Abstract.

Models of complex systems are widely used in the physical and social sciences, and the concept of layering, typically building upon graph-theoretic structure, is a common feature. We describe an intuitionistic substructural logic that gives an account of layering. As in bunched systems, the logic includes the usual intuitionistic connectives, together with a non-commutative, non-associative conjunction (used to capture layering) and its associated implications. We give soundness and completeness theorems for labelled tableaux and Hilbert-type systems with respect to a Kripke semantics on graphs. To demonstrate the utility of the logic, we show how to represent a range of systems and security examples, illuminating the relationship between services/policies and the infrastructures/architectures to which they are applied.
1 Introduction

Complex systems can be defined as the field of science that studies, on the one hand, how it is that the behaviour of a system, be it natural or synthetic, derives from the behaviours of its constituent parts and, on the other, how the system interacts with its environment. A commonly employed and highly effective concept that helps to manage the difficulty in conceptualizing and reasoning about complex systems is that of layering: the system is considered to consist of a collection of interconnected layers each of which has a distinct, identifiable role in the system’s operations. Layers can be informational or physical and both kinds may be present in a specific system. In [3, 12], multiple layers are given by multiple relations over a single set of nodes.

We employ three illustrative examples. First, a transport network that uses buses to move people. It has an infrastructure layer (i.e., roads, together with their markings, traffic signals, etc., and buses running to a timetable), and a social layer (i.e., the groupings and movements of people enabled by the bus services). Second, a simple example of the relationship between a security policy and its underlying system architecture. Finally, we consider the security architecture of an organization that operates high- and low-security internal systems as well as providing access to its systems from external mobile devices. These examples illustrate the interplay between services/policies and the architectures/infrastructures to which they are intended to apply.

We give a graph-theoretic definition of layering and provide an associated logic for reasoning about layers. There is very little work in the literature on layering in graphs. Notable exceptions are [9, 18, 17]. Layered graphs are an instance of a general algebraic semantics for the logic. Our approach stands in contrast to our previous work in this area [6, 7] in that the additive component of the bunched logic [16, 11] we employ is intuitionistic, with the consequence that we are able to obtain a tableaux system for the logic together with a completeness theorem for the layered graph semantics. In Section 2, we introduce layered graph semantics and ILGL, the associated intuitionistic layered graph logic. In Section 3, we establish its basic metatheory — the soundness and completeness of ILGL’s tableaux system with respect to layered graph semantics — and, in Section 4, we give an algebraic semantics and a (sound and complete) Hilbert-type proof system for ILGL. In Section 5, we sketch a modal extension of ILGL that is convenient for practical modelling, explaining its theoretical status and developing the three examples mentioned above.

2 Intuitionistic layered graph logic

Layered graph semantics. We begin with a formal, graph-theoretic account of the notion of layering that, we claim, captures the concept as used in complex systems. In this notion, two layers in a directed graph are connected by a specified set of edges, each element of which starts in the upper layer and ends in the lower layer.

Given a directed graph, $\mathcal{G}$, we refer to its vertex set and its edge set by $V(\mathcal{G})$ and $E(\mathcal{G})$ respectively, while its set of subgraphs is denoted $Sg(\mathcal{G})$ with $H \subseteq \mathcal{G}$ iff $H \in Sg(\mathcal{G})$. For a distinguished edge set $\mathcal{E} \subseteq E(\mathcal{G})$, the reachability relation $\sim_{\mathcal{E}}$ on subgraphs of $\mathcal{G}$ is $H \sim_{\mathcal{E}} K$ iff a vertex of $K$ can be reached from a vertex of $H$ by an $\mathcal{E}$-edge.
We then have a composition $G@E$ on subgraphs where $G@E H$ iff $V(G) \cap V(H) = \emptyset$, $G \rightarrow_{E} H$ and $H \rightarrow_{E} G$ (where $\downarrow$ denotes definedness) with output given by the graph union of the two subgraphs and the $E$-edges between them. For a graph $G$, we say it is layered (with respect to $E$) if there exist $H, K$ such that $H@E K \downarrow$ and $G = H@E K$ (see Figure 1). Layering is evidently neither commutative nor associative.

Within a given ambient graph, $\mathcal{G}$, we can identify a specific form of layered structure, called a preordered scaffold, that will facilitate our definition of a model of intuitionistic layered graph logic. Properties of graphs that are inherited by their subgraphs are naturally captured in an intuitionistic logic. This idea is generalized by the structure carried by a preordered scaffold. To set this up, we begin by defining an admissible subgraph set is a subset $X \subseteq Sg(G)$ such that, for all $G, H \in Sg(G)$, if $G@E H \downarrow$, then $G, H \in X$ iff $G@E H \in X$. Then, a preordered scaffold (see Figure 2) is a structure $X = (\mathcal{G}, E, X, \prec)$ such that $\mathcal{G}$ is a graph, $E \subseteq E(\mathcal{G}), X$ an admissible subgraph set, $\prec$ a preorder on $X$. Layers are present if $G@E H \downarrow$ for at least one pair $G, H \in X$.

Note that the scaffold is preordered and we choose a subset of the subgraph set. There are several reasons for these choices. From a modelling perspective, we can look closely at the precise layering structure of the graph that is of interest. In particular, we can avoid degenerate cases of layering. (Note that this is a more general definition of scaffold than that taken in [6, 7], where the structure was less tightly defined.) Technical considerations also come into play. When we restrict to interpreting ILGL on the full subgraph set, it is impossible to perform any composition of models without the worlds (states) proliferating wildly. A similar issue arises during the construction of counter-models from the tableaux system of Section 3, a procedure that is impossible when we are forced to take the full subgraph set as the set of worlds.

Having established the basic semantic structures that are required, we can now set up ILGL. Let Prop be a set of atomic propositions, ranged over by $p$. The set Form of all propositional formulae is generated by the following grammar:

$$\phi ::= p \mid \top \mid \bot \mid \phi \land \phi \mid \phi \lor \phi \mid \phi \rightarrow \phi \mid \phi \rightarrow^* \phi \mid \phi \rightarrow^* \phi$$

The familiar connectives will be interpreted intuitionistically. The non-commutative, non-associative conjunction, $\rightarrow^*$, which will be used to capture layering, is interpreted intuitionistically, as in BI [16, 11], and has associated right ($\rightarrow^*$) and left ($\rightarrow^*$) implications. We define intuitionistic negation in terms of the connectives:

$$\neg \phi ::= \phi \rightarrow \bot$$
Definition 1 (Layered graph model). A layered graph model, $\mathcal{M}$, of ILGL is a pair $(\mathcal{X}, \mathcal{V})$, where $\mathcal{X}$ is a preordered scaffold and $\mathcal{V} : \text{Prop} \to \varphi(\mathcal{X})$ is a persistent valuation; that is, $G \ll H$ and $G \in \mathcal{V}(p)$ implies $H \in \mathcal{V}(p)$.

Satisfaction in layered graph models is then defined in a familiar way.

Definition 2 (Satisfaction in layered graph models). Given a layered graph model $\mathcal{M} = (\mathcal{X}, \mathcal{V})$, we generate the satisfaction relation $\models_{\mathcal{M}} \subseteq \mathcal{X} \times \text{Form}$ as follows:

- $G \models_{\mathcal{M}} \top \text{ always}$
- $G \models_{\mathcal{M}} \bot \text{ never}$
- $G \models_{\mathcal{M}} \varphi$ iff $G \models_{\mathcal{M}} \psi$ and $G \models_{\mathcal{M}} \psi$
- $G \models_{\mathcal{M}} \varphi \lor \psi$ iff $G \models_{\mathcal{M}} \varphi$ or $G \models_{\mathcal{M}} \psi$
- $G \models_{\mathcal{M}} \varphi \rightarrow \psi$ iff, for all $G'$ such that $G \ll G'$, $G' \models_{\mathcal{M}} \varphi$ implies $G' \models_{\mathcal{M}} \psi$

Definition 3 (Validity). A formula $\phi$ is valid in a layered graph model $\mathcal{M}$ ($\models_{\mathcal{M}} \phi$) iff, for all $G \in \mathcal{X}$, $G \models_{\mathcal{M}} \phi$. A formula $\phi$ is valid ($\models \phi$) iff, for all layered graph models $\mathcal{M}$, $\models \phi$.

Lemma 1 (Persistence). Persistence extends to all formulae with respect to the layered graph semantics. That is, for all $\varphi \in \text{Form}$, $G \ll H$ and $G \models_{\mathcal{M}} \varphi$ implies $H \models_{\mathcal{M}} \varphi$.

Proof. By induction on the complexity of formulae. The additive fragment, corresponding to intuitionistic propositional logic (IPL), is standard and we restrict attention to two examples of the multiplicative connectives.

