JOURNAL OF APPLIED NON-CLASSICAL LOGICS
https://doi.org/10.1080/11663081.2024.2366752



# Truth diagrams for some non-classical and modal logics

Can Başkent

Department of Computer Science, Middlesex University, London, UK

**ABSTRACT**

This paper examines truth diagrams for some non-classical, modal and dynamic logics. Truth diagrams are diagrammatic and visual ways to represent logical truth akin to truth tables, developed by Peter C.-H. Cheng. Currently, it is only given for classical propositional logic. In this paper, we establish truth diagrams for Priest's Logic of Paradox, Belnap–Dunn's Four-Valued Logic, MacColl's Connexive Logic, Bochvar–Halldén's Logic of Non-Sense, Carnielli–Coniglio's logic of formal inconsistency as well as classical modal logic and its dynamic extension to shed light on the semantic behaviour of some non-classical and modal logics.

**ARTICLE HISTORY**
Received 4 July 2023
Accepted 20 April 2024

**KEYWORDS**
Truth diagrams; paraconsistent logic; non-classical logic; modal logic

## 1. Introduction

Diagrams have always been used in mathematics since historical times, Euclid being a notable example. Any student of mathematics is familiar with the use of diagrams in geometric reasoning and aiding proofs by visualisation (Nelsen, 1993). Arguably, such diagrammatic reasoning offers a 'real extension to our knowledge' (Macbeth, 2014). There seems to be no reason why logic would be an exception to this tradition.

*Truth diagrams* (TDs, for short) were first proposed by Cheng (2020), and offer an alternative graphical and visual representation for the semantics of classical propositional logic. What motivated TDs include Frege's and Wittgenstein's systems, and particularly, TDs are largely based on Wittgenstein's Tractatus Diagrams. Cheng's TDs, however, include variables and positions to indicate the truth value of a logical formula. As such, TDs are alternatives to truth tables, not to syntax, offering a new system of semantics. The work on TDs so far has remained focused on classical propositional logic. An immediate benefit of this is to have an intuitive and simple diagrammatic representation for classical logic. However, there are also disadvantages. First, the theoretical and graphical depth and breadth of truth diagrams remain understudied. Exploring this potential requires us to test TDs in different domains, such as non-classical and modal logics. Second, applying TDs to other logical systems allow us to compare and contrast semantical systems for a broader systems of logics, and explain their 'non-classicity' using different tools. Consequently, developing TDs for a variety

**CONTACT** Can Başkent c.baskent@mdx.ac.uk

of logical systems will contribute to the debate on the role of visual reasoning in mathematics and logic (Giaquinto, 2007, 2008) as well as offer an alternative semantics for various non-classical logics. This is our goal in this paper.

Non-classical logics diverge from classical logic in a variety of different ways. Largely confined to philosophical debates, semantic structures, visual representations and diagrams for non-classical logic long suffered from being limited to truth tables and algebraic structures. This is yet another symptom of a bigger problem: non-classical logics have not yet benefited from having a wide variety of semantics. The current study complements an earlier work where game theoretical semantics for them were presented, expanding the semantic structures of non-classical logics (Başkent, 2016, 2020; Başkent & Henrique Carrasqueira, 2020). Unlike semantics, proof theory has long enjoyed diagrammatic reasoning. Within the broad category of proof semantics, there is a large body of research on diagrammatic reasoning, including the work in game semantics for programming languages, category theory and proof nets (Abramsky & McCusker, 1999; Awodey, 2006; Girard, 1987). This paper, however, focuses on semantics for truth.

In this work, particularly, we extend TDs to propositional multi-valued, non-classical, modal and dynamic (epistemic) logics. First, we briefly discuss the philosophical background of diagrammatic reasoning in logic and mathematics to establish a context for our work. Then, we introduce TDs for classical logic, following Cheng's original work. Next, we introduce various well-known non-classical logics, including Priest's Logic of Paradox, Belnap and Dunn's 4-valued First-Degree Entailment, MacColl's Connexive Logic, Bochvar and Halldén's Logic of Non-sense, Carnielli and Coniglio's Logic of Formal Inconsistency, modal logic of S5 and public announcement logic. The main contribution of the paper follows where we develop TDs for the aforementioned logics and present some examples alongside their diagrams. Finally, we conclude with a discussion and future work ideas.

## 2. Background

As Giaquinto argued, visualising in mathematics has 'epistemically significant uses' (Giaquinto, 2008). One may argue this should include logic, yet, Giaquinto does not discuss the role of visual thinking in logic (Giaquinto, 2007). He discusses the role of diagrams in proofs without explaining how diagrams may help in logic and semantics. Macbeth, on the other hand, fills in this gap by discussing diagrammatic reasoning in logic by means of Frege (Macbeth, 2014). Carter's analysis of diagrams in mathematics underlines how fruitful they can be, questioning the role of Hilbertian formalism in mathematics (Carter, 2019).

Diagrammatic reasoning is not limited to computational and mathematical sciences. Category theory, one of the most popular and successful areas of diagrammatic reasoning in mathematics, has wide applications in natural sciences. Baez and Lauda's survey of categories in physics briefly discusses their roles in Feynman diagrams, string theory, quantum gravity, and topological quantum field theory (Baez & Lauda, 2011). Similarly, 'string diagrams' in theoretical physics is an interesting example of diagrammatic reasoning in physics (Bonchi et al., 2022a, 2022b).

Use of diagrammatic reasoning in proof theory and the semantics of programming languages is very prominent. Together with a wide use of category theory, there can be

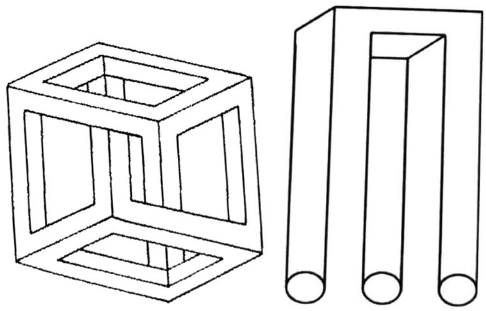


**Figure 1.** Escher's Cube (left) and Schuster's Fork, taken from Mortensen (2010).

found many uses of diagrams in the semantics of programming languages. Abramsky and McCusker discuss various applications of game semantics to programming languages (Abramsky & McCusker, 1999). A plethora of works followed up discussing the semantics of proofs, including (Hughes, 2006), and created a broad field of game semantics of programming languages and the semantics of proof theory. This line of research, however, falls outside the scope of this paper.

Oddly, such debates only consider classical logic. Even if non-classical logics often suggest a different understanding of truth, diagrammatic reasoning has not widely been used to *explain* such differences. However, there is some programmatic benefits for the use of diagrammatic reasoning in non-classical logics. As Giaquinto argued, visualisation helps in mathematical discovery, proofs as well as 'augmenting [our] understanding', latter of which is especially important for '… not only grasping the correctness of a claim, method or proof, but also appreciating why it is correct' in non-classical logic (Giaquinto, 2008) .[1]

Even if there has not been a notable trend in non-classical logic for the use of diagrams, there have been some exceptions. Notably, Mortensen's work on inconsistent geometry presents some insights (Mortensen, 2010). Mortensen's aforementioned work studies 'impossible' pictures, such as Escher's Cube and Schuster's fork (see Figure 1). As such, Mortensen uses diagrams to feed into inconsistency-friendly logics, where the impossible pictures also serve as ontological arguments for non-classical logics and dialatheic truth.

Non-classical logics introduce alternative understandings of truth – some allow truth gaps, some truth gluts. Some have more truth values, some have fixed-points under negations. It is therefore a curious task to examine how TDs approach such 'oddities'. As we mentioned earlier, such a task would advance the research in both diagrammatic reasoning and non-classical logics. In sequel, that is what we achieve.

## 3. Truth diagrams for classical logic

Let us first start with reviewing TDs for classical logic, following Cheng's original work (Cheng, 2020).

First, the elements of TDs. A TD is composed of three elements: *letters* for propositional variables ($p$, $q$ …), *nodes* to identify the positions around the letters, and

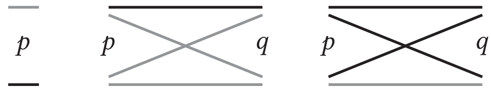



**Figure 2.** Truth diagrams for negation, conjunction and disjunction, respectively.

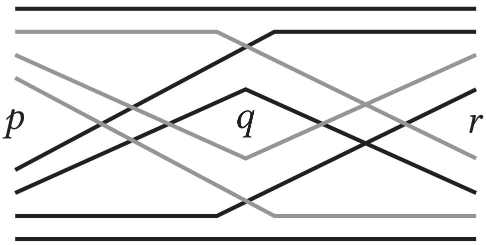



**Figure 3.** Truth diagram for $p \rightarrow q \wedge r$ in propositional logic.

*connectors* to link nodes. There are two nodes for each letter in propositional logic: high and low. High node is reserved to identify truth, whereas the low node is for falsity. Connectors are colour-coded. When they represent truth they are black; when they represent falsify, they are grey. Since we have only two truth values in classical propositional logic, there is no more colour-coding other than black and grey. We also do not allow vertical connectors for the same letter. This would represent a contradiction, which is not allowed in classical propositional logic. In classical logic, connectors intersect just one node for each letter.

For example, let us consider the TD for conjunction in Figure 2. The formula $p \wedge q$ is only satisfiable when both $p$ and $q$ are true. Thus, the black connector between the high nodes of the letters $p$ and $q$. All the other connectors between all the other possible nodes remain grey, signifying falsity. For example, the connector between the low nodes of both letters must be grey for falsity. The truth diagrams for negation, conjunction and disjunction are given in Figure 2.

As another example, let us consider the formula $p \rightarrow q \wedge r$, given in Figure 3. For this formula, we have five truth conditions under which it is true – hence five black connectors. The formula $p \rightarrow q \wedge r$ is true when (i) all $p, q, r$ are true, (ii) $p$ is false, $q$ and $r$ are true, (iii) $p$ is false, $q$ is true and $r$ is false, (iv) $p$ and $q$ are false, $r$ is true, (iv) all $p, q, r$ are false. The rest of the combinations of truth values for $p$, $q$ and $r$ render the formula false – hence three grey connectors.

The above examples will help us define TDs, following Cheng (2020). First, the elements of TDs.

**Definition 3.1:** Elements of truth diagrams for classical logic are given as follows.

- A TD is composed of *letters*, *nodes* and *connectors*.
- Letters are arranged horizontally (with regular spacing for readability).
- Nodes are small areas, one above and one below the letters.

- Connectors are lines linking nodes. Each connector intersects just one node at each letter and has straight segments that span pairs of immediately adjacent letters.
- One connector for each possible combination of high or low nodes of each (type of) letter is permitted: the shape of each connector in a TD is unique.
- The style of the connectors is solid and either black or grey.
- A TD can contain more than one instance of a letter.
- The horizontal order of the letters is arbitrary.
- Letters in separate TDs are not linked by connectors.
- A connector intersects the nodes at the same level for each instance of the same letter.

Now, the semantics of TDs is defined as follows.

**Definition 3.2:** Semantics of truth diagrams for classical logic is defined as follows.

- Letters are propositional *variables*.
- Each node represents a truth-value for the variable: high-node *T*, low-node *F*.
- The number of the distinct types of variables is the arity of the TD.
- A connector is a *case*: it constitutes a unique set of truth-value assignments to the variables.
- Connector style represents the overall truth-value assigned to its case. A black connector assigns *T*, grey connector assigns *F*.

