Intuitionistic Layered Graph Logic

IJCAI 2017

Simon Docherty

University College London

Wednesday 23rd August 2017

Joint work with David Pym

---

1S. Docherty and D. Pym. Intuitionistic Layered Graph Logic. *Proc. IJCAR 2016*
Complex Systems & Layering
Layering In The Wild

- Complex systems = structures comprised of interconnected and interacting layers.
- Eg: access control systems, biological systems, distributed systems, bus networks\(^2\).
- Common security issue: mismatch between policy and the structure of the system it applies to\(^3\).


Schneier’s Gate

Layering In The Wild

- Complex system = structure comprised of interconnected and interacting layers.
- Eg: access control systems, biological systems, distributed systems, bus networks\(^4\).
- Common security issue: mismatch between policy and the structure of the system it applies to\(^5\).
- We need reasoning tools for this kind of emergent behaviour.

---


A Mathematical Definition Of Layering

Let $G$ be an *ambient* directed graph, $\mathcal{E}$ a non-empty subset of $G$'s edges and $G$ a subgraph.
A Mathematical Definition Of Layering

Let $\mathcal{G}$ be an *ambient* directed graph, $\mathcal{E}$ a non-empty subset of $\mathcal{G}$’s edges and $\mathcal{G}$ a subgraph.
A Mathematical Definition Of Layering

This decomposition determines a *layering composition* operator $\circ$ on subgraphs of $G$. 

![Diagram showing layering composition](image)
Layered Graph Logic

Layered Graph Logic is a logic for reasoning about graph layering\(^6\). \(\ldots\)


Layered Graph Logic

- Layered Graph Logic is a logic for reasoning about graph layering\textsuperscript{6,7} ...
- ...but LGL lacks a completeness theorem for its graph semantics.


Layered Graph Logic

- Layered Graph Logic is a logic for reasoning about graph layering\(^6\)...
- ..but LGL lacks a completeness theorem for its graph semantics.
- Intuitionistic logic does not validate \(p \lor \neg p\): must directly compute witnesses for proofs.

---


Layered Graph Logic

- Layered Graph Logic is a logic for reasoning about graph layering\(^6\)...
- \(\ldots\) but LGL lacks a completeness theorem for its graph semantics.
- **Intuitionistic** logic does not validate \(p \lor \neg p\): must directly compute witnesses for proofs.
- **Intuitionistic layered graph logic** is LGL without \(p \lor \neg p\). Has a completeness theorem for its graph semantics.

---


Intuitionistic Layered Graph Logic
The Logic

\[ \phi ::= p \mid \top \mid \bot \mid \phi \land \phi \mid \phi \lor \phi \mid \phi \rightarrow \phi \mid \phi \overset{\triangleright}{\rightarrow} \phi \mid \phi \overset{\triangleright\triangleright}{\rightarrow} \phi \mid \phi \overset{\triangleright\triangleright\triangleright}{\rightarrow} \phi \]

\[ \triangleright \quad p \mid \top \mid \bot \mid \cdots \mid \phi \rightarrow \phi: \text{ intuitionistic propositional logic.} \]

\[ \triangleright \quad \phi \overset{\triangleright}{\rightarrow} \phi \mid \phi \overset{\triangleright\triangleright}{\rightarrow} \phi \mid \phi \overset{\triangleright\triangleright\triangleright}{\rightarrow} \phi: \text{ non-associative Lambek calculus} \]
The Logic

\[ \phi ::= p \mid \top \mid \bot \mid \phi \land \phi \mid \phi \lor \phi \mid \phi \rightarrow \phi \mid \phi \triangleright \phi \mid \phi \triangleright \phi \]

- \( \triangleright p \mid \top \mid \bot \mid \cdots \mid \phi \rightarrow \phi \): intuitionistic propositional logic.
- \( \triangleright \phi \triangleright \phi \mid \phi \triangleright \phi \mid \phi \triangleright \phi \): non-associative Lambek calculus

\[ \frac{\varphi \vdash \varphi}{(Ax)} \quad \frac{\varphi \vdash \psi \quad \psi \vdash \chi}{\varphi \vdash \chi} \quad \frac{\varphi \vdash \top}{(T)} \quad \frac{\bot \vdash \varphi}{(\bot)} \]