Suppose $G \models_{\mathcal{M}} \varphi \rightarrow \psi$ and $G \ll H$. There are $K, K'$ s.t. $K \ll K' \ll G$, with $K \models_{\mathcal{M}} \varphi$ and $K' \models_{\mathcal{M}} \psi$. By transitivity of $\ll$, $K \ll K' \ll H$, so $H \models_{\mathcal{M}} \varphi \rightarrow \psi$.

Suppose $G \models_{\mathcal{M}} \varphi \rightarrow \psi$. Then, for all $K$ such that $G \ll K$ and all $K'$ s.t. $K \ll K'$, if $K' \models_{\mathcal{M}} \varphi$, then $K \ll K' \models_{\mathcal{M}} \psi$. Let $G \ll H$ and suppose $H \ll K$ and $K'$ are s.t. $K \ll K' \ll K'$ s.t. $K \ll K' \models_{\mathcal{M}} \psi$. So, since $G \ll H \ll K$, it follows that $K \ll K' \models_{\mathcal{M}} \varphi$. So $H \models_{\mathcal{M}} \varphi \rightarrow \psi$.

Note that, unlike in BI, we require the restriction ‘for all $H, G \ll H \ldots$’ in the semantic clauses for the multiplicative implications. Without this we cannot prove persistence because we cannot proceed with the inductive step in those cases. The reason for this is that we put no restriction on the interaction between $\ll$ and $\ll$ in the definition of preordered scaffold. This is unlike the analogous case for BI, where the monoidal composition is required to be bifunctorial with respect to the ordering. One might resolve this issue with the following addendum to the definition of preordered scaffold: if $G \ll H$ and $H \ll E K \ll K$, then $G \ll E K \ll H \ll E K$.

Two natural examples of subgraph preorderings show that this would be undesirable. First, consider the layering preorder. Let $\ll$ be the reflexive, transitive closure of the relation $R(G, H)$ iff $H \ll G \ll$, restricted to the admissible subgraph set $X$. Figure 3 shows a subgraph $H$ with $G \ll H$ and $H \ll E K \ll K$ but $G \ll E K \ll K$ (we write $\ll$ for undefinedness). Second, consider the subgraph relation. In Figure 4, we have $G \ll H$ and $H \ll E K \ll K$ but $G \ll E K \ll K$. It is, however, the case that, with this ordering, if $G \ll H, H \ll E K$ and $G \ll E K \ll K$, then $G \ll E K \ll H \ll E K$.
Labelled tableaux. We define a labelled tableaux system for ILGL, utilising a method first showcased on tableaux systems for BBI and DMBI [15, 8] and in the spirit of previous work for BI [11].

Definition 4 (Graph labels). Let $\Sigma = \{c_i \mid i \in \mathbb{N}\}$ be a countable set of atomic labels. We define the set $\mathcal{L} = \{x \in \Sigma^* \mid 0 < |x| \leq 2\} \backslash \{c_i c_i \mid c_i \in \Sigma\}$ to be the set of graph labels. A sub-label $y$ of a label $x$ is a non-empty sub-word of $x$, and we denote the set of sub-labels of $x$ by $S(x)$.

The graph labels are a syntactic representation of the subgraphs of a model, with labels of length 2 representing a graph that can be decomposed into two layers. We exclude the possibility $c_i c_i$ as layering is anti-reflexive. In much the same way we give a syntactic representation of preorder.

Definition 5 (Constraints). A constraint is an expression of the form $x \preceq y$, where $x$ and $y$ are graph labels.

Let $\mathcal{C}$ be a set of constraints. The domain of $\mathcal{C}$ is the set of all non-empty sub-labels appearing in $\mathcal{C}$. In particular, $\mathcal{D}(\mathcal{C}) = \bigcup_{x \in \mathcal{L}} (S(x) \cup S(y))$ The alphabet of $\mathcal{C}$ is the set of atomic labels appearing in $\mathcal{C}$. In particular, we have $\mathcal{A}(\mathcal{C}) = \Sigma \cap \mathcal{D}(\mathcal{C})$.

\[
\begin{align*}
R_1: & \frac{x \preceq y}{x \preceq x} \\
R_2: & \frac{x \preceq y}{y \preceq y} \\
R_3: & \frac{x \preceq y}{y \preceq y} \\
R_4: & \frac{x \preceq y}{z \preceq z} \\
R_5: & \frac{x \preceq y}{x \preceq z} \\
R_6: & \frac{x \preceq y}{y \preceq z} \\
R_7: & \frac{x \preceq y}{x \preceq z} \\
(Tr): & \frac{x \preceq y}{x \preceq z}
\end{align*}
\]

Fig. 5. Rules for closure of constraints

Definition 6 (Closure of constraints). Let $\mathcal{C}$ be a set of constraints. The closure of $\mathcal{C}$, denoted $\overline{\mathcal{C}}$, is the least relation closed under the rules of Figure 5 such that $\mathcal{C} \subseteq \overline{\mathcal{C}}$. $\square$

RN/16/03 4
This closure yields a preorder on $\mathcal{D}(C)$, with $\langle R_1 \rangle - \langle R_6 \rangle$ generating reflexivity and (Tr) yielding transitivity. Crucially, taking the closure of the constraint set does not cause labels to proliferate and the generation of any particular constraint from an arbitrary constraint set $C$ is fundamentally a finite process.

**Proposition 1.** Let $C$ be a set of constraints. (1) $x \in \mathcal{D}(\overline{C})$ iff $x \preceq x \in \overline{C}$. (2) $\mathcal{D}(C) = \mathcal{D}(\overline{C})$ and $\mathcal{A}(C) = \mathcal{A}(\overline{C})$.  

**Lemma 2 (Compactness).** Let $C$ be a (possibly countably infinite) set of constraints. If $x \preceq y \in \overline{C}$, then there is a finite set of constraints $C_f \subseteq C$ such that $x \preceq y \in \overline{C_f}$.  

**Definition 7.** A labelled formula is a triple $(\Sigma, \varphi, x) \in [\mathbb{T}, \mathbb{F}] \times \mathbb{X} \times \mathbb{L}$, written $\varphi : x$. A constrained set of statements (CSS) is a pair $\langle F, C \rangle$, where $F$ is a set of labelled formulae and $C$ is a set of constraints, satisfying the following properties: for all $x \in \mathbb{L}$ and distinct $c_i, c_j, c_k \in \Sigma$, (1) (Ref) if $\varphi : x \in \mathcal{F}$, then $x \preceq x \in \overline{C}$, (2) (Contra) if $c, c_j \in \mathcal{D}(C)$, then $c, c_j \not\in \mathcal{D}(C)$, and (3) (Freshness) if $c, c_j \in \mathcal{D}(C)$, then $c, c_k, c_i, c_j, c, c_k \not\in \mathcal{D}(C)$. A CSS $\langle F, C \rangle$ is finite if $\mathcal{F}$ and $\mathcal{C}$ are finite. The relation $\subseteq$ is defined on CSSs by $\langle F, C \rangle \subseteq \langle F', C' \rangle$ if $\mathcal{F} \subseteq \mathcal{F}'$ and $\mathcal{C} \subseteq \mathcal{C}'$. We denote by $\langle F_f, C_f \rangle \subseteq \langle F, C \rangle$ when $\langle F_f, C_f \rangle \subseteq \langle F, C \rangle$ holds and $\langle F_f, C_f \rangle$ is finite.  

The CSS properties ensure models can be built from the labels: (Ref) ensures we have enough data for the closure rules to generate a preorder, (Contra) ensures the contra-commutativity of graph layering is respected, and (Freshness) ensures the layering structure of the models we construct is exactly that specified by the labels and constraints in the CSS. As with constraint closure, CSSs have a finite character.

**Proposition 2.** For any CSS $\langle F_f, C_f \rangle$ in which $\mathcal{F}_f$ is finite, there exists $C_f \subseteq C$ such that $C_f$ is finite and $\langle F_f, C_f \rangle$ is a CSS.  

Figure 6 presents the rules of the tableaux system for ILGL. That ‘$c_i$ and $c_j$ are fresh atomic labels’ means $c_i \neq c_j \in \Sigma \setminus \mathcal{A}(C)$. We denote by $\oplus$ the concatenation of lists.

**Definition 8 (Tableaux).** Let $\langle F_0, C_0 \rangle$ be a finite CSS. A tableau for this CSS is a list of CSS, called branches, built inductively according the following rules:

1. The one branch list $\langle F_0, C_0 \rangle$ is a tableau for $\langle F_0, C_0 \rangle$;
2. If the list $\mathcal{T}_m \oplus \langle F, C \rangle \oplus \mathcal{T}_n$ is a tableau for $\langle F_0, C_0 \rangle$ and

$$
\text{cond}(\mathcal{F}, C) \\
\langle F_1, C_1 \rangle \ldots \langle F_k, C_k \rangle
$$

is an instance of a rule of Figure 6 for which $\text{cond}(\mathcal{F}, C)$ is fulfilled, then the list $\mathcal{T}_m \oplus \langle \mathcal{F} \cup \mathcal{F}_1, C \cup C_1 \rangle ; \ldots ; \langle \mathcal{F} \cup \mathcal{F}_k, C \cup C_k \rangle \oplus \mathcal{T}_n$ is a tableau for $\langle F_0, C_0 \rangle$.