It is important to note the limitations of classical logic here in classical TDs. First, logical connectives are functional, for that reason the connectors cannot skip a letter as each letter must have a truth value. Second, the classical case does not allow us to represent multi-valued logics.

## 4. Non-classical and modal logics

In this section, we briefly present some well-known and well-studied non-classical logics as well as modal logic S5, a relatively straight forward example of modal logics. Consequently, in the following section, we examine how TDs can be defined for them.

For our purposes, we examine the following well-studied and well-known non-classical logics: (i) Priest's Logic of Paradox, (ii) Belnap and Dunn's four-valued first-degree entailment, (iii) MacColl's connexive logic, (iv) Bochvar and Halldén's logic of non-sense, and (v) Carnielli and Coniglio's logic of formal inconsistency. A more comprehensive approach to non-classical logic can be found in Priest (2008) and Carnielli and Coniglio (2016). Broadly speaking, these logics belong to different classes of logics. The logic of paradox and the logic of formal inconsistency are paraconsistent. MacColl's system is one of the earliest systems of connexive logics. Belnap and Dunn's system is one of the most well-known systems of four-valued logics. They all have different logical and philosophical motivations.

| | ¬ |
|---|---|
| $T$ | $F$ |
| $F$ | $T$ |
| $P$ | $P$ |

| ∧ | $T$ | $P$ | $F$ |
|---|---|---|---|
| $T$ | $T$ | $P$ | $F$ |
| $P$ | $P$ | $P$ | $F$ |
| $F$ | $F$ | $F$ | $F$ |

| ∨ | $T$ | $P$ | $F$ |
|---|---|---|---|
| $T$ | $T$ | $T$ | $T$ |
| $P$ | $T$ | $P$ | $P$ |
| $F$ | $T$ | $P$ | $F$ |

**Figure 4.** The truth tables for LP and K3.

| | ¬ |
|---|---|
| $T$ | $F$ |
| $P$ | $P$ |
| $N$ | $N$ |
| $F$ | $T$ |

| ∧ | $T$ | $P$ | $N$ | $F$ |
|---|---|---|---|---|
| $T$ | $T$ | $P$ | $N$ | $F$ |
| $P$ | $P$ | $P$ | $F$ | $F$ |
| $N$ | $N$ | $F$ | $N$ | $F$ |
| $F$ | $F$ | $F$ | $F$ | $F$ |

| ∨ | $T$ | $P$ | $N$ | $F$ |
|---|---|---|---|---|
| $T$ | $T$ | $T$ | $T$ | $T$ |
| $P$ | $T$ | $P$ | $T$ | $P$ |
| $N$ | $T$ | $T$ | $N$ | $N$ |
| $F$ | $T$ | $P$ | $N$ | $F$ |

**Figure 5.** The truth table for BL.

### 4.1. Priest's logic of paradox

The logic of paradox (LP, for short) introduces an additional truth value $P$, called *paradoxical*, which intuitively stands for both true and false (Priest, 1979).

The logics LP and Kleene's three valued logic K3 have the same truth tables. However, they differ on the truth values that they preserve in valid inferences, and how they read $P$. The truth values that are preserved in validities are called *designated truth values* and they can be thought of as the extensions of the classical notion of truth (Priest, 2008). In LP, it is the set $\{T, P\}$; in K3 (and classical logic), it is the set $\{T\}$. Even if the truth tables of two logics are the same, different sets of designated truth values produce different sets of validities, thus different logics. For instance, $p \vee \neg p$ is a theorem in LP, but not in K3. In K3, the third truth value has an intuitionistic reading and can be viewed as an undervaluation in contrast to its reading as an overvaluation in LP. It is also important to note that the set of validities of LP contains the set of validities of the classical logic.

The conditional arrow $\rightarrow$ can be taken as an abbreviation in the usual sense: $p \rightarrow q \equiv \neg p \vee q$. We give the truth table of LP in Figure 4.

### 4.2. Belnap–Dunn's four-valued system

Belnap's four valued logic (BL, for short) introduces two additional truth values besides the classical ones. The truth value $P$, as before, represents over-valuation, and $N$ represents under-valuation. Traditionally, $P$ stands for both truth values and $N$ stands for neither of the truth values. As the truth table in Figure 5 indicates, $P$ and $N$ are the fixed-points under negation.

With a slight abuse of notation, the 'problematic' formulas in BL are $P \vee N \equiv T$ and $P \wedge N \equiv F$. A Hasse-style truth value lattice for BL is given in Figure 6.

The above Hasse diagram illustrates the standard method to compute disjunction and conjunction of two truth values as the least upper bound and the greatest lower

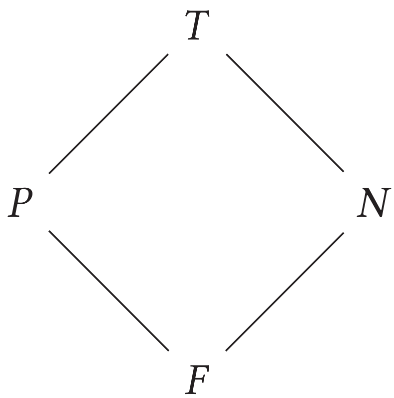



**Figure 6.** Hasse diagram for BL.

| | ¬ |
|---|---|
| *T* | *F* |
| *t* | *f* |
| *f* | *t* |
| *F* | *T* |

| ∧ | *T* | *t* | *f* | *F* |
|---|---|---|---|---|
| *T* | *T* | *t* | *f* | *F* |
| *t* | *t* | *T* | *F* | *f* |
| *f* | *f* | *F* | *f* | *F* |
| *F* | *F* | *f* | *F* | *f* |

| ∨ | *T* | *t* | *f* | *F* |
|---|---|---|---|---|
| *T* | *t* | *T* | *t* | *T* |
| *t* | *T* | *t* | *T* | *t* |
| *f* | *t* | *T* | *F* | *f* |
| *F* | *T* | *t* | *f* | *F* |

**Figure 7.** The truth table for CC.

bound of the two values respectively. Then, with a slight abuse of notation, it is possible to read off $P \vee N \equiv T$ and $P \wedge N \equiv F$ from the diagram. For simplicity, we take the designated values for BL as $\{P, T\}$.

### *4.3. MacColl's connexive logic*

As Wansing argued, connexive logic remains a 'comparatively little-known and to some extent neglected branch of non-classical logic' (Wansing, 2015). In this work, we focus one of the earliest examples of connexive logics CC, which is due to McCall (1966).

Connexive logic is defined as a system which satisfies the following two schemes of conditionals:

- *Aristotle's Theses*: $\neg(\neg\varphi \to \varphi)$
- *Boethius' Theses*: $(\varphi \to \neg\psi) \to \neg(\varphi \to \psi)$

The logic CC is axiomatised by adding the scheme $(\varphi \to \varphi) \to \neg(\varphi \to \neg\varphi)$ to the axiomatisation of classical propositional logic. The rules of inference for CC is modus ponens and adjunction, which is given as $\vdash \varphi, \vdash \psi \therefore \vdash \varphi \wedge \psi$. We consider CC with the standard propositional syntax.

The semantics for CC is given with 4 truth values: *T*, *t*, *f* and *F* which can be viewed as 'logical necessity', 'contingent truth', 'contingent falsehood', and 'logical impossibility' respectively (Routley & Montgomery, 1968). In CC, the designated truth values are *T* and *t*. The truth table for CC is given in Figure 7.

| | ¬ |
|---|---|
| $T$ | $F$ |
| $N$ | $N$ |
| $F$ | $T$ |

| ∧ | $T$ | $N$ | $F$ |
|---|---|---|---|
| $T$ | $T$ | $N$ | $F$ |
| $N$ | $N$ | $N$ | $N$ |
| $F$ | $F$ | $N$ | $F$ |

| ∨ | $T$ | $N$ | $F$ |
|---|---|---|---|
| $T$ | $T$ | $N$ | $T$ |
| $N$ | $N$ | $N$ | $N$ |
| $F$ | $T$ | $N$ | $F$ |

**Figure 8.** The truth table for BH3.

### 4.4. Bochvar–Halldén's logic of non-sense

Bochvar–Halldén Logic introduces an additional truth value $N$, called *nonsense*, which intuitively stands for sentences which are nonsensical or meaningless. Bochvar–Halldén logics are actually two distinct logics with the same truth table. In Bochvar's system, the designated truth value is $T$ whereas in Halldén's it is $\{T, N\}$. For our semantic considerations, we treat them together and call this formalism the Bochvar–Halldén Logic (BH3, for short) with the following truth table, given in Figure 8.

### 4.5. Carnielli-Coniglio's logic of formal inconsistency

Logics of formal inconsistency (LFI, for short) extend da Costa systems and generate a broad class of paraconsistent logics (Carnielli et al., 2007; da Costa et al., 2007). In this work, we focus on a particular LFI, called mbC. The system mbC exhibits some of the most important aspects of LFIs, as it is 'strong enough to contain the germ of classical negation, possessing a kind of hidden classical negation' and contains a consistency operator (Carnielli & Coniglio, 2016). Compared to the other systems we have presented, LFIs are more complex systems.

Let us start with defining the language $\mathcal{L}$ of mbC. Given a set of propositional variables $P$, we define the syntax of mbC in the Backus–Naur form as follows, where $p \in P$.

$$\varphi ::= p \mid \neg\varphi \mid \circ\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \to \varphi$$

One of the features which distinguishes mbC and other LFIs from other paraconsistent logics is their use of the *consistency* operator $\circ$. The consistency operator simply checks whether a formula $\varphi$ *explodes* – that is if $\varphi, \neg\varphi \vdash \psi$ for all formula $\psi$ in the language. This allow us to distinguish and *control* the formulas that can explode the model.

A model $M$ for mbC is a tuple $M = (S, V)$ where $S$ is a non-empty set and $V : \mathcal{L} \mapsto \{T, F\}$ is a valuation function. The function $V$ for mbC assigns a unique truth value to propositional variables, and satisfies the following conditions (Carnielli & Coniglio, 2016):

- $V(\neg\varphi) = F$ then $V(\varphi) = T$,
- $V(\circ\varphi) = T$ then $V(\varphi) = F$ or $V(\neg\varphi) = F$,
- $V(\varphi \to \psi) = T$ if and only if $V(\varphi) = F$ or $V(\psi) = T$,
- $V(\varphi \wedge \psi) = T$ if and only if $V(\varphi) = T$ and $V(\psi) = T$,
- $V(\varphi \vee \psi) = T$ if and only if $V(\varphi) = T$ or $V(\psi) = T$.

In this semantics, the truth values of $\neg\varphi$ and $\circ\varphi$ are not necessarily determined by the truth value of $\varphi$. That is, for instance, if $V(\varphi) = T$, then $V(\neg\varphi)$ is not determined based on $V(\varphi)$. It can be either $T$ or $F$, but not both nor neither. Therefore, the

| $p$ | $\neg p$ | $\circ p$ | $p \wedge \neg p$ |
|---|---|---|---|
| T | T | F | T |
| | F | T | F |
| | | F | F |
| F | T | T | F |
| | | F | F |

**Figure 9.** The truth table for some formulas in mbC.

valuation function $V$ is functional, but not truth-functional: mbC valuations are, just as classical valuations, simple functions, but mbC logical operations themselves are *not* functions from tuples of truth values to truth values, but *multifunctions* instead (Carnielli et al., 2007). They assign to the given tuple of truth values a set of possible truth values, from which an mbC valuation is then to pick one for the value of the corresponding complex formula. This valuation is sometimes called *bivaluation*. We give the non-deterministic truth table for some formulas in mbC in Figure 9.

