some attention.

In classical logic all tautologies are logically equivalent. In the setting of derived logics, we say a sentence is a tautology if any possible measurement always yields the value "true". Hence in twin open set logic, a proposition is a tautology if it is equal to the proposition \([X, \emptyset]\), where \( X \) is the topological space. By nontrivial we do not mean that a proposition is not equal to the always true statement. Rather, what we mean is that the statement is the result of several iterations of the negation and/or disjunction operators. So, for example, in classical logic \( P \lor \neg P \) is the nontrivial tautology referred to
as tertium non datur. In the Boolean restriction of discrete twin logic this is derived as follows:

\[ P \lor \neg P =_{\text{def}} [P_U, P_R] \lor [P_L, P_R] \]
\[ = [P_U \lor P_L, P_R \land P_L] \]
\[ = [P_S, \emptyset] \]
\[ = [X, \emptyset]. \]

Recall that in Boolean restriction of discrete twin logic we always have that \( P_S = X \). We also see why tertium non datur is not a tautology in general twin open set logics.

The reason there is only the trivial tautology in twin open set logic is seen by considering the truth tables for negation and disjunction:

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>P \lor Q</th>
<th>\neg P</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>I</td>
<td>T</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>I</td>
<td>I</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>I</td>
<td>I</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

If \( A, B, \) and \( C \) are three nonempty, pairwise disjoint subsets of \( X \) then we may let \( P = [A, B] = Q \) and let \( m \) be any measurement such that \( m \subseteq C \) then any set of applications of disjunction and/or negation to \( P \) will always yield the value of “\( T \)”, so that any sentence composed of iterated applications of these operations will not be equal to \( [X, \emptyset] \). As conjunction and implication can be defined in terms of disjunction and negation then there are no nontrivial tautologies that involve these operations as well.

### 1.6 Derivation Schemes in Collision Models

The alternatives we discuss further in this section are

1. **Modify the derivation rule**: The implication operation and the entailment given by \( \land \) are left in place but *modus ponens* is replaced by “the premises \( P \) and \( P \to Q \) yield \( [Q_U, P_R \land Q_L] \)” and *modus tollens* is replaced with “the premises \( \neg Q \) and \( P \to Q \) yield \( [Q_L, P_R \land Q_L] \)”.

2. **Replace the entailment given by \( \land \)**: *Modus ponens* and the implication operation are unchanged but the entailment given by partial ordering (\( \subseteq \)) derived from \( \land \) is replaced by the quasi-ordering \( \leq \) defined by
\[ P \leq Q \equiv \text{def} \begin{cases} \nexists v, \quad P_v \subseteq Q_v, \text{and} \vspace{1em} \\
exists f, \quad Q_f \cap P_f = Q_f \cap P_f. \end{cases} \]

As we have also seen in Sect. 1.4 other choices than those listed above are possible. Indeed, Subsect. 1.4.1 explicitly discussed alternative substitutes for *modus ponens* just as Subsect. 1.4.2 exhibited alternative quasi-orderings that could act as substitutes for “\(\leq\)”. We have chosen the alternatives listed above as each has been established to be minimal perturbations of the standard derivation rule found in classical Boolean logic. By “minimal”, we mean that when each alternative is chosen, each of the other two features of the derivation scheme is left unchanged.

We now ask: which alternative listed above is more amenable to this task of error detection than the other in a collision model? Let us consider a situation where the issue is relevant: we can imagine implementing a “derivation gate” that is the gate will accept as inputs the value of proposition \(P\) and \(P \rightarrow Q\). What would it mean for a computer to operate under regimes given by alternatives 1 or 2 listed above?

Before we turn to each alternative above we must consider how collision gates would implement the twin open set logic [18]. For definiteness let us consider billiard ball gates. The particular billiard ball gates we implement here will involve a control ball. We explain, in turn, the operation of the control ball gates that represent “or” (\(\lor\)), “and” (\(\land\)) and “not” (\(\neg\)). As implication will always be defined as \(\neg P \rightarrow Q\), these three gates will be sufficient to implement our so-called derivation gate.