A tableau for the formula $\varphi$ is a tableau for $\langle \varphi : c_0 \rangle, \{ c_0 \preceq c_0 \}$.

It is a simple but tedious exercise to show that the rules of Figure 6 preserve the CSS properties of Definition 7. We now give the notion of proof for our labelled tableaux.
Definition 9 (Closed tableau). A CSS \( (\mathcal{F}, \mathcal{C}) \) is closed if one of the following conditions holds: (1) \( \mathcal{T} \varphi : x \in \mathcal{F} \); (2) \( \mathcal{F} \varphi : y \in \mathcal{F} \) and \( x \leq y \in \mathcal{C} \); and (3) \( \top \mathcal{F} : x \in \mathcal{F} \). A CSS is open iff it is not closed. A tableau is closed iff all its branches are closed. A proof for a formula \( \varphi \) is a closed tableau for \( \varphi \).

CSSs are related back to the graph semantics via the notion of realization.

Definition 10 (Realization). Let \( (\mathcal{F}, \mathcal{C}) \) be a CSS. A realization of \( (\mathcal{F}, \mathcal{C}) \) is a triple \( \mathcal{R} = (X, \mathcal{V}, [\_]) \) where \( \mathcal{M} = (X, \mathcal{V}) \) is a layered graph model and \( [\_] : \mathcal{D}(\mathcal{C}) \rightarrow X \) is such that (1) \([\_]\) is total: for all \( x \in \mathcal{D}(\mathcal{C}) \), \([x] \downarrow \), (2) for all \( x \in \mathcal{D}(\mathcal{C}) \), if \( x = c_i c_j \), then \([c_i] @ [c_j] \downarrow \) and \([x] = [c_i] @ [c_j] \downarrow \), (3) if \( x \leq y \in \mathcal{C} \), then \([x] \leq_M [y] \), (4) if \( \mathcal{T} \varphi : x \in \mathcal{F} \), then \([x] \models_M \varphi \), and (5) if \( \mathcal{F} \varphi : x \in \mathcal{F} \), then \([x] \not\models_M \varphi \).

We say that a CSS is realizable if there exists a realization of it. We say that a tableau is realizable if at least one of its branches is realizable. We can also show that the relevant clauses of the definition extend to the closure of the constraint set automatically.

Proposition 3. Let \( (\mathcal{F}, \mathcal{C}) \) be a CSS and \( \mathcal{R} = (X, \mathcal{V}, [\_]) \) a realization of it. Then: (1) for all \( x \in \mathcal{D}(\mathcal{C}) \), \([x]\) is defined; (2) if \( x \leq y \in \mathcal{C} \), then \([x] \leq_M [y] \).

3 Metatheory

We now establish the soundness and, via countermodel extraction, the completeness of ILGL’s tableaux system with respect to layered graph semantics. The proof of sound-
ness is straightforward (cf. [8, 10, 11, 15]). We begin with two key lemmas about realizability and closure. Their proofs proceed by simple case analysis.

**Lemma 3.** The tableaux rules for ILGL preserve realizability. □

**Lemma 4.** Closed branches are not realizable. □

**Theorem 1 (Soundness).** If there exists a closed tableau for the formula \( \varphi \), then \( \varphi \) is valid in layered graph models.

*Proof.* Suppose that there exists a proof for \( \varphi \). Then there is a closed tableau \( T_\varphi \) for the CSS \( \mathcal{C} = \langle \{ \mathcal{P}_C : c_0 \}, [c_0 \preceq c_0] \rangle \). Now suppose that \( \varphi \) is not valid. Then there is a countermodel \( \mathcal{M} = (X, \mathcal{V}) \) and a subgraph \( G \in X \) such that \( G \not\models \varphi \). Define \( \mathcal{R} = (\mathcal{M}, \mathcal{V}, \mathcal{L}) \) with \([c_0] = G\). Note that \( \mathcal{R} \) is a realization of \( \mathcal{C} \), hence by Lemma 3, \( T_\varphi \) is realizable. By Lemma 4, \( T_\varphi \) cannot be closed. But, this contradicts the fact that \( T_\varphi \) is a proof and therefore a closed tableau. It follows that \( \varphi \) is valid. □

We now proceed to establish the completeness of the labelled tableaux with respect to layered graph semantics. We begin with the notion of a Hintikka CSS, which will facilitate the construction of countermodels.

**Definition 11 (Hintikka CSS).** A CSS \( \langle \mathcal{F}, C \rangle \) is a Hintikka CSS iff, for any formulas \( \varphi, \psi \in \text{Form} \) and any graph labels \( x, y \in \mathbb{L} \), we have the following:

1. \( T_\varphi : x \notin \mathcal{F} \) or \( \mathcal{P}_\varphi : y \notin \mathcal{F} \) or \( x \preceq y \not\in \mathcal{C} \)
2. \( \mathcal{P}_\top : x \notin \mathcal{F} \)
3. \( \mathcal{P}_\bot : x \notin \mathcal{F} \)
4. If \( T_\varphi \land \psi : x \in \mathcal{F} \), then \( T_\varphi : x \in \mathcal{F} \) and \( T_\psi : x \in \mathcal{F} \)
5. If \( \mathcal{P}_\varphi \land \psi : x \in \mathcal{F} \), then \( \mathcal{P}_\varphi : x \in \mathcal{F} \) or \( \mathcal{P}_\psi : x \in \mathcal{F} \)
6. If \( T_\varphi \lor \psi : x \in \mathcal{F} \), then \( T_\varphi : x \in \mathcal{F} \) or \( T_\psi : x \in \mathcal{F} \)
7. If \( \mathcal{P}_\varphi \lor \psi : x \in \mathcal{F} \), then \( \mathcal{P}_\varphi : x \in \mathcal{F} \) and \( \mathcal{P}_\psi : x \in \mathcal{F} \)
8. If \( T_\varphi \rightarrow \psi : x \in \mathcal{F} \), then, for all \( y \in \mathcal{L} \), if \( x \preceq y \not\in \mathcal{C} \), then \( \mathcal{P}_\varphi : y \in \mathcal{F} \) or \( T_\psi : y \in \mathcal{F} \)
9. If \( \mathcal{P}_\varphi \rightarrow \psi : x \in \mathcal{F} \), then there exists \( y \in \mathcal{L} \) such that \( x \preceq y \not\in \mathcal{C} \) and \( T_\varphi : y \in \mathcal{F} \) and \( T_\psi : y \in \mathcal{F} \)
10. If \( T_\varphi \triangleright \psi : x \in \mathcal{F} \), then there are \( c_i, c_j \in \Sigma \) such that \( c_i, c_j \not\in x \not\in \mathcal{C} \) and \( T_\varphi : c_i \in \mathcal{F} \) and \( T_\psi : c_j \in \mathcal{F} \)
11. If \( \mathcal{P}_\varphi \triangleright \psi : x \in \mathcal{F} \), then, for all \( c_i, c_j \in \Sigma \), if \( c_i, c_j \not\in x \not\in \mathcal{C} \), then \( \mathcal{P}_\varphi : c_i \in \mathcal{F} \) or \( \mathcal{P}_\psi : c_j \in \mathcal{F} \)
12. If \( T_\varphi \triangleright \psi : x \in \mathcal{F} \), then, for all \( c_i, c_j \in \Sigma \), if \( x \preceq c_i \not\in \mathcal{C} \) and \( c_i, c_j \in \mathcal{D}(\mathcal{C}) \), then \( T_\varphi : c_i \in \mathcal{F} \) or \( T_\psi : c_j, c_i \in \mathcal{F} \)
13. If \( \mathcal{P}_\varphi \triangleright \psi : x \in \mathcal{F} \), then there are \( c_i, c_j \in \Sigma \) such that \( x \preceq c_i \not\in \mathcal{C} \) and \( c_i, c_j \in \mathcal{D}(\mathcal{C}) \) and \( T_\varphi : c_j \in \mathcal{F} \) and \( T_\psi : c_i, c_j \in \mathcal{F} \)
14. If \( T_\varphi \triangleright \psi : x \in \mathcal{F} \), then, for all \( c_i, c_j \in \Sigma \), if \( x \preceq c_i \not\in \mathcal{C} \) and \( c_i, c_j \in \mathcal{D}(\mathcal{C}) \), then \( T_\varphi : c_j \in \mathcal{F} \) or \( T_\psi : c_i, c_j \in \mathcal{F} \)
15. If \( \mathcal{P}_\varphi \triangleright \psi : x \in \mathcal{F} \), then there are \( c_i, c_j \in \Sigma \) such that \( x \preceq c_i \not\in \mathcal{C} \) and \( c_i, c_j \in \mathcal{D}(\mathcal{C}) \) and \( T_\varphi : c_j \in \mathcal{F} \) and \( \mathcal{P}_\psi : c_i, c_j \in \mathcal{F} \). □