Non-determinacy is one of the complications of mbC and LFIs in general. Furthermore, compared to classical propositional logic, mbC has an extended language with the consistency operator – if a formula and its negation are both true, then the consistency of the formula must be false. Second, perhaps semantically more importantly, mbC assumes that a formula and its negation are *subcontraries*, but not necessarily *contraries*. That is they cannot both be false under the same valuation, but they *can* both be true under the same valuation.

### *4.6. Modal logic*

Classical modal logic is a well-studied branch of logic, extending classical propositional logic. For the completeness of our treatment, we briefly discuss its syntax and semantics (Blackburn et al., 2001).

Given a set of propositional variables $P$, we define the syntax of modal logic K in the Backus-Naur form as follows, where $p \in P$.

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \to \varphi \mid \Box\varphi \mid \Diamond\varphi$$

Intuitively, $\Box\varphi$ means that $\varphi$ is necessarily the case, and $\Diamond\varphi$ means that $\varphi$ is possibly the case. Classically, $\Box$ and $\Diamond$ are their duals: $\Box\varphi \equiv \neg\Diamond\neg\varphi$.

Modal formulas are interpreted on modal models $M = (W, R, V)$ where (i) $W$ is a non-empty set of states, (ii) $R \subseteq W \times W$ is a binary relation (called the 'accessibility relation'), and (iii) $V$ is a valuation function assigning to each propositional variable $p$ as set $V(p)$ of states.

The truth of (classical) modal formulas is intensional as it depends not only on the truth of the formulas but also the accessibility of the state that they are evaluated in. We give the semantics of the modal formulas as follows, skipping the obvious propositional cases.

$$M, w \models \Box\varphi \quad \text{if and only if} \quad M, v \models \varphi, \text{ for all } v \in W \text{ with } wRv.$$
$$M, w \models \Diamond\varphi \quad \text{if and only if} \quad M, v \models \varphi, \text{ for some } v \in W \text{ with } wRv.$$

Classical modal logic is a rich research area as the modal operators can be interpreted in a wide variety of different ways. Epistemically, $\Box\varphi$ means 'I know that $\varphi$', doxastically it means 'I believe that $\varphi$', and deontically it means 'It is obligatory that $\varphi$'. A similar reading for various computer scientific concepts can be given for modal operators and their duals. As emphasised earlier, classical logic allows us to define modal dualities, along with their epistemic, doxastic etc. interpretations.

Modal logic is axiomatised with the axioms of propositional logic, the modal normality axiom (otherwise known as the Kripke Axiom) $\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$ and the necessitation rule (from $\vdash \phi$ follows $\vdash \Box\phi$ where $\vdash$ stands for syntactic derivability). The basic classical modal logic axiomatised by the normality axiom and the necessitation rule is called logic K.

Introducing additional axioms produces different modal logics. Let us first see some axioms.

**T** $\Box\varphi \to \varphi$
**4** $\Box\varphi \to \Box\Box\varphi$
**5** $\Diamond\varphi \to \Box\Diamond\varphi$

A modal logic is called S4 if it is axiomatised by those of logic K and axioms T and 4. Modal logic S5 is obtained by adding axioms T, 4 and 5 to K. For S5 for *n*-many agents is denoted as $S5_n$ where each agent *i* has its corresponding accessibility relation $R_i$ in the model.

In this work we focus on modal logic S5 due to its simplicity: in S5 the accessibility relation is an equivalence relation. Under these conditions, it is possible to simplify the equivalence relation, too.

It is important to note that the classical S5 can be characterised by the models of universal frames where every state is accessible from every state. In S5, $\Box\varphi$ is satisfied if and only if $\varphi$ is satisfied at each state. Similarly, $\Diamond\varphi$ is satisfied if and only if $\varphi$ is satisfied at some state. This simplicity makes it easier to use TDs for S5.

### *4.7. A dynamic modal logic: public announcement logic*

An interesting extension of modal logic, given originally for the epistemic reading of the modal operator, allows model updates. Dynamic logics in general express changes in the model using syntactic operators. In this section, we focus on a well-studied example of it: Public Announcement Logic for S5 (PAL5), where the underlying epistemic logic is S5 (van Ditmarsch et al., 2007). For simplicity, we only discuss a single agent version.

PAL5 is an extension of S5 where changes in agents' knowledge is expressed using a new modal operator $[\varphi!]$. The way the model is updated is governed by an external information that is introduced – often called an announcement. The announcement is considered truthful. Consequently, the states that do not agree with the announcement are eliminated. As a result, the accessibility relation is also redefined after the removed states.

The syntax of PAL5 is that of the modal logic extended with a dynamic modal operator $[\varphi!]\varphi$:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \rightarrow \varphi \mid \Box\varphi \mid \Diamond\varphi \mid [\varphi!]\varphi$$

Consequently, the semantics of the dynamic operator is given as follows.

$$M, w \models [\varphi!]\psi \quad \text{if and only if} \quad \text{if } M, w \models \varphi \text{ then } M|\varphi, w \models \psi$$

where the updated model $M \mid \varphi = (W \mid \varphi, R \mid \varphi, V \mid \varphi)$ is defined as $W \mid \varphi = \{w \in W : M, w \models \varphi\}$, $R \mid \varphi = R \cap (W \mid \varphi \times W \mid \varphi)$, $V \mid \varphi = V \cap W \mid \varphi$.

## 5. Truth diagrams for non-classical logics

Now, we introduce TDs for the logical systems we have seen earlier. Our goal is to offer a catalogue of TDs for various non-classical logics emphasising the visual elements of non-classicity. Our methodological approach will allow researchers to build TDs for other logical systems that share common elements with the ones that we cover here. We supplement our introduction with some diagrammatic examples, too. This is the main contribution of the current work.

### *5.1. Truth diagrams for priest's logic of paradox*

For Logic of Paradox, we employ three positions around a propositional letter: one for true, one for false, and a middle one for the third (paradoxical) truth value *P*. For the paradoxical truth value, we will use a differently coloured line – a shade of red. Using the truth table for LP given in Figure 4, we produce the following figure for the connectives in LP in Figure 10.

Let us briefly explain how the TD for LP is constructed for the connectives given in Figure 10. Since the truth value *P* is a fixed-point under negation, we put a connective on the middle-nod around *p*. For conjunction, we have the truth value *P* for the cases when (i) both *p* and *q* have the truth values *P*, (ii) *p* is *T* and *q* is P, and (iii) *p* is *P* and *q* is *T*. Such cases are represented by a red connector. The cases are very similar for disjunction.

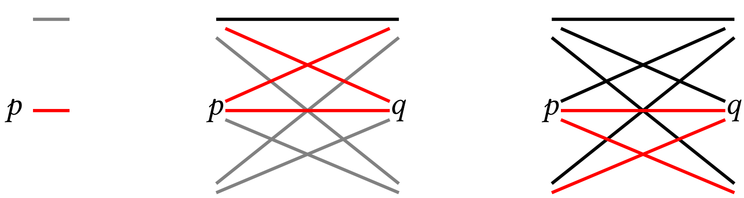



**Figure 10.** Truth diagrams for negation, conjunction and disjunction in LP, respectively, where the red colour represents the paradoxical truth value *P*.

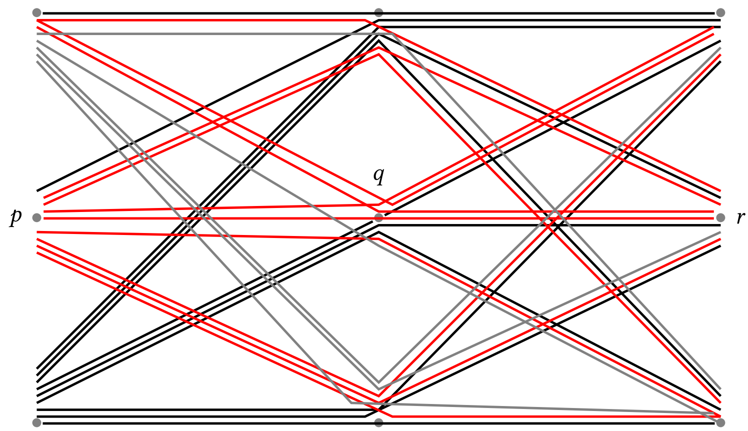



**Figure 11.** Truth diagram for the formula $p \to q \wedge r$ in logic of paradox.

Now, as an example, let us consider the TD of the very same formula $p \to q \wedge r$ which we have discussed earlier in LP. The TD, given in Figure 11, shows that the formula is true for 11 truth conditions, false for 5 truth conditions, and paradoxical for the remaining 11 truth conditions out of 27 possibilities in LP.

The difference in complexity between Figures 3 and 11 underlines how complicated the TDs can get for multi-valued logics. For clarification, we also decompose the truth diagram in Figure 11 into three sub-diagrams, each with just a single colour, see Figure 12 .[2]

Notice that designated truth values can be represented in TDs by simply noting the colour of the truth values that are considered designated. Diagrammatically, this makes it easier to keep track of the designated truth values and their interaction with the other truth values and logical connectives. Particularly, the black and red connectors represent the designated truth values of LP in TDs.

Now, we can define TDs for LP more formally, *only* emphasising the differences in the definitions for the logic of paradox. Let us start with the elements of TDs for LP.

**Definition 5.1:** Elements of truth diagrams for the logic of paradox are given as follows.

- A TD is composed of *letters*, *nodes* and *connectors*.
- Letters are arranged horizontally (with regular spacing for readability).
- Nodes are three small areas, one above, one in the middle of and one below the letters.
- One connector for each possible combination of high, middle or low nodes of each letter is permitted.
- The style of the connectors is solid and either black, grey or red.

The semantics of TDs for LP is defined as follows.