Fig. 1.1. Collision of billiard balls in twin open set logic gates.
Preliminary to showing the three gates, we show a diagram (Fig. 1.1) that illustrates how the collisions of billiard balls will work in this system. As seen in Fig. 1.1, the left-hand side of each gate allows the passage of the control billiard ball (denoted by "C" in the figure). The gate will accept from the top margin the balls that represent the values of the inputs \(P\) and \(Q\). The right-hand side will allow the control ball to exit the gate, the value of the gate will be determined by the position of the control ball as it crosses the right-hand margin. The paths of \(P\), \(Q\) and \(C\) will be set so that \(C\) will have the opportunity to interact first with \(P\) and then it will have an opportunity to interact with \(Q\) regardless of whether or not \(P\) was present. This can be accomplished by angling the inputs for \(P\) and \(Q\) in such a way that the distance from the interaction point for \(C\) and \(P\) will allow \(C\) to pass to the interaction point for \(Q\) at the point labeled "\(x\)" if \(P\) is not present or to travel to the interaction point with \(Q\) labeled "\(y\)". The balls representing \(P\) and \(Q\) will exit the interaction regions so that they will not cross the right-hand margin of the gate.

This basic operation will be the same for each of the three gates, it is the determination of verifiability and falsifiability sets as subintervals of the right-hand margin that will distinguish each gate from the other two. It should be pointed out that these control billiard gates can be represented by two standard Fredkin-Toffoli interaction gates connected in series so that one input for the first gate is always present (this will be our control ball). This variant of billiard ball gates is particularly amenable to analysis by a twin open set logic based on open subintervals of the right-hand margin. In a sense, the control billiard provides a timing mechanism for the operation of the gates.

The verifiability, falsifiability and indeterminacy sets for the propositions \(P\) and \(Q\) will correspond to subintervals of the right-hand margin as shown in Fig 1.1. We now explain why this is so. Consider the case in which both \(P\) and \(Q\) are false; this is represented physically by the absence of both balls. In the gate, this will result in the control billiard "\(C\)" passing through the gate unperturbed. Thus interval "\(a\)" will be the falsifiability sets of both \(P\) and \(Q\). Note that the falsifiability set for \(P\) is \(a \cup e\) while its verifiability set is \(b\). The size of intervals will depend on the degree of resolution. Should the ball fall within the indeterminacy set (the union of the three shaded areas in Fig 1.1) this will be a signal that the execution of the gate is erroneous.

We now explain the interpretation of the gate for the "or" operator. This is shown in Fig. 1.2. The only case in which \(P \lor Q\) takes the value "false" is when each ball for \(P\) and \(Q\) is absent. In the gate, this will be represented by the control billiard "\(C\)" passing through the gate unperturbed. Thus interval "\(a\)" will be the falsifiability set for \(P \lor Q\). If the ball for \(P\) is present but the ball for \(Q\) is not then the control billiard will cross interval "\(b\)". If the ball for \(P\) is absent but the ball for \(Q\) is present then the control billiard will cross interval "\(c\)". If the ball for \(P\) is present and so is the ball for \(Q\) then the control ball
Fig. 1.2. Collision-based implementation of the OR-gate of twin open set logic.

will cross interval “d”. The verifiability set for $P \lor Q$ will therefore be the union of intervals “b”, “c” and “d”.

Fig. 1.3. Collision-based implementation of the AND-gate of twin open set logic.

The operation of the “and” (\$\land\$) gate, Fig. 1.3, is essentially the same. The difference between the two gates comes in the identity of the verifiability and falsifiability sets. $P \land Q$ is assigned the value “true” if and only if the balls for both $P$ and $Q$ are present. Hence interval “d” is the verifiability set for the “and” gate. The falsifiability set for the “and” gate is the union of intervals “a”, “b” and “c”.

Fig. 1.4. Collision-based implementation of the NOT-gate of twin open set logic.

Again the operation of the “not” gate, Fig. 1.4, is similar. In the case of a “$\neg P$” gate, the input for $Q$ is blocked, so only subintervals “a” or “b” will be traversed by the control ball. Subinterval “a” will be the verifiability set for $\neg P$ and subinterval “b” will be its falsifiability set.
We now consider the regime given by alternative 1 above. The attraction of this alternative for a collision model is that the falsifiability set does not depend upon whether the input is for "P true" or "Q false". That is, the gates will implement \textit{modus ponens} or \textit{modus tollens}. A negative aspect of this scheme is that it would require us to differentiate between the situation where our gates represent \( P \land (P \rightarrow Q) \) and an application of \textit{modus ponens} to \( P \) and \( P \rightarrow Q \). Such a distinction is usually not made at the machine (object language) level.

This leaves alternative 2. That is, we provide a nonstandard interpretation of entailment. This means that at the collision level, the definitions of implication and \textit{modus ponens} are unchanged. In this regime we must be careful to remember that \( P \sqsubseteq Q \) does not mean that \( Q \) is more verifiable and less falsifiable than \( P \). In particular we cannot say that \( Q \) is more verifiable and less falsifiable than \( P \land (P \rightarrow Q) \). The quasi-ordering presented here is one possible alternative.