We now give the definition of a function \( \Omega \) that extracts a countermodel from a Hintikka CSS. A Hintikka CSS can thus be seen as the labelled tableaux counterpart of Hintikka sets, which are maximally consistent sets satisfying a subformula property.
**Definition 12 (Function \( \Omega \)).** Let \( \langle F, C \rangle \) be a Hintikka CSS. The function \( \Omega \) associates to \( \langle F, C \rangle \) a tuple \( \Omega(\langle F, C \rangle) = (G, E, X, \ll, V) \), such that (1) \( V(G) = A(C) \), (2) \( E(G) = \{(c_i, c_j) \mid c_i, c_j \in D(C)\} = E \), \( X = \{x^{\Omega} \mid x \in D(C)\} \), where \( V(c_i^{\Omega}) = \{c_i\} \), \( E(c_i^{\Omega}) = \emptyset \). \( V((c_i, c_j)^{\Omega}) = \{c_i, c_j\} \), and \( E((c_i, c_j)^{\Omega}) = \{(c_i, c_j)\} \). (3) \( x^{\Omega} \ll y^{\Omega} \iff x \ll y \in \overline{C} \), and (4) \( x^{\Omega} \in V(p) \iff \exists y \in D(C) \text{ such that } y \ll x \in \overline{C} \) and \( \exists \tau : y \in F \).

The next lemma shows that there is a precise correspondence between the structure that the Hintikka CSS properties impose on the labels and the layered structure specified by the construction of the model.

**Lemma 5.** Let \( \langle F, C \rangle \) be a Hintikka CSS and \( \Omega(\langle F, C \rangle) = (G, E, X, \ll, V) \). (1) If \( c_i, c_j \in A(C) \), then \( c_i c_j \in D(C) \) iff \( c_i \ll c_j \). (2) If \( c_i, c_j \in D(C) \), then \( (c_i, c_j)^{\Omega} = c_i^{\Omega} \otimes E c_j^{\Omega} \). (3) \( x^{\Omega} \ll y^{\Omega} \downarrow \) iff there exist \( c_i, c_j \in A(C) \) s.t. \( x = c_i, y = c_j \) and \( c_i, c_j \in D(C) \).

**Proof.** 1. Let \( c_i c_j \in D(C) \). Then by CSS property (Contra) we have \( c_i c_j \notin D(C) \). Hence by definition of \( \Omega \) we have \( (c_i, c_j) \notin E \) and \( (c_j, c_i) \notin E \). Thus \( c_i \ll c_j \) as required. The other direction is trivial.

2. Immediate from 1. and the definition of \( \Omega \).

3. The right-to-left direction is trivial, so assume \( x^{\Omega} \ll y^{\Omega} \downarrow \). There are three possible cases for \( x \) and \( y \) other than \( x = c_i \) and \( y = c_j \); we attend to one as the others are similar. Suppose \( x = c_i c_j \) and \( y = c_k \). Then \( x^{\Omega} \ll y^{\Omega} \downarrow \) must hold because of either \((c_i, c_k) \in E \) or \((c_j, c_k) \in E \). That is, \( c_i c_k \in D(C) \) or \( c_j c_k \in D(C) \). In both cases the CSS property (Freshness) is contradicted so neither can hold. It follows that only the case \( x = c_i \) and \( y = c_j \) is non-contradictory, and so by 1. \( c_i c_j \in D(C) \).

**Lemma 6.** Let \( \langle F, C \rangle \) be a Hintikka CSS. \( \Omega(\langle F, C \rangle) \) is a layered graph model.

**Proof.** \( G \) is clearly a graph and \( \ll \) being a preorder on \( X \) can be read off of the rules for the closure of constraint sets. Thus the only non-trivial aspects of the proof are that \( X \) is admissible and that \( \forall \) is persistent.

- \( X \) is an admissible subgraph set.

Let \( G, H \in Sg(G) \) with \( G \otimes E H \downarrow \). First we assume \( G, H \in X \). Then \( G = x^{\Omega} \) and \( H = y^{\Omega} \) for labels \( x, y \). By the previous lemma it follows that \( x = c_i \) and \( y = c_j \) and \( c_i c_j \in D(C) \). Thus \( G \otimes E H = c_i^{\Omega} \otimes E c_j^{\Omega} = (c_i, c_j)^{\Omega} \in X \). Now suppose \( G \otimes E H \notin X \).

Then \( G \otimes E H = x^{\Omega} \) for some \( x \in D(C) \). The case \( x = c_i \) is clearly impossible as \( E(c_i^{\Omega}) = \emptyset \) so necessarily \( x = c_i c_j \). Then we have \( c_i, c_j \in D(C) \) as sub-labels of \( c_i c_j \) and \( c_i^{\Omega} \otimes E c_j^{\Omega} \downarrow \) with \( c_i^{\Omega} \otimes E c_j^{\Omega} \) the only possible composition equal to \((c_i, c_j)^{\Omega}\). It follows that \( G = c_i^{\Omega} \in X \) and \( H = c_j^{\Omega} \in X \) as required.

- \( \forall \) is a persistent valuation.

Let \( G \in V(p) \) with \( G \ll H \). Then \( G = x^{\Omega} \) and \( H = y^{\Omega} \) for some \( x, y \in D(C) \) with \( x \ll y \in \overline{C} \). By definition of \( V \) there exists \( z \in D(C) \) with \( z \ll x \in \overline{C} \) and \( \exists \tau : z \in F \).

By closure rule \((T_r)\) we have \( z \ll y \in \overline{C} \), so \( H = y^{\Omega} \in V(p) \).

**Lemma 7.** Let \( \langle F, C \rangle \) be a Hintikka CSS and \( M = \Omega(\langle F, C \rangle) = (G, E, X, \ll, V) \). For all formulas \( \varphi \in \text{Form} \), and all \( x \in D(C) \), we have (1) if \( F : x \in F \), then \( x^{\Omega} \not\models_M \varphi \), and (2) if \( \exists \varphi : x \in F \), then \( x^{\Omega} \models_M \varphi \). Hence, if \( F : x \in F \), then \( \varphi \) is not valid and \( \Omega(\langle F, C \rangle) \) is a countermodel of \( \varphi \).
Proof. We proceed by a simultaneous structural induction on \( \varphi \).

- Base cases.
  - Case \( \mathcal{F}p : x \in \mathcal{F} \). We suppose that \( x^0 \models_M p \). Then \( x^0 \in \mathcal{V}(p) \). By the definition of \( \mathcal{V} \), there is a label \( y \) such that \( y \preceq x \in \bar{C} \) and \( \mathcal{F}p : y \in \mathcal{F} \). Then by condition (1) of Definition 11, \( (\mathcal{F}, C) \) is not a Hintikka CSS, a contradiction. It follows that \( x^0 \not\models_M p \).
  - Case \( \mathcal{T}p : x \in \mathcal{F} \). By property (Ref), \( x \preceq x \in \bar{C} \). Thus, by definition of \( \mathcal{V} \) we have \( x^0 \in \mathcal{V}(p) \). Thus \( x^0 \models_M p \).
  - Cases \( \mathcal{F}⊥ : x \in \mathcal{F} \), \( \mathcal{T}⊥ : x \in \mathcal{F} \), \( \mathcal{F}τ : x \) and \( \mathcal{T}τ : x \) are straightforward consequences of the definition of Hintikka CSS and the layered graph semantics.