**Definition 5.2:** Semantics of truth diagrams for the logic of paradox is defined as follows.

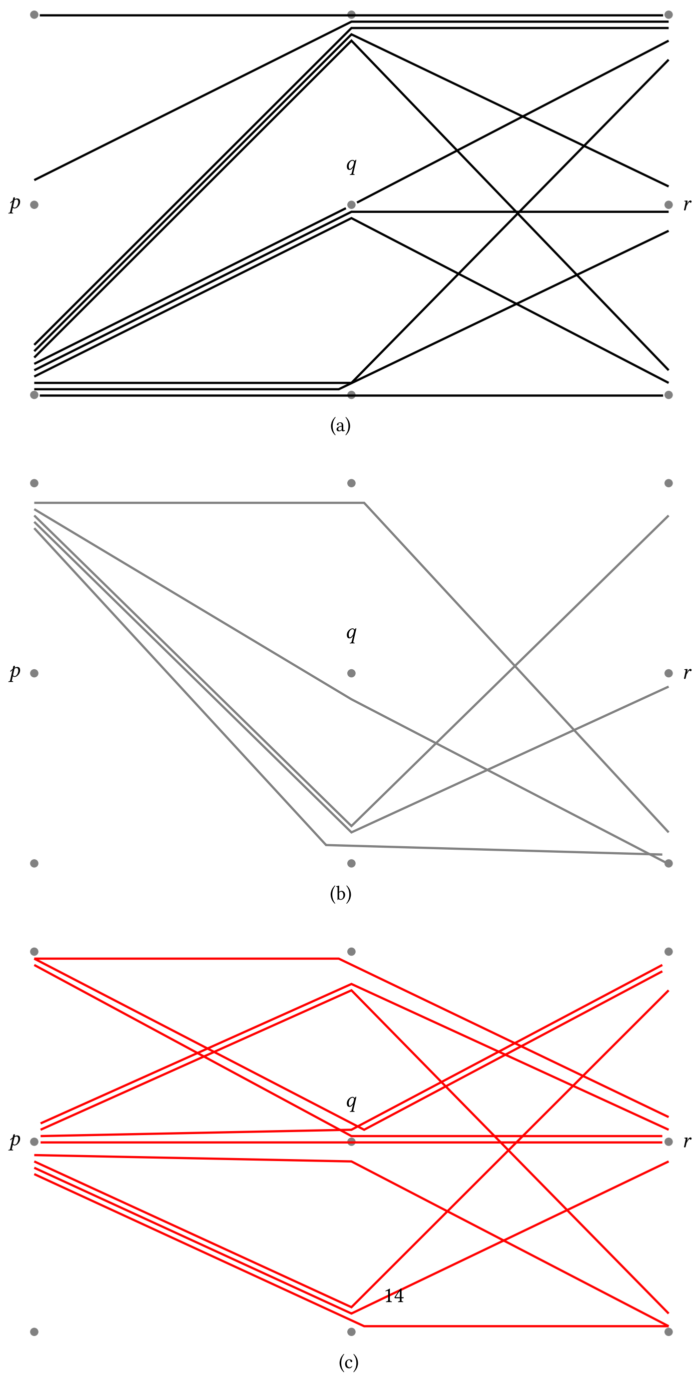



**Figure 12.** Truth diagrams for each truth value for the formula $p \to q \wedge r$ in logic of paradox. (a) Truth diagram for the truth value $T$ for the formula $p \to q \wedge r$ in logic of paradox. (b) Truth diagram for the truth value $F$ for the formula $p \to q \wedge r$ in logic of paradox. (c) Truth diagram for the truth value $P$ for the formula $p \to q \wedge r$ in logic of paradox.

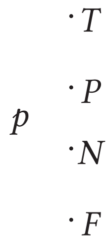



**Figure 13.** Nodes for the truth diagrams in BL and the truth values they represent.

- Letters are propositional *variables*.
- Each node represents a truth-value for the variable: high-node for truth *T*, low-node for false *F* and middle-node for paradoxical *P*.
- Connector style represents the overall truth-value assigned to its case. A black connector assigns *T*, grey connector assigns *F* and red connector assigns *P*.

The definition above can easily be extended to various other multi-valued logics which we leave to reader.

The correctness of TDs for LP is driven from the truth table of LP, given in Figure 4. LP is a three-valued and a functional logic (Priest, 2008). And these are reflected directly in Definitions 5.1 and 5.2.

It is important to note how TDs have got complex due to the exponential nature of the truth value combinations. Using a relatively simple truth table of LP, expressing ternary formulas have produced complex diagrams, given in Figures 11 and 12. TDs for classical propositional logic are simplified cases, therefore much easier to represent and much easier to understand visually. Once the number of possible truth values increase, the said advantages of TDs begin to weaken. LP is a good example for it and we will observe next how TDs behave in the case of four truth values. This shows a disadvantage of TDs – for multi-valued logics, we need many colours and many connectors, which eventually complicates the graphical elements of TDs.

### *5.2. Truth diagrams for Belnap–Dunn's four valued system*

Truth diagrams for BL require four colours to represent four truth values. Similar to high and low nodes in classical propositional logic, in B4 we will use four nodes. High node will be for truth *T*, low node will be for falsity *F*, under-middle node will be for under-valuation *N*, and the lower-middle node will be for over-valuation *P* (Figures 13 and 14).

The reason why the truth values are 'stacked' on top of each other is to generate a TD that is easy to read and compose .[3] This is not necessarily the only way to represent multiple truth values. Alternatively, placing the truth value nodes around the propositional letters with different angular positions can also be considered. For example, one may argue that the top position goes to *T*, bottom to *F*, left to *N* and right to *P*. However, this would make TDs very difficult to read and compose visually, and TDs constructed in this way may fail to capture the graphical essence of classical TDs .[4]

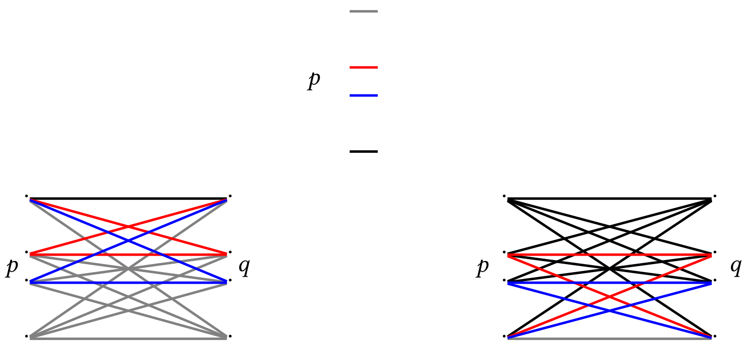



**Figure 14.** Truth diagrams for negation, conjunction and disjunction in BL, respectively, where the red connectors represent the over-valuation truth value *P* and the blue connectors represent the under-valuation truth value *N*.

Connectors of relevant colours will be introduced based on the truth table for BL in Figure 5. In addition to black and grey, we will use red and blue to represent the over-valuation truth value *P* and the under-valuation truth value *N*, respectively.

Now, we can define TDs for BL formally, *only* emphasising the differences in the definitions for BL. Let us start with the elements of TDs for BL.

**Definition 5.3:** Elements of truth diagrams for Belnap-Dunn's four valued logic are given as follows.

- A TD is composed of *letters*, *nodes* and *connectors*.
- Letters are arranged horizontally (with regular spacing for readability).
- Nodes are four small areas, one above, one in the upper-middle of, one in the lower-middle of and one below the letters.
- One connector for each possible combination of high, upper-middle, lower-middle or low nodes of each letter is permitted.
- The style of the connectors is solid and either black, grey, blue or red.

The semantics of TDs for BL is defined as follows.

**Definition 5.4:** Semantics of truth diagrams for Belnap-Dunn's four valued logic is defined as follows.

- Letters are propositional *variables*.
- Each node represents a truth-value for the variable: high-node for truth *T*, low-node for false *F*, upper-middle for over-valuation *P* and lower-middle for undervaluation *N*.
- Connector style represents the overall truth-value assigned to its case. A black connector assigns *T*, grey connector assigns *F*, red connector assigns *P* and blue connector assigns *N*.

In BL, many different logical conditionals can be defined with different truth tables and with different philosophical motivations. This may complicate the truth diagrams unnecessarily, and in order to keep the logic side simpler, let us focus on a TD for the formula $p \wedge (q \vee r)$. This formula has three variables which requires 64 connectors. The TD for the aforementioned formula is presented in Figure 15 by using subdiagrams for each truth value for ease in reading. We leave it to the reader to determine how easy TDs make the full and complete diagram to read and understand in BL, compared to a 64-row truth table for the very same formula.

The correctness of TD for BL follows the same argument we presented for TDs for LP, hence skipped.

### 5.3. Truth diagrams for MacColl's connexive logic

Truth diagrams for CC require four colours to represent four truth values, similar to BL. High node will be for $T$, low node will be for $F$, upper-middle node will be for $t$ and the lower-middle node will be for $f$ (Figure 16).

Connectors of relevant colours will be introduced based on the truth table for CC in Figure 7. In addition to black and grey, we will use red and blue to represent the truth values $t$ and $f$, respectively.

Similarly, our example $p \rightarrow q \wedge r$ and $p \wedge (q \vee r)$ have three variables which require 64 connectors. This makes it difficult to read the TD in one diagram, hence it is left to the reader in order to economise the length of the paper. The diagrammatic behaviour of the connectives in CC is given in Figure 17.

Let us now define TDs for CC formally, *only* emphasising the differences in the definitions for CC. We start with the elements of TDs for CC.

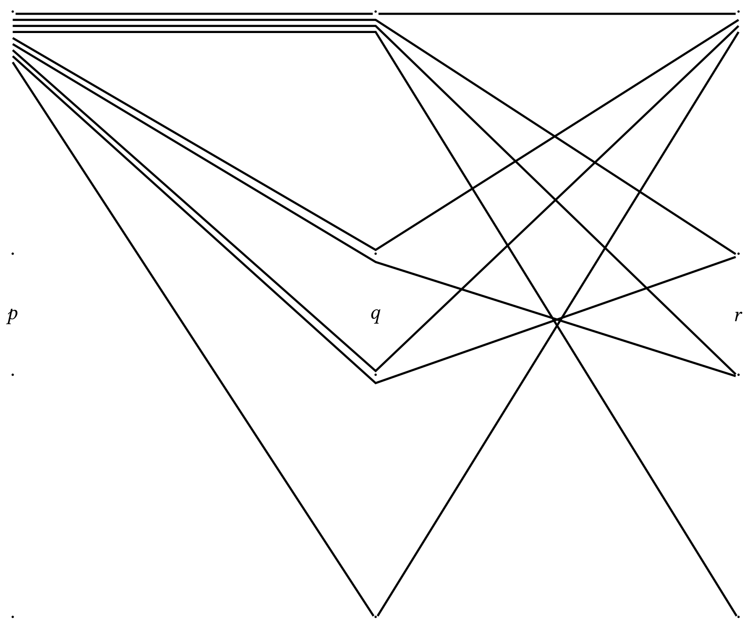
**Definition 5.5:** Elements of truth diagrams for MacColl's connexive logic are given as follows.

- A TD is composed of *letters*, *nodes* and *connectors*.
- Letters are arranged horizontally (with regular spacing for readability).
- Nodes are four small areas, one above, one in the upper-middle of, one in the lower-middle of and one below the letters.
- One connector for each possible combination of high, upper-middle, lower-middle or low nodes of each letter is permitted.
- The style of the connectors is solid and either black, grey, blue or red.

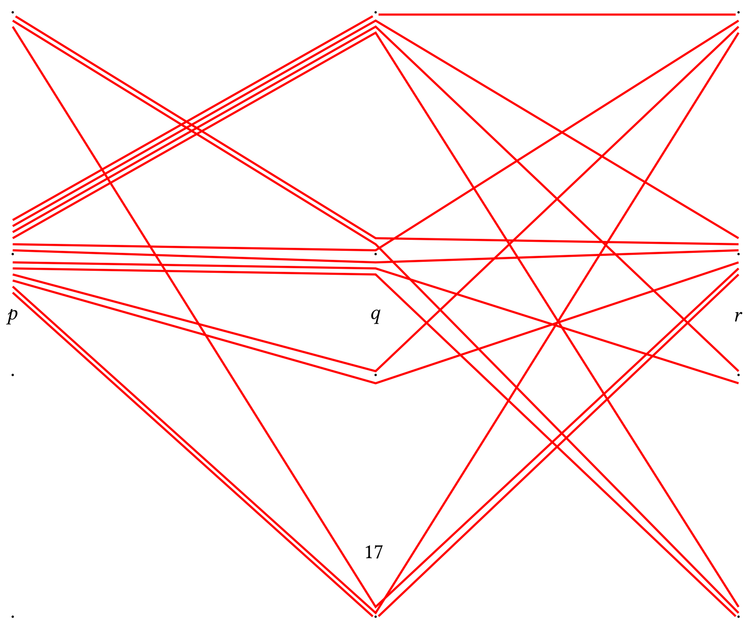
The semantics of TDs for CC is defined as follows.