### 1.7 Conclusion

At several points in the development above, it is clear that the topology imposed on the space of propositions and/or the space of measurements has important consequences. Indeed, if we modify the underlying topology then we have a different twin open set logic. Thus, a possible path for future consideration is to consider such topological modifications giving rise to particular twin open set logics. We now consider a few of the questions that might be considered in such a context.

We note an interesting progression presented in Sect. 1.3. If we allow for measurements of infinite precision, then we obtain a three-valued logic that is fully truth-functional. The topology of the underlying space that represents measurements of infinite precision is the discrete topology. Thus, at least in the setting of twin open set logics, the discrete topology gives rise to full truth functionality.

The restriction of the resulting discrete twin topology to propositions of the form \( [A, A^c] \) is not (strictly speaking) a topological modification. This restriction still employs the discrete topology. What has changed is the theory of measurement. This change in measurement, in turn, implies a change in the lattice of substructures (in this case the ordered pairs of disjoint open sets) that play the role of propositions in this particular derived logic.

In the discrete twin logic measurements of infinite precision are allowed but resources for determining truth or falsity are limited. For example, a proposition of the form \( [A, B] \) where \( A \cup B \neq X \) reflects that we only have access to region \( A \) to verify the proposition and to region \( B \) to falsify it. In contrast, a proposition of the form \( [A, A^c] \) reflects access to the full space \( X \) to either verify or falsify the proposition.
These considerations show that a change in the theory of measurement can lead to a change in the topology or a change in the lattice of propositions. The former is exemplified by the change in the theory of measurement in passing from general twin open set logics to the discrete twin logic. The latter is exemplified by the change in the theory of measurement in passing from the discrete twin logic to its Boolean restriction.

Recall that twin open set logics are members of a wider class of structures known as derived logics. Changes in the theory of measurement that relates to a given derived logic will result in similar changes to either the underlying mathematical structure of the propositions or to the lattice structure given by the propositions. A general theory of the relationship between the following is desirable but as yet undeveloped:

1. the theory of measurement for a given physical system,
2. the mathematical structure that reflects the theory of measurements, and
3. the resulting lattice (or lattice-like structure) of mathematical substructures of item 2 that play the role of propositions in the derived logic.

We found in Subsect. 1.4.3 that a modification to implication alone was not compatible with the Boolean restriction to the discrete twin logic for general twin open set logics. This suggests the following questions:

- Is there a twin open set logic in which a modified implication operator would be compatible with the limiting case of the Boolean restriction to the discrete twin logic?
- Is there a twin open logic for which the Boolean restriction to the discrete twin logic could not be considered to be a limiting case?

As we noted in Sect. 1.5 above, it may be possible to modify more than one of the three aspects of derivation schemes to arrive at a more useful system for a particular computational model. Only one particular collision model has been considered here. In some sense we have considered only the least radical modifications.

The conclusion reached here is that the least radical modification for the billiard ball model considered here is to maintain the standard form of implication and modus ponens while changing the meaning of entailment. That is, we have argued that a quasi-ordering of our propositions leads to a natural logic that is reflected by the operation of the billiard ball gates considered here. To summarize the argument: Subsect. 1.4.3 showed that the implication operator that is compatible with the Boolean restriction of twin discrete logic is not compatible with a standard entailment and modus ponens. Modifications to the implication operator only are rejected on the basis of this fact. Section 1.6 showed that modifications to the derivation method only lead to an undesirable distinction between gates implementing instances of the modified version of modus ponens or modus tollens and gates implementing instances of \( P \land (P \rightarrow Q) \). Section 1.6 also showed that modifications to the
entailment would not operate at the gate (or syntactic) level but rather in the interpretation (semantics) of the output of the gates.

The points raised in the present section suggest other possibilities. For example, there may be modified collision models with theories of measurement sufficiently restrictive so as to allow a modified implication operation together with standard derivation method and entailment. In a sense this is what occurs when we use the unphysical assumptions (infinite precision and unlimited access) to derive the Boolean restriction of the twin discrete logic. Of course still another possibility would be to change more than one of the aspects of a derivation scheme at the same time. We leave such considerations to future research.

Acknowledgment

The authors gratefully acknowledge their indebtedness to Benjamin Schumacher for his many helpful comments, suggestions and ideas.

References