- Inductive step. We now suppose that (1) and (2) hold for formulae \( \varphi \) and \( ψ \) (IH). We attend only to the cases \( (\mathcal{T} →), (\mathcal{T} ▶) \) and \( (\mathcal{T} ▶) \) as the others are similar.
  - Case \( \mathcal{T}φ → ψ : x \in \mathcal{F} \). Suppose \( x^0 \preceq y^0 \). Then \( x \preceq y \in \bar{C} \) and by Definition 11 property (8) it follows that \( \mathcal{F}φ : y \in \mathcal{F} \) or \( \mathcal{T}ψ : y \in \mathcal{F} \). By (IH) it follows that if \( y^0 \models_M φ \) then \( y^0 \models_M ψ \) as required.
  - Case \( \mathcal{T}φ ▶ ψ : x \in \mathcal{F} \). By Definition 11 property (10) there exist labels \( c_i, c_j \in \mathcal{D}(C) \) such that \( c_i c_j \preceq y \in \bar{C} \) and \( \mathcal{T}φ : c_i \in \mathcal{F} \) and \( \mathcal{T}ψ : c_j \in \mathcal{F} \). By (IH) we have \( c_i^0 \models_M φ \) and \( c_j^0 \models_M ψ \). Further, by definition of \( Ω \) we have that \( (c_i, c_j)^0 = c_i^0 @_{E} c_j^0 \preceq x^0 \), so \( x^0 \models_M φ ▶ ψ \).
  - Case \( \mathcal{T}φ ▶ ▶ ψ : x \in \mathcal{F} \). Suppose \( x^0 \preceq y^0 \) with \( y^0 @_{E} x^0 \downarrow \) and \( x^0 \models_M φ \). By Lemma 5 we know \( y = c_i, z = c_j \in \mathcal{A}(C) \) with \( c_i c_j \in \mathcal{D}(C) \). Hence by Definition 11 property 12, either \( \mathcal{F}φ : c_j \in \mathcal{F} \) or \( \mathcal{T}ψ : c_i c_j \in \mathcal{F} \). By (IH) it follows either \( c_j^0 \models_M φ \) or \( (c_i, c_j)^0 = c_i^0 @_{E} c_j^0 \models_M ψ \). As we know the former cannot be true, it must be the latter. Hence \( x^0 \models_M φ ▶ ▶ ψ \) as required. □

This construction of a countermodel would fail in a labelled tableaux system for LGL (i.e., the layered graph logic with classical additives [6]). This is because it is impossible to construct the internal structure of each subgraph in the model systematically, as the classical semantics for \( ▶ \) demands strict equality between the graph under interpretation and the decomposition into layers. This issue is sidestepped for ILGL since each time the tableaux rules require a decomposition of a subgraph into layers we can move to a ‘fresh’ layered subgraph further down the ordering. Thus we can safely turn each graph label into the simplest instantiation of the kind of graph it represents: either a single vertex (indecomposable) or two vertices and an edge (layered).

We now show how to construct such a CSS. We first require a listing of all labelled formulae that may need to be added to the CSS in order to satisfy properties 4–15. We require a particularly strong condition on the listing to make this procedure work: that every labelled formula appears infinitely often to be tested.

Definition 13 (Fair strategy). A fair strategy for a language \( \mathcal{L} \) is a labelled sequence of formulae \( (\mathcal{S} \chi_i : (x_i))_{i \in \mathbb{N}} \) in \( [\mathcal{T}, \mathcal{F}] \times \text{Form} \times \mathcal{L} \) such that \( \{ i \in \mathbb{N} \mid \mathcal{S} \chi_i : (x_i) \equiv \mathcal{S} \chi : x \} \) is infinite for any \( \mathcal{S} \chi : x \in [\mathcal{T}, \mathcal{F}] \times \text{Form} \times \mathcal{L} \).

Proposition 4. There exists a fair strategy for the language of ILGL.

Proof. See [8]. □
Next we need the concept of an oracle. Here an oracle allows Hintikka sets to be constructed inductively, testing the required consistency properties at each stage.

**Definition 14.** Let $\mathcal{P}$ be a set of CSSs. (1) $\mathcal{P}$ is $\subseteq$-closed if $\langle F, C \rangle \in \mathcal{P}$ holds whenever $\langle F', C' \rangle \subseteq \langle F, C \rangle$ and $\langle F', C' \rangle \in \mathcal{P}$ holds. (2) $\mathcal{P}$ is of finite character if $\langle F, C \rangle \in \mathcal{P}$ holds whenever $\langle F_f, C_f \rangle \in \mathcal{P}$ holds for every $\langle F_f, C_f \rangle \subseteq_f \langle F, C \rangle$. (3) $\mathcal{P}$ is saturated if, for any $\langle F, C \rangle \in \mathcal{P}$ and any instance $\text{cond}(F, C)$

$$\langle F_1, C_1 \rangle | \ldots | \langle F_k, C_k \rangle$$

of a rule of Figure 6 if $\text{cond}(F, C)$ is fulfilled, then $\langle F \cup F_f, C \cup C_f \rangle \in \mathcal{P}$, for at least one $i \in \{1, \ldots, k\}$.

**Definition 15 (Oracle).** An oracle is a set of open CSSs which is $\subseteq$-closed, of finite character, and saturated.

**Definition 16 (Consistency/finite consistency).** Let $\langle F, C \rangle$ be a CSS. We say $\langle F, C \rangle$ is consistent if it is finite and has no closed tableau. We say $\langle F, C \rangle$ is finitely consistent if every finite sub-CSS $\langle F_f, C_f \rangle$ is consistent.

**Proposition 5.** (1) Consistency is $\subseteq$-closed. (2) A finite CSS is consistent iff it is finitely consistent.

**Proof.** See [8].

**Lemma 8.** The set of finitely consistent CSS, $\mathcal{P}$, is an oracle.

**Proof.** For $\subseteq$-closure and finite character see [8]. We show the cases $\langle \top \rightarrow \top \rangle$ and $\langle \top \rightarrow \top \rangle$ for saturation: the rest are similar. Let $\langle F, C \rangle \in \mathcal{P}$

- $\langle \top \rightarrow \top \rangle \psi : x \in F$. We show $\langle F \cup \langle \top \rightarrow \top \rangle : c_i, \top \psi : c_j, C \cup \{c_j x \leq x\} \rangle \in \mathcal{P}$. Let $\langle F_f, C_f \rangle \subseteq_f \langle F \cup \langle \top \rightarrow \top \rangle : c_i, \top \psi : c_j, C \cup \{c_j x \leq x\} \rangle \in \mathcal{P}$. Since, $\langle \top \rightarrow \top \rangle \psi : x \in F$, by compactness, there exists $C_0 \subseteq C$ such that $x \leq x \in C_0$. Now define

$$F'_f = \langle F_f \setminus \langle \top \rightarrow \top \rangle : c_i, \top \psi : c_j \rangle \cup \langle \top \rightarrow \top \rangle \psi : x \rangle$$

$$C'_f = C_f \cup C_0$$

Then $\langle F'_f, C'_f \rangle$ is a CSS and $\langle F'_f, C'_f \rangle \subseteq_f \langle F, C \rangle$ so it is consistent. We have that $\langle F'_f \cup \langle \top \rightarrow \top \rangle : c_i, \top \psi : c_j \rangle, C'_f \cup \{c_j x \leq x\} \rangle$ is a tableau for $\langle F'_f, C'_f \rangle$. Thus if it is possible for $\langle F'_f \cup \langle \top \rightarrow \top \rangle : c_i, \top \psi : c_j \rangle, C'_f \cup \{c_j x \leq x\} \rangle$ to be closed then so too is it for $\langle F'_f, C'_f \rangle$: a contradiction. Hence it is consistent. We have that $\langle F_f, C_f \rangle \subseteq \langle F'_f \cup \{\top \rightarrow \top \} : c_i, \top \psi : c_j \rangle, C'_f \cup \{c_j x \leq x\} \rangle$ so $\langle F_f, C_f \rangle$ is consistent by Proposition 5.

- $\langle \top \rightarrow \top \rangle \psi : x \in F$ and $x \leq y, yz \leq yz \in C$. Suppose neither $\langle F \cup \{\top \psi : z\} \rangle \in \mathcal{P}$ nor $\langle F \cup \{\top \psi : xz\} \rangle \in \mathcal{P}$. Then there exist $\langle F^A \cup \{\top \psi : z\} \rangle \subseteq_f \langle F \cup \{\top \psi : z\} \rangle$ and $\langle F^B \cup \{\top \psi : yz\} \rangle \subseteq_f \langle F \cup \{\top \psi : yz\} \rangle$ that are inconsistent. By compactness, there exist $C_0, C_1 \subseteq C$ such that $x \leq x \in C_0$ and $yz \leq yz \in C_1$. Thus we define $F'_f = \langle F^A \setminus \{\top \psi : z\} \rangle \cup \langle F^B \setminus \{\top \psi : yz\} \rangle \cup \langle \top \rightarrow \top \rangle \psi : x \rangle$ and $C'_f = C_f \cup C_0 \cup C_1$. Then $\langle F'_f, C'_f \rangle$ is a finite CSS and $\langle F'_f \cup \{\top \psi : z\}, C'_f \rangle; \langle F'_f \cup \{\top \psi : yz\}, C'_f \rangle$ is a tableau for it.
We have \( \langle F_f^A, C_f^0 \rangle \subseteq_f \langle F_f^j \cup \{ \varphi : z \}, C_f^j \rangle \) and \( \langle F_f^B, C_f^0 \rangle \subseteq_f \langle F_f^j \cup \{ \forall \psi : yz \}, C_f^j \rangle \) so by \( \subseteq \)-closure of consistency \( \langle F_f^A, C_f^0 \rangle \) and \( \langle F_f^B, C_f^0 \rangle \) are inconsistent: respectively let \( T_A \) and \( T_B \) be closed tableaux for them. Then \( T_A \oplus T_B \) is a closed tableau for \( \langle F_f^j, C_f^j \rangle \) and the CSS is inconsistent: contradicting \( \langle F_f^j, C_f^j \rangle \not\subseteq_f \langle F, C \rangle \in \mathcal{P}. \)