**Definition 5.6:** Semantics of truth diagrams for MacColl's connexive logic is defined as follows.

- Letters are propositional *variables*.
- Each node represents a truth-value for the variable: high-node for logical necessity $T$, low-node for logical impossibility $F$, upper-middle for contingent truth $t$ and lower-middle for contingent falsehood $f$.




(a)




(b)

**Figure 15.** Truth diagram for the formula $p \wedge (q \vee r)$ in BL. (a) Truth diagram for the truth value $T$ for the formula $p \wedge (q \vee r)$ in BL. (b) Truth diagram for the truth value $P$ for the formula $p \wedge (q \vee r)$ in BL. (c) Truth diagram for the truth value $N$ for the formula $p \wedge (q \vee r)$ in BL. (d) Truth diagram for the truth value $F$ for the formula $p \wedge (q \vee r)$ in BL.

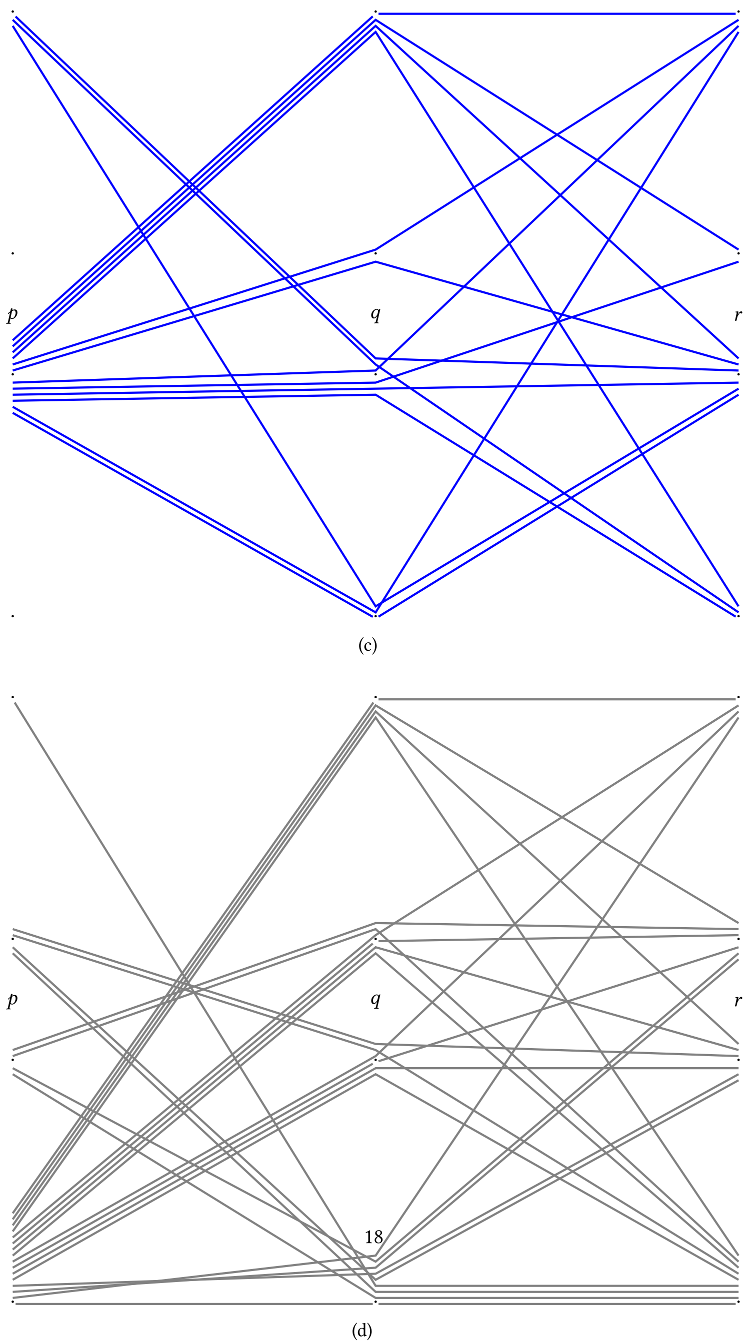



**Figure 15.** Continued.

- Connector style represents the overall truth-value assigned to its case. A black connector assigns *T*, grey connector assigns *F*, red connector assigns *t* and blue connector assigns *f*.

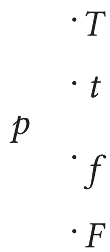



**Figure 16.** Nodes for the truth diagrams in CC and the truth values they represent.

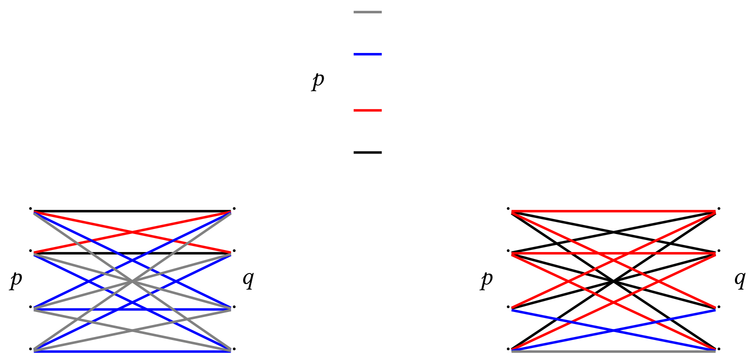



**Figure 17.** Truth diagrams for negation, conjunction and disjunction in CC, respectively, where the red connectors represent *t* and the blue connectors represent *f*.

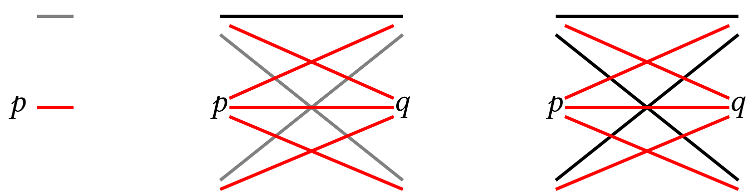



**Figure 18.** Truth diagrams for negation, conjunction and disjunction in BH3, respectively, where the red colour represents the nonsensical truth value *N*.

### *5.4. Truth diagrams for Bochvar–Halldén's logic of non-sense*

Truth diagrams for BH3 will use the same methodology as LP as both are three-valued systems. Similar to TD for LP, we will use red connectors to represent the nonsensical truth value *N*. The TDs for the main Boolean connectives are given in Figure 18.

In what follows, we define TDs for BH3 more formally, *only* emphasising the differences in the definitions for the logic of paradox. Let us start with the elements of TDs for BH3.

**Definition 5.7:** Elements of truth diagrams for Bochvar–Halldén's logic of non-sense are given as follows.

- A TD is composed of *letters*, *nodes* and *connectors*.

- Letters are arranged horizontally (with regular spacing for readability).
- Nodes are three small areas, one above, one in the middle of and one below the letters.
- One connector for each possible combination of high, middle or low nodes of each letter is permitted.
- The style of the connectors is solid and either black, grey or red.

The semantics of TDs for BH3 is defined as follows.

**Definition 5.8:** Semantics of truth diagrams for Bochvar–Halldén's logic of non-sense is defined as follows.

- Letters are propositional *variables*.
- Each node represents a truth-value for the variable: high-node for truth $T$, low-node for false $F$ and middle-node for non-sense $N$.
- Connector style represents the overall truth-value assigned to its case. A black connector assigns $T$, grey connector assigns $F$ and red connector assigns $N$.

For BH3, let us consider the formula $p \wedge (q \vee r)$ in BH3, given in Figure 19. For simplicity in reading, we give the TD in three sub-diagrams.

The *dominance* of certain truth values are important to note here (Başkent, 2020). In none of the diagrams for black and grey, that is for true and false, connectors pass through the node for $N$ the non-sense truth value. It is because $N$ is a dominant truth value: if any of the subformulas of a given formula in BH3 has $N$ truth value, it will propagate and render the whole formula $N$. So, a connector cannot maintain its grey or black colour if they go through an $N$ node. That is the reason why the TD for $N$ (red one) is very crowded compared to the other ones. In fact, it is easy to compute. We have $3^3 - 2^3 = 19$ red connectors.

Such observations can be advanced to similar logics of non-sense that are constructed in the same fashion, following the dominance of non-sense truth values (Başkent, 2020; Szmuc, 2016).

### *5.5. Truth diagrams for Carnielli–Coniglio's logic of formal inconsistency*

Truth diagrams for LFI require a method to represent bivaluation. Unlike all the other multi-valued non-classical logics which we hitherto have discussed, LFI is a binary system. However, the semantics of the negation and consistency operators create a problem. Negation of a true formula can be both true and false, and similarly, consistency of a true formula can be both true and false in three different ways, based on the negations of the true formula. The TD for LFI should reflect this. The immediate solution to this problem is to allow more than one connectors at nodes.

Let us start with considering the TD for negation. TD for negation in LFI shows that the negation of falsity is truth, whereas the negation of truth is both truth and falsity. Therefore, for the high nod, we need two connectors: black and grey, representing truth and falsity, respectively. Similarly, for the consistency operator $\circ$, we have three

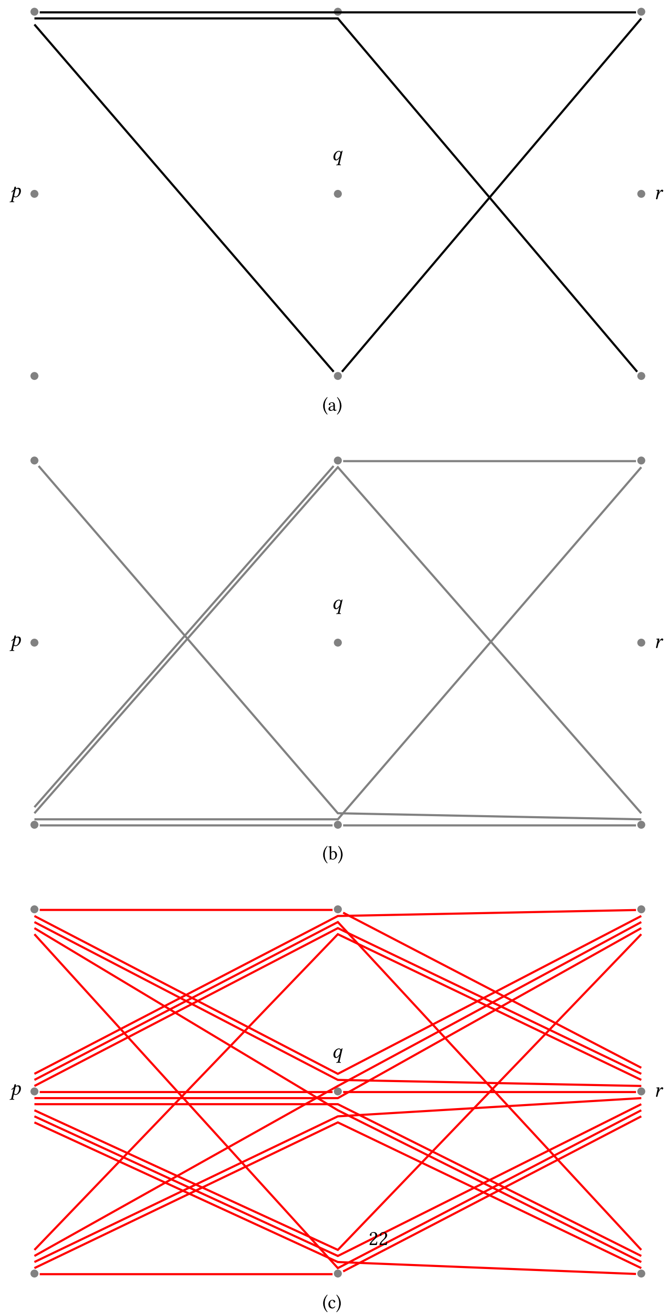



**Figure 19.** Truth diagrams for each truth value for the formula $p \wedge (q \vee r)$ in Bochvar–Halldén's logic. (a) Truth diagram for the truth value $T$ for the formula $p \wedge (q \vee r)$ in Bochvar–Halldén's logic. (b) Truth diagram for the truth value $F$ for the formula $p \wedge (q \vee r)$ in Bochvar–Halldén's logic. (c) Truth diagram for the truth value $N$ for the formula $p \wedge (q \vee r)$ in Bochvar–Halldén's logic.


**Figure 20.** Truth diagrams for negation and the consistency operator in LFI, respectively.

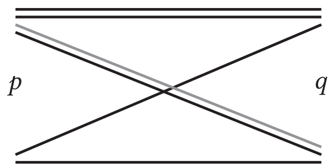



**Figure 21.** Truth Diagram for $p \wedge \neg p \rightarrow q$ in LFI.

cases for $\circ p$ when $p$ is true, and two cases when $p$ is false. TDs for both diagrams are given in Figure 20. TD for the other Booleans connectives in LFI are classical.

Let us see some examples. We start by considering the formula $p \wedge \neg p \rightarrow q$. Based on the truth table for LFI given in Figure 9, we have six different combinations of truth values for $p$, $\neg p$ and $q$: (i) all three are true, (ii) $p$ and $\neg p$ are true but $q$ is false, (iii) $p$ is true $\neg p$ is false, $q$ is true, (iv) $p$ is true $\neg p$ is false, $q$ is false, (v) $p$ is false $\neg p$ is true, $q$ is true, and (vi) $p$ is false $\neg p$ is true, $q$ is false. The cases (i) and (ii) are the non-classical ones. Hence, the TD for LFI requires six connectors. And the non-classical cases will require parallel connectors to the classical ones. The connector for case (i) will run in parallel to that of case (iii), and the connector for case (ii) will run in parallel to that of case (iv). The TD for the formula $p \wedge \neg p \rightarrow q$ is given in Figure 21.

Examples can be multiplied in a similar fashion.

Now, we define TDs for LFI more formally, *only* emphasising the differences in the definitions for the logic of paradox. Let us start with the elements of TDs for LFI.

**Definition 5.9:** Elements of truth diagrams for Carnielli–Coniglio's logic of formal inconsistency are given as follows.

- A TD is composed of *letters*, *nodes* and *connectors*.
- Letters are arranged horizontally (with regular spacing for readability).
- Nodes are two small areas, one above and one below the letters.
- More than one connector for each possible combinations of high or low nodes of each letter is permitted.
- The style of the connectors is solid and either black or grey.

It is important to note that in LFI we allow more than one connector for each possible combination of high or low nodes.

The semantics of TDs for LFI is defined as follows.

**Definition 5.10:** Semantics of truth diagrams for Carnielli–Coniglio's logic of formal inconsistency is defined as follows.

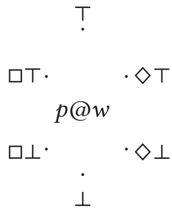



**Figure 22.** Nodes for the truth diagrams in modal logic and the truth values they represent at state $w$.

- Letters are propositional *variables*.
- Each node represents a truth-value for the variable: high-node for truth $T$ and low-node for false $F$.
- Connector style represents the overall truth-value assigned to its case. A black connector assigns $T$ and grey connector assigns $F$.

As argued before, the above definition allows more than one connector to connect nodes, reflecting the truth table of LFI given in Figure 9.

## 6. Truth diagrams for modal logics

In this section we start with presenting TDs for S5 and later on extend them to basic modal logic and public announcement logic. One of the goals of this section is to underline the theoretical richness of the connections between TDs and various modal logics. Whilst doing so, we will not necessarily limit ourselves to classical modal logical way of reasoning. Not every modal logic is necessarily classic nor normal, not every modal logic has duality between $\Box$ and $\Diamond$ operators. Therefore, we define TDs to cover the broadest possible class of modal logics even if our focus in this section is S5 modal logic, which is a relatively straight-forward classical modal logic. We will take advantage of its simplicity when we construct TDs.

Modal logics give an 'internal, local perspective on relational structures' (Blackburn et al., 2001). 'Localities' change the way we evaluate truth in modal logic. Consequently, there are few possibilities of truth for a formula in modal logic. It is either (i) true, (ii) false, (iii) true everywhere, (iv) true somewhere, (v) not true everywhere, and (vi) not true somewhere. Moreover, the truth of each formula depends on a state $w$. All these need to be incorporated into TDs. Therefore, we need four additional nodes around letters: top (that is 12 o'clock) for 'true', bottom (that is 6 o'clock) for 'false', left-upper-middle (that is 10 o'clock) for 'necessarily true' or 'true everywhere', right-upper-middle (that is 2 o'clock) for 'possibly true' or 'true somewhere', left-lower-middle for (that is 8 o'clock) 'necessarily false' or 'not true somewhere' and right-lower-middle (that is 4 o'clock) for 'possibly false' or 'not true everywhere'. Similarly, we associate the truth of a propositional letter $p$ with a state $w$ and write it as $p@w$ in the TD. We summarise our ideas in Figure 22 .[5]

The truth in S5 is the truth in a model of equivalence classes. Now, let us, for example, consider the formula $\Box p \wedge q$ at state $w$ in S5, as given in Figure 23. In S5, the formula is true when $\Box p$ and $q$ are true. Furthermore, due the characteristics of S5 – that is

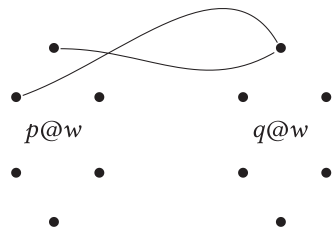



**Figure 23.** Truth diagrams for $\Box p \wedge q$ at state $w$ in S5. Grey connectors are omitted for ease in reading.

reflexivity–, when $\Box p \wedge q$ is satisfied, so are $p$ and $q$. In both cases, we omit the grey connectors representing falsity for ease in reading.

The classical modal logic S5 may appear to be a simple case for TDs. Developing a TD for modal logics in general proves to be a difficult task for a variety of reasons. First, without actually drawing the picture of the modal model, it is difficult to express the recursiveness of modal formulas in TDs. The depth of modal formulas is another difficulty, where depth is defined as the highest number of modal operators appearing in any subformula of a given formula. The formula $\Box p \wedge q$ of depth 1 can perhaps be expressed in a TD for K, but the formula $\Box\Diamond\Box\Diamond p \wedge q$ of depth 4 poses some challenges due to the recursiveness of the truth semantics of modal logic.

Let us now start with the easy case and define TDs for modal logic S5 formulas of depth 1 formally – first the elements then the semantics.

**Definition 6.1:** Elements of truth diagrams for modal logic S5 for formulas of depth 1 are given as follows.

- A TD is composed of *letters*, *nodes*, *connectors* and *states* attached to letters.
- Letters (with attached states) are arranged horizontally (with regular spacing for readability).
- Nodes are six small areas, one above, one in the upper-left-middle of, one in the lower-left-middle of, one in the upper-right-middle of, one in lower-right middle of and and one below the letters.
- More than one connector for any possible combination of nodes of each letter is permitted.
- Letters are arranged with states.
- The style of the connectors is solid and either black or grey.

The semantics of TDs for modal logic is defined as follows.

**Definition 6.2:** Semantics of truth diagrams for modal logic S5 for formulas of depth 1 is defined as follows.

- Letters are propositional *variables* with states attached to them in the form of $p@w$, where $p$ is a letter and $w$ is a state.

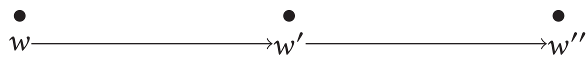



**Figure 24.** The frame $F$.

- Each node represents a truth-value for the variable: high-node for truth $T$, low-node for false $F$, lower-left-middle node for necessarily false $\Box\bot$, upper-left-middle node for necessarily true $\Box\top$, upper-right-middle node for possibly true $\Diamond\top$ and lower-right middle node for possibly false $\Diamond\bot$.
- Connector style represents the overall truth-value assigned to its case. A black connector assigns $T$ and grey connector assigns $F$.

The TDs presented in this section take advantage of the structural properties of S5 and leave room for potential non-classical elements in modal logics, such as lack of duality between $\Box$ and $\Diamond$ formulas.

### *6.1. Towards TDs for basic modal logics*

In this section, we explore how we can generalise the ideas presented in the previous section to capture the TDs for basic modal logic K.

For modal formulas with higher depth in modal logic K, we can proceed step by step. Let us start with an example. Consider the formula $\Box\neg\Diamond p \wedge q$ with the given modal frame $F = (W, R)$ where $W = \{w, w', w''\}$ and $R = \{(w, w'), (w', w'')\}$. Modal operators bind stronger, so there is no need for parentheses. The frame $F$ is given as above (Figure 24).

Let us consider the formula $\Box\Diamond p$ at state $w$. The truth of this formula relies on the truth of $\neg\Diamond p$ at $w'$ whose truth relies on truth of $p$ at $w''$. Therefore, a TD for $\Box\neg\Diamond p \wedge q$ at $w$ should reflect that.

It is possible to reflect the recursiveness of Kripkean semantics of modal formulas by using a *tower* of TDs for modal formulas.

Here is how to construct a tower of TDs for modal logics. First, using red, we draw the frame. At each state $w, w', w''$ of the frame, we construct a truth diagram, directly reflecting the model theory of modal logic. For different variables $p, q, \dots$ at each state we will write them in parallel with sufficient space. Vertically, we will construct the the frame as an upside tree with a root. This will allow us to recursively trace and compute the truth of the modal formulas at any state of the frame using TDs.

For example, let us construct a tower of TDs for the formula $\Box\neg\Diamond p \wedge q$ at frame $F$ as given above. At state $w$, the formula $\Box\neg\Diamond p$ and $q$ must be satisfied in order to make the formula true. However, $\Box\neg\Diamond p$ being satisfied at $w$ means that, using the information from frame $F$, the formula $\neg\Diamond p$ must be true at $w'$. Similarly, $p$ must fail at $w''$. All these conditions, for frame $F$, need to be satisfied for the given formula $\Box\neg\Diamond p \wedge q$ at state $w$. Now, we connect the satisfiable nodes and leave the grey connectors out for ease in reading. The modal TD for the formula $\Box\neg\Diamond p \wedge q$ in frame $F$ is given in Figure 25.