We can now show completeness of our tableau system. Consider a formula \( \varphi \) for which there exists no closed tableau. We show there is a countermodel to \( \varphi \). We start with the initial tableau \( T_0 \) for \( \varphi \). Then, we have (1) \( T_0 = \{ (\exists \varphi : c_0, [c_0 \ll c_0]) \} \) and (2) \( T_0 \) cannot be closed. Let \( \mathcal{P} \) be as in Lemma 8. By Proposition 4, there exists a fair strategy, which we denote by \( S \). With \( S, x_i : (x_i) \) the \( i \)-th formula of \( S \). As \( T_0 \) cannot be closed, \( \langle (\exists \varphi : c_0, [c_0 \ll c_0]) \rangle \in \mathcal{P} \). We build a sequence \( \langle F_i, C_i \rangle_{i \geq 0} \) as follows:

- \( \langle F_0, C_0 \rangle = \{ (\exists \varphi : c_0, [c_0 \ll c_0]) \} \);
- if \( \langle F_i \cup \{ x_i : (x_i) \}, C_i \rangle \not\in \mathcal{P} \), then we have \( \langle F_{i+1}, C_{i+1} \rangle = \langle F_i, C_i \rangle \); and
- if \( \langle F_i \cup \{ x_i : (x_i) \}, C_i \rangle \in \mathcal{P} \), then we have \( \langle F_{i+1}, C_{i+1} \rangle = \langle F_i \cup \{ x_i : (x_i) \} \cup F_c, C_i \cup C_c \rangle \) such that \( F_c \) and \( C_c \) are determined by

<table>
<thead>
<tr>
<th>( S_i )</th>
<th>( x_i )</th>
<th>( F_c )</th>
<th>( C_c )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( \varphi \rightarrow \psi )</td>
<td>( \exists \varphi : c_{331}, \exists \psi : c_{331+1} )</td>
<td>( x_i \ll c_{331+1} )</td>
</tr>
<tr>
<td>2</td>
<td>( \varphi \rightarrow \psi )</td>
<td>( \exists \varphi : c_{332}, \exists \psi : c_{332+1} )</td>
<td>( c_{332+1} \ll c_{332} )</td>
</tr>
<tr>
<td>3</td>
<td>( \varphi \rightarrow \psi )</td>
<td>( \exists \varphi : c_{331}, \exists \psi : c_{331+1} c_{332+1} )</td>
<td>( x_i \ll c_{331+1} c_{332+1} )</td>
</tr>
<tr>
<td>Otherwise</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

with \( 3 = \max \{ j \mid c_j \in A(C_i) \cup S(x_i) \} \).

**Proposition 6.** For any \( i \in \mathbb{N} \), the following properties hold: (1) \( F_i \subseteq F_{i+1} \) and \( C_i \subseteq C_{i+1} \); (2) \( \langle F_i, C_i \rangle \in \mathcal{P} \).

**Proof.** Only 2 is non-trivial. and we prove it by induction on \( i \). The base case \( i = 0 \) is given by our initial assumption. Now for the inductive hypothesis (IH) we have that \( \langle F_i, C_i \rangle \in \mathcal{P} \). Then the inductive step is an immediate consequence of Lemma 8 for the non-trivial cases.

We now define the limit \( \langle F_{\infty}, C_{\infty} \rangle = \langle \cup_{i>0} F_i, \cup_{i>0} C_i \rangle \) of the sequence \( \langle F_i, C_i \rangle_{i \geq 0} \).

**Proposition 7.** The following properties hold: (1) \( \langle F_{\infty}, C_{\infty} \rangle \in \mathcal{P} \); (2) For all labelled formulae \( \varphi : x \), if \( \langle F_{\infty} \cup [\varphi : x], C_{\infty} \rangle \in \mathcal{P} \), then \( \varphi : x \in F_{\infty} \).

**Proof.** 1. First note that \( \langle F_{\infty}, C_{\infty} \rangle \) is a CSS since each stage of construction satisfies (Ref) and by our choice of constants throughout the construction (Contra) and (Freshness) are satisfied. Further, it is open since otherwise there would be some stage \( \langle F_k, C_k \rangle \) at which the offending closure condition is satisfied, which would contradict that each \( \langle F_i, C_i \rangle \) is consistent. Now let \( \langle F_f, C_f \rangle \subseteq_f \langle F_{\infty}, C_{\infty} \rangle \). Then there exists \( k \in \mathbb{N} \) such that \( \langle F_f, C_f \rangle \subseteq_f \langle F_k, C_k \rangle \). By Proposition \( \langle F_k, C_k \rangle \in \mathcal{P} \) so it follows \( \langle F_f, C_f \rangle \in \mathcal{P} \). As \( \mathcal{P} \) is of finite character, we thus have \( \langle F_{\infty}, C_{\infty} \rangle \in \mathcal{P} \).

2. First note that \( \langle F_{\infty} \cup [\varphi : x], C_{\infty} \rangle \) is a CSS so (Contra) and (Freshness) are satisfied when the label \( x \) is introduced. By compactness, there exists finite \( C_0 \subseteq C_{\infty} \) such that \( x \ll x \in C_0 \). As it is finite, there exists \( k \in \mathbb{N} \) such that \( C_0 \subseteq C_k \) and by fairness
there exists \( l \geq k \) such that \( S_k x : (x_i) \cong S \varphi : x \). Since (Freshness) and (Contra) are fulfilled with respect to \( F_\infty \) they are also fulfilled with respect to \( F_1 \cup \{ \varphi \} \) so \( \langle F_{i+1}, C_{i+1} \rangle = (F_1 \cup \{ \varphi \}, C_i) \in \mathcal{P} \) and \( \langle F_{i+1}, C_{i+1} \rangle = (F_1 \cup \{ \varphi \}, F C_i) \). Hence \( S \varphi : x \in F_\infty \).

Lemma 9. The limit CSS is a Hintikka CSS.

**Proof.** For properties (1) – (3) we have that \( \langle F_\infty, C_\infty \rangle \) is open. For the other conditions, the saturation property of the oracle \( \mathcal{P} \) and 2. of Proposition 7 suffice.

Theorem 2 (Completeness). If \( \varphi \) is valid, then there exists a closed tableau for \( \varphi \).

**Proof.** Suppose there exists no proof for the formula \( \varphi \). Then by Lemma 9 we can construct the Hintikka CSS \( \langle F_\infty, C_\infty \rangle \) from \( T_0 = \{ \{ F \varphi : c_0, [c_0 \lessdot c_0]) \} \} \) as outlined above, with \( F \varphi : c_0 \in F_\infty \). Then by Lemma 7, \( \Omega(\langle F_\infty, C_\infty \rangle) \) is a countermodel for \( \varphi \). That is, \( \varphi \) is not valid.

4 A Hilbert system and an algebraic semantics

We give a Hilbert-type proof system, ILGL_{41}, for ILGL in Figure 7. The additive fragment, corresponding to intuitionistic propositional logic, is standard (e.g., [2]). The presentation of the multiplicative fragment is similar to that for BI’s multiplicatives [19], but for the non-commutative and non-associative (following from the absence of a multiplicative counterpart to \( \land_2 \) conjunction, \( \lor \), together with its associated left and right implications (cf. [13, 14]).

\[
\begin{array}{cccc}
\varphi \vdash \varphi & \text{(Ax)} & \varphi \vdash \psi & \psi \vdash \chi & \text{(Cut)} & \varphi \vdash \top & \text{(T)} & \bot \vdash \bot & \text{((\bot))} \\
\varphi \vdash \varphi \lor \varphi \land \chi & \text{(\lor_1)} & \varphi \land \varphi \vdash \varphi_1 & \varphi_2 \vdash \varphi_1 \lor \varphi_2 & \text{((\lor_2))} & \varphi \vdash \chi & \varphi \vdash \chi & \psi \vdash \chi & \text{((\lor))} \\
\varphi \vdash \psi \rightarrow \chi & \varphi \vdash \psi & \chi \vdash \varphi \rightarrow \chi & \text{((\rightarrow_1))} & \varphi \land \psi \vdash \chi & \varphi \vdash \psi \rightarrow \chi & \text{((\rightarrow_2))} & \varphi \vdash \psi & \chi \vdash \psi & \psi \vdash \psi & \text{((\lor))} \\
\varphi \vdash \psi \lor \chi & \varphi \vdash \psi \land \chi & \text{((\lor))} & \varphi \vdash \psi \lor \chi & \varphi \vdash \psi \land \chi & \text{((\land))} & \varphi \vdash \psi \lor \chi & \text{((\land))} & \varphi \vdash \psi \lor \chi & \text{((\land))} \\
\end{array}
\]

Fig. 7. Rules of the Hilbert system, ILGL_{41}, for ILGL

This section concludes with equivalence of ILGL_{41} and ILGL’s tableaux system.