The purpose of having six nodes around letters and formulas can clearly be seen here as they provide more information in the diagram, and makes it easier for the readers to follow the recursive truth conditions of modal logic. For example, we can read off

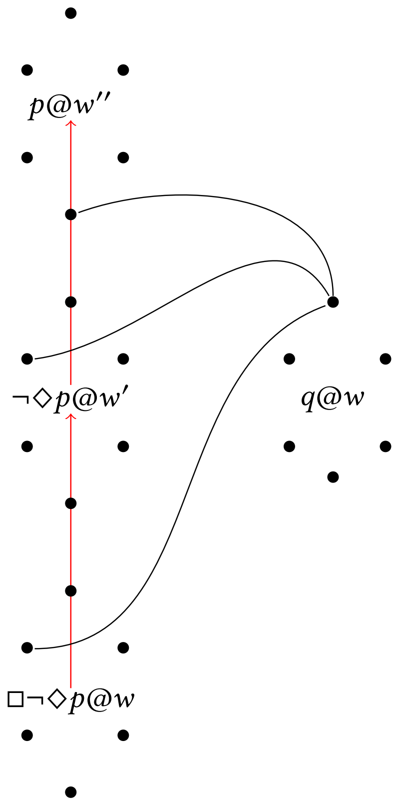



**Figure 25.** Truth Diagrams for $\Box\neg\Diamond p \wedge q$ at state $w$. Grey connectors are omitted for ease in reading.

the given formula $\Box\neg\Diamond p$ at $w$ from the TD in Figure 25. For the coloumn in TD for $p$, we can read off from the bottom. The bottom diagram for $p$ suggest that the formula is a $\Box$-formula, followed by a $\Box\neg$ formula, followed by a false $p$. Putting them together we obtain $\Box\Box\neg p$ which is equivalent to $\Box\neg\Diamond p$. Surely, taking advantage of classical duality of $\Box$ and $\Box$ formulas, the TD for modal logic can further be simplified .[6]

As the example given in Figure 25 illustrates, TDs for modal logics need to combine propositional letters, modal formulas with no binary connectives, the relational structure and the truth. TD towers achieve this.

Now, we can define modal TDs.

**Definition 6.3:** Elements of truth diagrams for modal logic are given as follows.

- A TD is composed of *letters*, *letters with modal operators in front* (i.e modal letters), *nodes*, *connectors*, and *states* attached to letters and modal letters.
- Letters (with attached states) are arranged horizontally (with regular spacing for readability).
- Modal letters (with attached states) are arranged vertically (with regular spacing for readability), forming *towers*. Vertical arrangement starts with the given formula and goes vertically up by removing one modal operator at each stage.

- Nodes are six small areas, one above, one in the upper-left-middle of, one in the lower-left-middle of, one in the upper-right-middle of, one in lower-right middle of and one below the letters.
- More than one connector for any possible combination of nodes of each letter is permitted.
- Letters, modal or otherwise, are arranged with states.
- The style of the connectors is solid and either black or grey.
- The frame is included in the diagram using the colour red. The frame is represented as an upside-down rooted tree.
- TDs for modal formulas with their associated states are placed on the modal frame in appropriate nodes.

The semantics of TDs for modal logic is defined as follows.

**Definition 6.4:** Semantics of truth diagrams for modal logic is defined as follows.

- Letters are propositional *variables* with states attached to them in the form of $p@w$, where $p$ is a letter and $w$ is a state.
- Modal letters are propositional *modal formulas* with states attached to them in the form of $\Box^n\Diamond^m p@w$, where $p$ is a letter and $w$ is a state.
- Each node represents a truth-value for the variable: high-node for truth $T$, low-node for false $F$, lower-left-middle node for necessarily false $\Box\bot$, upper-left-middle node for necessarily true $\Box\top$, upper-right-middle node for possibly true $\Diamond\top$ and lower-right middle node for possibly false $\Diamond\bot$.
- Connector style represents the overall truth-value assigned to its case. A black connector assigns $T$ and grey connector assigns $F$.

For clarity, this is how we define $\Box^n$ recursively, for $n, m \in \mathcal{N}$.

$$\Box^0 p = p$$
$$\Box^{n+1} p = \Box\Box^n p$$

The same goes for the $\Diamond^m$ operator.

The above definition is a first for modal logic. What is left is to generalise to a broader class of modal logics and explain how various modal axioms relate to TDs and how different frames may generate different TDs and towers. Along the same lines, it can be perceived to simplify our modal TDs by just using binary nodes for truth and falsity. Such extensions and simplifications are left for future work.

Modal logics can be complex, so are the TDs for them. The current work raises a number of important questions regarding TDs for modal logic. Particularly,

(1) What is the diagrammatic representation for modal logical equivalence?
(2) What is the systematic way to diagrammatically represent or simplify different modal logics such as T, S4 etc.?
(3) What is the most efficient way to represent the modal logical frames, that is the red lined diagram?

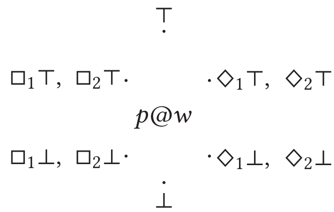



**Figure 26.** Nodes for the truth diagrams in dynamic epistemic logic for two agents, and the truth values they represent.

(4) How can we make use of TDs for modal logic for the proof theory of modal logic?
(5) How can we give TDs for first-order classical modal logics?

Each of these questions deserve an individual treatment, thus left for future work.

### *6.2. Truth diagrams for public announcement logic*

In this section, as a case study, we discuss TDs for public announcement logic PAL5.

Let us first consider a model for two agents. In this case, we need nodes and connectors for each agent. The nodes in the classical case will double as the nodes for both agents to identify the modal truth conditions as given in Figure 26. The connectors, however, need to be distinguished between the agents. For Agent 1, we will use continuous-line, whereas for Agent 2 we will use dash-line connectors. Positions of truth around a propositional letter remain as before.

Here is a toy example for agents 1 and 2. Assume at $w$, $q$ holds and 1 knows $p$. Similarly, at $w$, $p$ holds and 2 knows $p$. Therefore, we have

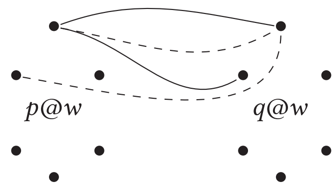
$$w \models p \wedge \Box_1 q \qquad\qquad w \models q \wedge \Box_2 p$$

Now, assume that a truthful announcement of $\neg\Box_1 q$ is made, suggesting agent 1 does not indeed know $q$. Consequently, the model is eliminated by removing the states where the announcement does not hold.

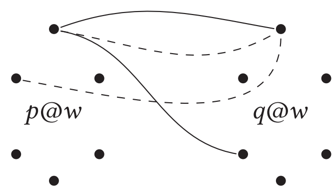
Let us now see how the update is reflected in the TD. First, for simplicity, let us denote agent 2's knowledge situation with dashed connectors and omit the grey connectors which represent falsity. After the announcement, agents get to learn that $\neg\Box_1 q$ is indeed true. This introduces the node for it to the diagram with a black connector. The way the model is updated is represented in Figure 27 which clearly shows the change in black connectors.

It is also important to note that after an update some states can be removed from the model thus from the TD. Following up from the model above, if, for example, the formula $\neg p$ is announced, the state $w$ would be removed from the model as a result of the update. This reminds us that the modal semantics is local and TDs cannot provide a global picture of the modal models. All they can do is to give the semantics of propositional variables at a particular state.

Let us now define TDs for public announcement logic S5 for two agents formally – first the elements then the semantics.




(a)




(b)

**Figure 27.** Truth Diagrams for $w \models p \wedge \Box_1 q$ and $w \models q \wedge \Box_2 p$ at $w$ in a two-agent dynamic epistemic model, before and after the announcement of $\neg\Box_1 q$, in S5$_n$. Grey connectors are omitted for ease in reading. (a) The original epistemic model. (b) After an update by $\neg\Box_1 q$.

**Definition 6.5:** Elements of truth diagrams for public announcement logic S5 for two agents are given as follows.

- A TD is composed of *letters*, *nodes*, *connectors* and *states* attached to letters.
- Letters (with attached states) are arranged horizontally (with regular spacing for readability).
- Nodes are six small areas, one above, one in the upper-left-middle of, one in the lower-left-middle of, one in the upper-right-middle of, one in lower-right middle of and and one below the letters.
- More than one connector for any possible combination of nodes of each letter is permitted.
- Letters are arranged with states.
- The style of the connectors is solid black or solid grey for agent 1, dashed black or dashed grey for agent 2.

The semantics of TDs for modal logic is defined as follows.

**Definition 6.6:** Semantics of truth diagrams for public announcement logic S5 for two agents is defined as follows.

- Letters are propositional *variables* with states attached to them in the form of *p@w*, where *p* is a letter and *w* is a state.
- Each node represents a truth-value for the variable: high-node for truth *T*, low-node for false *F*, lower-left-middle node for necessarily false $\Box_i\bot$, upper-left-middle node for necessarily true $\Box_i\top$, upper-right-middle node for possibly true $\Diamond_i\top$ and lower-right middle node for possibly false $\Diamond_i\bot$ where $i \in \{1,2\}$.

- Connector style represents the overall truth-value assigned to its case. A black connector (dashed or solid) assigns *T* and grey connector (dashed or solid) assigns *F*.

The above definition describes a TD for a dynamic logic for two agents. The structure we describe makes it possible to extended to systems with more than two agents.

A plethora of opportunities exist to extend our introductory TD to various many-agent dynamic modal systems and methodologies. Such extensions and offering a complete catalogue of TDs for dynamic modal logics fall outside the scope of current work.

## 7. Discussion and conclusion

Broadly, there are two ways to understand the relation between logic and TDs. 'Logic to diagrams' direction attempts at developing diagrammatic and visual tools to express the semantics of a given logic. As such, it asks the following question: 'What is the TD for this logic?' Conversely, 'diagrams to logic' direction investigates the diagrammatic elements found in TDs and seek their logical interpretation. As such, it asks the following question: 'Given an imaginary TD, what is the logic whose semantics can diagrammatically be explained by it?' The current paper follows the 'logic to diagrams' direction and generates a variety of TDs for a broad class of logics. As such, it offers a new diagrammatic semantics for various non-classical logics. Moreover, TDs hint out some logical properties as well. For instance, considering the TDs discussed in Figures 2 and 3, we can see that only two colours are allowed – because the classical logic has two truth values. The connectors lines are always horizontal – because vertical lines can be used to express contradictions which are not allowed in classical logics. In BH3, for example, we can see how the 'hierarchy' of truth values can visually be expressed in TDs.

### *7.1. What we learn from TDs for non-classical logics*

Arguably, TDs make it easier to trace what makes a formula to have a certain truth value. For example, consider the TD given in Figure 15. For example, if one needs to ensure how to render the given formula $p \wedge (q \vee r)$ false, it is then relatively easier and quicker to observe what makes it false. For example, once $p$ is false the formula always ends up false. Similarly, it is impossible to make the formula *P* when $p$ is *N* or *F*. TDs can be (cognitively) quicker to work with in such circumstances.