**Definition 17 (Layered Heyting algebra).** A layered Heyting algebra is a structure \( A = (A, \land, \lor, \rightarrow, \bot, \top, \blacktriangleright, \blacktriangleleft) \) such that \( (A, \land, \lor, \rightarrow, \bot, \top) \) is a Heyting algebra, \( \blacktriangleleft, \blacktriangleright \), and \( \blacktriangleright \) are binary operations on \( A \) satisfying \( a \leq a' \) and \( b \leq b' \) implies \( a \blacktriangleleft b \leq a' \blacktriangleleft b' \) and \( a \blacktriangleright b \leq c \) iff \( a \leq b \blacktriangleright c \) iff \( b \leq a \blacktriangleright c \).
We interpret ILGL on layered Heyting algebras. Let $\mathcal{V} : \text{Prop} \to A$ be a valuation on the layered Heyting algebra $(A, \land A, \lor A, \rightarrow A, \top A, \downarrow A)$. We maintain the subscripts to distinguish the operations of the algebra from the connectives of ILGL.

We uniquely define an interpretation function $[\cdot] : \text{Form} \to A$ by extending with respect to the connectives in the usual fashion: $[\top] = \top_A$, $[\bot] = \bot_A$, $[p] = V(p)$, and $[\varphi \circ \psi] = [\varphi] \circ_A [\psi]$ for $\circ \in \{\land, \lor, \rightarrow, \downarrow, \uparrow\}$.

**Proposition 8 (Soundness).** For any layered Heyting algebra $A$ and any interpretation $[\cdot] : \text{Prop} \to A$: if $\varphi \vdash \psi$ then $[\varphi] \leq [\psi]$.

**Proof.** By induction on the derivation rules of ILGL$_{\land A}$. The cases for the additive fragment are standard. For rule $(\uparrow)$, we use the property $a \leq_A a'$ and $b \leq_A b'$ implies $a \uparrow_A b \leq_A a' \uparrow_A b'$ and for the remaining rules pertaining to the multiplicative implications we use the adjointness property $a \uparrow_A b \leq_A c$ if $a \leq_A b \uparrow_A c \text{ iff } b \leq_A a \uparrow_A c$. □

**Lemma 10.** There is a layered Heyting algebra $T$ and an interpretation $[\cdot]_T : \text{Prop} \to T$ such that if $\varphi \nvdash \psi$ then $[\varphi]_T \not\leq [\psi]_T$.

**Proof.** We give a Lindenbaum term-algebra construction on the syntax of ILGL with the equivalence relation $\varphi \equiv \psi$ iff $\varphi \vdash \psi$ and $\psi \vdash \varphi$. The set of all such equivalence classes $[\varphi]$ gives the underlying set of the layered Heyting algebra, $T : T_T := [\top]$, $\bot_T := [\bot]$, and $[\varphi] \circ_T [\psi] := [\varphi \circ \psi]$ for $\circ \in \{\land, \lor, \rightarrow, \downarrow, \uparrow\}$.

The fragment $(T, \land_T, \lor_T, \rightarrow_T, \downarrow_T, \uparrow_T)$ forms a bounded distributive lattice with order $[\varphi] \leq_T [\psi]$ iff $[\varphi] \land_T [\psi] = [\varphi]$. It is straightforward to use rules (Ax), $(\land 1)$ and $(\land 2)$ to show that the right hand condition holds iff $\varphi \vdash \psi$. We then obtain adjointness of $\land_T$ and $\rightarrow_T$ from rules $(\rightarrow 1)$ and $(\rightarrow 2)$, monotonicity of $\downarrow_T$ from rule $(\uparrow)$ and the adjointness of $\downarrow_T, \rightarrow_T$ and $\uparrow_T$ from rules $(\rightarrow 1), (\rightarrow 2), (\uparrow 1)$, and $(\uparrow 2)$. Thus $T$ is a layered Heyting algebra with an interpretation given by $[\varphi] = [\varphi]$. By the definition of the ordering, $\varphi \nvdash \psi$ implies $[\varphi] \not\leq_T [\psi]$, as required. □

We now standardly obtain completeness.

**Theorem 3 (Completeness).** For any propositions $\varphi, \psi$ of ILGL, if $[\varphi] \leq [\psi]$ for all interpretations $[\cdot]$ on layered Heyting algebras then $\varphi \vdash \psi$ in ILGL$_{\land A}$. □

We now show that the layered graph semantics is a special case of the algebraic semantics.

**Definition 18 (Preordered layered magma).** A preordered layered magma is a tuple $(X, \leq, \circ)$, with $X$ a set, $\leq$ a preorder on $X$, and $\circ$ a binary partial operation on $X$. □

It is clear that, given a preordered scaffold $(G, E, X, \leq)$, the structure $(X, \leq, \circ_{\mathcal{E}})$ is a preordered layered magma. Analogously to the classical case [6], we can generate a layered Heyting algebra.

**Proposition 9.** Every preordered layered magma generates a layered Heyting algebra.
Proposition 10. For any layered graph model \( \mathcal{M} \) of the graph \( G \), this induces a degree of statefulness to ILGL without changing the underlying semantics. This extension is based on an assignment of a set of resources \( R \) to the vertices of the graph \( G \). That is, each \( r \in R \) is situated at vertices of \( G \). This assignment is denoted \( G[R] \), where we think of \( G \) as the (directed) graph of locations in a system.

Proof. Let \((X, \preceq, \circ)\) be a preordered layered magma. An up-set of the preorder \((X, \preceq)\) is a set \( U \subseteq X \) such that \( x \in U \) and \( x \preceq y \) implies \( y \in U \). Denote the set of all up-sets of \( X \) by \( \text{Up}(X) \). The structure \((\text{Up}(X), \cup, \cap, \rightarrow, \emptyset, X)\) is a Heyting algebra, where \( \rightarrow \) is defined as follows: \( U \rightarrow V := \{ x \in X \mid \text{for all } y(x \preceq y \text{ and } y \in U \text{ implies } y \in V) \} \)

We define the operators \( \uparrow, \downarrow, \blacktriangledown, \blacktriangledown' \) as follows:

\[
\begin{align*}
U \uparrow V &= \{ x \in X \mid \text{there exists } y \in U, z \in V (y \circ z \preceq x) \} \\
U \downarrow V &= \{ x \in X \mid \text{for all } y, z (x \preceq y \text{ and } y \circ z \preceq x) \} \\
U \blacktriangledown V &= \{ x \in X \mid \text{for all } y, z (x \preceq y \text{ and } z \circ y \preceq x) \} \\
U \blacktriangledown' V &= \{ x \in X \mid \text{for all } y \in U (x \preceq y \text{ and } z \circ y \preceq x) \}
\end{align*}
\]

It is straightforward that these all define up-sets, and are thus well-defined. It remains to prove monotonicity of \( \uparrow \) and adjointness of the operators. For monotonicity, let \( U \subseteq U' \), \( V \subseteq V' \) and \( x \in U \uparrow V \). Then there exist \( y \in U \subseteq U' \) and \( z \in V \subseteq V' \) such that \( y \circ z \preceq x \) and \( y \circ z \preceq x \). It follows immediately that \( x \in U' \uparrow V' \).

Next, adjointness. We give just one case, for \( \uparrow \). The others are similar. Suppose \( V \subseteq U \uparrow W \). We must show \( U \uparrow V \subseteq W \). It follows that there exist \( x_0 \in U \) and \( x_1 \in V \) such that \( x_0 \circ x_1 \preceq x \) and \( x_0 \circ x_1 \preceq x \). By assumption, \( x_1 \in U \uparrow W \) and \( x_0 \circ x_1 \preceq x \). It follows that \( x_0 \circ x_1 \in W \). Finally, \( W \) is an up-set, so \( x_0 \circ x_1 \preceq x \) entails \( x \in W \), and the verification is complete.

We can now get the soundness and completeness of the layered graph semantics with respect to ILGL\( _H \) as a special case of the algebraic semantics. Note that a persistent valuation \( \uparrow V : \text{Prop} \rightarrow \varphi(X) \) corresponds uniquely to a valuation \( V' : \text{Prop} \rightarrow \text{Up}(X) \). By definition, for each propositional variable \( p \), \( V(p) \) is an up-set of the preorder \((X, \preceq)\) and trivially an up-set of \((X, \preceq, \circ)\) is an element of \( \varphi(X) \). We can thus use a persistent valuation to generate an interpretation \( \llbracket - \rrbracket_V \) on the layered Heyting algebra generated by \((X, \preceq, @_E)\).

**Proposition 10.** For any layered graph model \( \mathcal{M} \) with valuation \( \uparrow V : \text{Prop} \rightarrow \varphi(X) \) and every formula \( \varphi \) of ILGL, we have \( \llbracket \varphi \rrbracket_V = \{ G \in X \mid G \models_M \varphi \} \in \text{Up}(X) \).