Modal logic is an interesting and central case. Depending on the axiomatisation, different modal logics can require different TDs. For example, in classical cases, due to modal duality ($\Box\varphi \equiv \neg\Diamond\neg\varphi$), nodes can be simplified. In some modal logics, where negations are defined non-classically as $\neg\Box$, for example, further simplifications can be made. This is the reason why we introduce a broader framework for TDs that is sufficiently expressive for modal logics. Modal logics are manifold and rich in their mathematical structures, TDs should be able to cater for them. Our study of modal TDs shows how fragile truth is in modal logic and how complicated it can be to represent it diagrammatically.

### 7.2. Correctness

Our work introduces an alternative semantics for various propositional non-classical logic. Apart from modal logic, TDs are alternatives to truth tables.

The correctness of TDs carry over from the truth tables and model theories of the aforementioned logics. Our definitions of TDs for each logic allow us to represent each and every truth table given for such logics in Section 4. Therefore, TDs are correct and adequate for the non-classical logics we have discussed as they cover all logical operators and represent each and every row in the truth tables of aforementioned logics.

Modal semantics, on the other hand, follow the methodology of Kripkean semantics and unfold complex modal formulas step-by-step. This offers an interesting way to deal with modal depth of formulas.

### 7.3. What this work is about

This work is complementary to research on the semantics of non-classical logics as well as modal and dynamic logics. An earlier body of work discussed game semantics for non-classical logics (Başkent, 2016, 2020; Başkent & Henrique Carrasqueira, 2020) in a systematic fashion, and the current work achieves the same using truth diagrams. A similar trajectory for dynamic logic was carried out using topological semantics (Başkent, 2012). Together with the current work, this line of research is focusing on 'non-classical' truth semantics for non-classical logics.

Moreover, the current work provides a case study for those interested in diagrammatic reasoning in logic. The way the diagrams themselves work and interact, and the way in which such systems can be generated serve logic directly – now we can simply work on diagrams, perhaps create some diagrammatic systems, *and then* study their logic.

The current work can also be seen as a proof that TDs are genuine mathematical diagrams, according to de Toffoli's criteria (de Toffoli, 2022): (i) both human and artificial agents can easily and effectively produce a TD, (ii) TDs' features are easily indentifiable and they carry logical content reliably, and (iii) they correspond to well-defined mathematical operations – that is truth functions in non-classical logics. Briefly, we have tested de Toffoli's criteria for mathematical diagrams using non-classical logics in the context of TDs successfully.

Unsurprisingly, it is also important to visually identify the difficulties that multi-valued logics introduce to the study of TDs. In many cases (see Figures 12 and 15), we had to separate the TDs into several sub-diagrams to improve readability. This is more fundamental than it seems as it may practically make it impossible to make a diagram for, for instance, a 4-valued logic using 5 variables as it requires 1024 connectors. As such, arguably, it is no easier or transparent than constructing a truth table. And, the clarity and ease that TDs present with the classical logic completely vanish.

### 7.4. What this work is not about

This work attempts at introducing new truth semantics for various logics. It does not attempt at discussing their proof theory or how TDs may help us improving it.

Furthermore, this paper is not on the *philosophy* of TDs except for being a showcase of how non-classical logics may have an impact first on TDs and then on diagrammatic reasoning.

This paper is not about the semantics of proofs or syntax. Use of TDs in proof theory of non-classical logics remains an open question. Our work does not discuss syntax neither, and therefore does not distinguish the TDs of, say, $p \wedge q$ and $q \wedge p$.

One interesting issue that we learned whilst working on this paper is the challenge of actually producing TDs. Since all the logics considered are finite, the process of constructing TDs is computationally effective. Making the diagrams easy to read and follow, on the other hand, is not as easy for us mere humans. That is an issue we have not touched upon in this paper – how to actually produce the TDs in an 'easy to read and follow' fashion, and how to automate this process. We believe this is an interesting direction to pursue, especially for multi-valued logics.

### 7.5. Future work ideas

An immediate future work area is the proof theory of the non-classical logics which we have discussed, using TDs.

Dynamic modal logic and its cousins constitute a lively and popular research area. A broad spectrum of logical systems in this area, including AGM belief revision to gossip models, may directly benefit from TDs. Similarly, bisimulations and process equivalences in modal logic can benefit from TDs. The way they work can be given a visual representation using TDs.

Along the same lines, another line of inquiry is to relate TDs to alternative semantics for modal logics. Topological semantics for modal and dynamic logics is an interesting research direction (Başkent, 2012), and the way that TDs can be translated to topologies remains an open question. Similarly, TDs and *non*-normal modal logics remain an unexplored direction.

Introducing avant-garde semantic structures for non-classical logics is not a new endeavour. Hintikkan game semantics can be viewed as a similar approach. It is thus important to identify what game semantics and truth diagrams can learn from each other. First, it is worth exploring how TDs can possibly help developing strategies for players in a semantic games. Similarly, for a given TD for a logic, one can ask if it is possible to develop a semantic game for the logic in question. Such issues relate TDs to games and suggest an interesting research direction.

Definitions 3.1 and 3.2 describe how diagrams are constructed. Ignoring the logical aspects of it, one can easily work on alternating the given definitions. For example, one can allow connectors to intersect with propositional letters in more than one place. This can also help to develop ideas for the first-order extension of TDs and reinforce the 'diagrams to logic' direction.

## Notes

1. It is often thought that why a 'claim, method or proof' is correct can be given a mathematical, geometric or philosophical arguments. Without proof, I tend to think that one can add 'game theoretical' arguments to the aforementioned bunch.

2. Thanks to the anonymous referee for suggesting this.
3. Thanks to the anonymous referee for suggesting it.
4. Perhaps with some exceptions, as we shall see in due course.
5. Thanks to Peter C.-H. Cheng for suggesting to use the clock terminology.
6. Since our starting point is non-classical logics, it is important to leave sufficient room in TDs to be able to express non-classical modal logics, too. Cheng's approach in Cheng (2020), on the other hand, starts from classical logic and requires extensions as discussed in this work.

## Acknowledgments

I am thankful to Peter C.-H. Cheng for his feedback on the earlier drafts of this work and the anonymous referees for their constructive feedback which improved the presentation of the paper.

## Disclosure statement

No potential conflict of interest was reported by the author(s).

## ORCID

*Can Başkent* http://orcid.org/0000-0001-6229-8699

## References

Abramsky, S., & McCusker, G. (1999). Game semantics. In U. Berger and H. Schwichtenberg (Eds.), *Computational logic* (pp. 1–55). NATO ASI Series, Vol. 165. Springer.
Awodey, S. (2006). *Category theory*. Oxford University Press.
Baez, J. C., & Lauda, A. D. (2011). A prehistory of *n*-categorical physics. In H. Halvorson (Ed.), *Deep beauty: Understanding the quantum world through mathematical innovation* (pp. 13–128). Cambridge University Press.
Başkent, C. (2012). Public announcement logic in geometric frameworks. *Fundamenta Infomaticae*, *118*(3), 207–223. https://doi.org/10.3233/FI-2012-710
Başkent, C. (2016). Game theoretical semantics for some non-classical logics. *Journal of Applied Non-Classical Logics*, *26*(3), 208–239. https://doi.org/10.1080/11663081.2016.1225488
Başkent, C. (2020). A game theoretical semantics for a logic of nonsense. In J.-F. Raskin & D. Bresolin (Eds.), *Proceedings of the 11th International Symposium on Games, Automata, Logics, and Formal Verification (G&ALF 2020). Electronic Proceedings in Theoretical Computer Science* (Vol. 326, pp. 66–81). Open Publishing Association.
Başkent, C., & Henrique Carrasqueira, P. (2020). A game theoretical semantics for a logic of formal inconsistency. *Logic Journal of the IGPL*, *28*(5), 936–952. https://doi.org/10.1093/jigpal/jzy068
Blackburn, P., de Rijke, M., & de Venema, Y. (2001). *Modal logic*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press.
Bonchi, F., Gadducci, F., Kissinger, A., Sobocinski, P., & Zanasi, F. (2022a). String diagram rewrite theory I: Rewriting with frobenius structure. *Journal of the ACM*, *69*(2), 1–58. https://doi.org/10.1145/3502719
Bonchi, F., Gadducci, F., Kissinger, A., Sobocinski, P., & Zanasi, F. (2022b). String diagram rewrite theory II: Rewriting with symmetric monodial structure. *Mathematical Structures in Computer Science*, *32*(4), 511–541. https://doi.org/10.1017/S0960129522000317
Carnielli, W. A., & Coniglio, M. E. (2016). *Paraconsistent logic: Consistency, contradiction and negation*. Springer.
Carnielli, W. A., Coniglio, M. E., & Marcos, J. (2007). Logics of formal inconsistency. In D. Gabbay & F. Guenthner (Eds.), *Handbook of philosophical logic* (Vol. 14, pp. 15–107). Springer.

Carter, J. (2019). Exploring the fruitfulness of diagrams in mathematics. *Synthese*, *196*(10), 4011–4032. https://doi.org/10.1007/s11229-017-1635-1

Cheng, P. C.-H. (2020). Truth diagrams versus extant notations for propositional logic. *Journal of Logic, Language and Information*, *29*(2), 121–161. https://doi.org/10.1007/s10849-019-09299-y

da Costa, N. C. A., Krause, D., & Bueno, O. (2007). Paraconsistent logics and paraconsistency. In D. Jacquette (Ed.), *Philosophy of logic* (Vol. 5, pp. 655–781). Elsevier.

de Toffoli, S. (2022). What are mathematical diagrams? *Synthese*, *200*(2), 86. https://doi.org/10.1007/s11229-022-03553-w

Giaquinto, M. (2007). *Visual thinking in mathematics*. Oxford University Press.

Giaquinto, M. (2008). Visualizing in mathematics. In P. Mancosu (Ed.), *The philosophy of mathematical practice*. Oxford University Press.

Girard, J.-Y. (1987). Linear logic. *Theoretical Computer Science*, *50*(1), 1–101. https://doi.org/10.1016/0304-3975(87)90045-4

Hughes, D. J. D. (2006). Proofs without syntax. *Annals of Mathematics*, *164*(3), 1065–1076. https://doi.org/10.4007/annals

Macbeth, D. (2014). *Realizing reason*. Oxford University Press.

McCall, S. (1966). Connexive implication. *Journal of Symbolic Logic*, *31*(3), 415–433. https://doi.org/10.2307/2270458

Mortensen, C. (2010). *Inconsistent geometry*. College Publications.

Nelsen, R. B. (1993). *Proofs without words*. The Mathematical Association of America.

Priest, G. (1979). The logic of paradox. *Journal of Philosophical Logic*, *8*(1), 219–241. https://doi.org/10.1007/BF00258428

Priest, G. (2008). *An introdiction to non-classical logic*. Cambridge University Press.

Routley, R., & Montgomery, H. (1968). On systems containing Aristotle's thesis. *The Journal of Symbolic Logic*, *33*(1), 82–96. https://doi.org/10.2307/2270055

Szmuc, D. E. (2016). Defining lfis and lfus in extensions of infectious logics. *Journal of Applied Non-classical Logics*, *26*(4), 286–314. https://doi.org/10.1080/11663081.2017.1290488

van Ditmarsch, H., van der Hoek, W., & Kooi, B. (2007). *Dynamic epistemic logic*. Springer.

Wansing, H. (2015). Connexive logic. In E. N. Zalta (Ed.), *The stanford encyclopedia of philosophy* (fall 2015 ed.). Metaphysics Research Lab, Stanford University.