Hence the layered graph semantics of ILGL is a special case of the algebraic semantics and ILGL\( _H \) is sound and complete with respect to the layered graph semantics.

**Proposition 11.** (Equivalence of the Hilbert and tableaux systems), \( \vdash \varphi \) is provable in ILGL\( _H \) iff there is a closed tableau for \( \varphi \).

5 Extension to resources and actions: examples

To express the examples mentioned in Section 1 conveniently and efficiently, we consider an extension of layered graph semantics and ILGL in which we label the ambient graph with resources and consider action modalities (cf. Stirling’s intuitionistic Hennessy–Milner logic [21]) that express resource manipulations. This extension introduces a degree of statefulness to ILGL without changing the underlying semantics.

This extension is based on an assignment of a set of resources \( R \) to the vertices of the graph \( G \). That is, each \( r \in R \) is situated at vertices of \( G \). Such assignments are denoted \( G[R] \), where we think of \( G \) as the (directed) graph of locations in a system.
model. Resources should also carry sufficient structure to allow some basic operations on resource elements. In [16, 5, 4], resources are required to form pre-ordered partial monoids, such as the natural numbers \((\mathbb{N}, \leq, +, 0)\), and we use this approach here. Let \((\mathcal{R}, \subseteq, \circ, e)\) be a resource monoid, where \(\mathcal{R}\) is a collection of sets of resources and \(\circ : \mathcal{R} \times \mathcal{R} \to \mathcal{R}\) is a commutative and associative binary operation. It is easy to see that assignments of resources can be composed and that the algebraic semantics can be easily extended (cf. [6]).

**Lemma 11.** Consider \(\oplus\) and \(\circ\). Both are binary operations with \(\oplus\) non-commutative and \(\circ\) non-associative while \(\circ\) is commutative and associative. A non-commutative, non-associative operation on graphs labelled with resources can be defined.

**Proof.** We have \(\oplus^\mathcal{E} : \mathcal{G} \times \mathcal{G} \to \mathcal{G}\) and \(\circ : \mathcal{R} \times \mathcal{R} \to \mathcal{R}\). Define \(\bullet^\mathcal{E} : (\mathcal{G} \times \mathcal{R}) \times (\mathcal{G} \times \mathcal{R}) \to (\mathcal{G} \times \mathcal{R})\) as \((G_1, R_1) \bullet^\mathcal{E} (G_2, R_2) = (G_1 \oplus^\mathcal{E} G_2, R_1 \circ R_2)\). It is clear that \(\bullet^\mathcal{E}\) is both non-commutative and non-associative.

We write \(G[R] \ll G'[R']\) to denote the evident containment ordering on labelled graphs and resources (i.e., \(G'\) is a subgraph of \(G\) and \(R \subseteq R'\)). We assume also a countable set \(\text{Act}\) of actions, with elements \(a\), etc.. Action modalities, \(\langle a\rangle\) and \([a]\) manipulate (e.g., add to, remove from) the resources assigned to the vertices of the graph.

**Definition 19 (Satisfaction in resource-labelled models).** We extend layered graph models to graphs labelled with resources and extend the interpretation of formulae to the action modalities. For a resource monoid \(\mathcal{R}\), a countable set of actions, \(\text{Act}\), and a layered graph model \(\mathcal{M} = (X, \mathcal{V})\) over labelled graphs, with the containment ordering on labelled graphs, we generate the satisfaction relation \(\models^\mathcal{M} X[R] \times \text{Form}\) as

\[
G[R] \models^\mathcal{M} \top \text{ always } \quad G[R] \models^\mathcal{M} \bot \text{ never } \quad G[R] \models^\mathcal{M} p \text{ iff } G[R] \in \mathcal{V}(p)
\]

\[
G[R] \models^\mathcal{M} \varphi \land \psi \text{ iff } G[R] \models^\mathcal{M} \varphi \text{ and } G[R] \models^\mathcal{M} \psi \\
G[R] \models^\mathcal{M} \varphi \lor \psi \text{ iff } G[R] \models^\mathcal{M} \varphi \text{ or } G[R] \models^\mathcal{M} \psi \\
G[R] \models^\mathcal{M} \varphi \rightarrow \psi \text{ iff, for all } G'[R'] \text{ such that } G[R] \ll G'[R'], G'[R'] \models^\mathcal{M} \varphi \text{ implies } G'[R'] \models^\mathcal{M} \psi
\]

\[
G[R] \models^\mathcal{M} [a] \varphi \text{ iff for some well-formed } G'[R'] \text{ such that } G[R] \xrightarrow{a} G'[R'], G[R'] \models^\mathcal{M} \varphi \\
G[R] \models^\mathcal{M} [a] \varphi \text{ iff for all well-formed } G'[R'] \text{ such that } G[R] \xleftarrow{a} G'[R'], G[R'] \models^\mathcal{M} \varphi
\]

We defer the presentation of the metatheory to account for this extension, including proof systems and completeness results, to another occasion. To do so we follow the approach of dynamic epistemic logics [22], wherein the transitions underlying the action modalities correspond to maps between models rather than states. It is clear persistence will not (and should not) hold for action modalities, but at any given model persistence will hold. To extend the tableaux system we should instead take sequences of CSSs, together with a history of actions following similar approaches in the proof theory of Public Announcement Logic [1].
Example 1 (A transportation network). Here we abstract a public transportation network into social and infrastructure layers. For a meeting in the social layer to be quorate, sufficient people (say 50) must attend. To achieve this, there must be buses of sufficient capacity to transport 50 people, represented as resources, to the meeting hall, in the infrastructure layer (see Figures 8 and 9). The formula $\phi_{\text{quorum}}$ denotes a quorate meeting, $\phi_x$ denotes that $x$ number of people are picked up at bus stops, and the arrival of buses of capacity $x$ in the infrastructure layer is denoted by the action modality $\langle \text{bus}_x \rangle$. These actions move $x$ amount of people from the bus stops to the meeting hall in the social layer. Let $\phi_{\text{meeting}}$ assert the existence of a meeting in the social layer, $G_1$. Then, if $G_2$

\[
G_2[R] \models_M (\langle \text{bus}_{25} \rangle(\langle \text{bus}_{35} \rangle)((\phi_{\text{meeting}} \rightarrow \phi_{50}) \rightarrow \phi_{\text{quorum}}))
\]

\[
G_2[R] \models_M (\langle \text{bus}_{40} \rangle((\phi_{\text{meeting}} \rightarrow \phi_{40}) \rightarrow \neg \phi_{\text{quorum}}))
\]

which assert that having two buses available with a total capacity of more than 50 will allow the meeting to proceed, but that a single bus with capacity 40 will not.

Example 2 (A security barrier). This example (see Figure 10) is a situation highlighted by Schneier [20], wherein a security system is ineffective because of the existence of a side-channel that allows a control to be circumvented. The security policy, as expressed in the security layer, with graph $G_1$, requires that a token be possessed in order to pass
from the outside to the inside; that is, \(\langle \text{pass} \rangle (\phi_{\text{inside}} \rightarrow \phi_{\text{token}})\). However, in the routes layer, with graph \(G_2\), it is possible to perform an action \(\langle \text{swerve} \rangle\) to drive around the gate, as shown in the Figure 11; that is,

\[ G_1 \otimes G_2 \models_M (\langle \text{pass} \rangle (\phi_{\text{inside}} \rightarrow \phi_{\text{token}}) \gg (\langle \text{swerve} \rangle (\phi_{\text{inside}} \land \neg \phi_{\text{token}}))) \]

Thus we can express the mismatch between the security policy and architecture to which it is intended to apply.

**Example 3 (An organizational security architecture).** Our final example concerns an organization which internally has high- and low-security parts of its network. It also operates mobile devices that are outside of its internal network but able to connect to it. Figure 12 illustrates our layered graph model of this set-up. We can give a characterization in ILGL of a side channel that allows a resource from the high-security part of the internal network to transfer to the low-security part via the external mobile connection. Associated with the mobile layer are actions that allow the transference of data. We have two local compliance properties, in the high- and low-security parts of the network, respectively: \(\chi_{\text{high}}(r)\) describes compliance with a policy allowing resource in the high-security network and \(\chi_{\text{sec}}(r)\) is a correctness condition that if a resource \(r\) is not permitted in the low-security network, then it is not in it. We take actions copy, download, upload associated with the mobile layer \(G_2\), allowing data to be copied to another location as well as moved down and up \(E\)-edges respectively, with \(\theta(r)\) a compliance property such that \(G_2[R] \models_M \langle \text{copy} \rangle \theta(r)\) in order to copy data \(r\). Now we have that

\[ G_2[R] \models_M \langle \text{download} \rangle ((\chi_{\text{high}}(r) \gg \theta(r)) \land \langle \text{copy} \rangle (\theta(r) \gg \neg \chi_{\text{sec}}(r))) \]

showing that the mobile layer is a side channel that can undermine the policy \(\chi_{\text{sec}}\).

**References**