\[ \frac{\varphi \vdash \psi \quad \varphi \vdash \chi}{\varphi \vdash \psi \land \chi} \quad \frac{\varphi_1 \land \varphi_2 \vdash \varphi_i}{(\land_2)} \quad \frac{\varphi_i \vdash \varphi_1 \lor \varphi_2}{(V_1)} \quad \frac{\varphi \vdash \chi \quad \psi \vdash \chi}{\varphi \lor \psi \vdash \chi} \quad \frac{\varphi \vdash \psi \quad \chi \vdash \chi}{(\triangleright)} \]

\[ \frac{\varphi \vdash \psi \land \chi}{\varphi \land \psi \vdash \chi} \quad \frac{\varphi \vdash \psi \rightarrow \chi \quad \psi \vdash \chi}{\varphi \vdash \psi \rightarrow \chi} \quad \frac{\varphi \land \psi \vdash \chi}{\varphi \vdash \psi \rightarrow \chi} \quad \frac{\varphi \vdash \psi \rightarrow \chi \quad \psi \vdash \chi}{\varphi \vdash \psi \rightarrow \chi} \quad \frac{\varphi \vdash \psi \rightarrow \chi \quad \chi \vdash \psi}{\psi \vdash \varphi \rightarrow \chi} \quad \frac{\varphi \vdash \psi \rightarrow \chi \quad \psi \vdash \chi}{\psi \vdash \varphi \rightarrow \chi} \]

7 / 16
Graph Semantics: Separating Conjunction

\[ G \vdash \phi \rightarrow \psi \]

IFF

\[ \exists H \quad \phi \]

\[ \exists K \quad \psi \]

\[ H \land K \]

and

\[ G \]
Graph Semantics: Separating Implication 1

Eg: Schneier’s Gate: \text{Barrier} \models \phi_{\text{ExtraRoute}} \rightarrow \phi_{\text{Insecure}}
Graph Semantics: Separating Implication 2
Modelling A Transportation Network
Modelling A Transportation Network

- $\phi_{\text{meeting}} \equiv \"there is a meeting at the destination\"
- $\phi_x \equiv \"buses pick up x people at the bus stops\"
- Goal of system: $\phi_{\text{quorum}} \equiv \" \geq 50 \text{ people attend the meeting} \"
Modelling A Transportation Network

- $\phi_{\text{meeting}} \equiv "\text{there is a meeting at the destination}"$
- $\phi_x \equiv "\text{buses pick up x people at the bus stops}"$
- Goal of system: $\phi_{\text{quorum}} \equiv "\geq 50 \text{ people attend the meeting}"$
- $(\phi_{\text{meeting}} \Rightarrow \phi_{60}) \rightarrow \phi_{\text{quorum}}$ expresses that buses of joint capacity of 60 are sufficient to make the meeting quorate.
Modelling A Transportation Network

- $\phi_{\text{meeting}} \equiv "\text{there is a meeting at the destination}"$
- $\phi_x \equiv "\text{buses pick up } x \text{ people at the bus stops}"$
- Goal of system: $\phi_{\text{quorum}} \equiv "\geq 50 \text{ people attend the meeting}"
- $(\phi_{\text{meeting}} \triangleright \phi_{60}) \rightarrow \phi_{\text{quorum}}$ expresses that buses of joint capacity of 60 are sufficient to make the meeting quorate.
- $(\phi_{\text{meeting}} \triangleright \phi_{40}) \rightarrow \neg \phi_{\text{quorum}}$ expresses that buses of capacity 40 are insufficient to make the meeting quorate.
Modelling A Transportation Network

- $\phi_{\text{meeting}} \equiv \"there is a meeting at the destination\"$
- $\phi_x \equiv \"buses pick up $x$ people at the bus stops\"$
- Goal of system: $\phi_{\text{quorum}} \equiv \"\geq 50$ people attend the meeting\"
- $(\phi_{\text{meeting}} \rightarrow \phi_{\text{60}}) \rightarrow \phi_{\text{quorum}}$ expresses that buses of joint capacity of 60 are sufficient to make the meeting quorate.
- $(\phi_{\text{meeting}} \rightarrow \phi_{\text{40}}) \rightarrow \neg \phi_{\text{quorum}}$ expresses that buses of capacity 40 are insufficient to make the meeting quorate.
- Where this goes: simulation modelling by dynamically evolving distribution of resources in layers.
Metatheory
Labelled Tableaux Proof System

**Tableaux**: tree with labelled formulas and constraints at each node, constructed according to tableaux rules:
**Labelled Tableaux Proof System**

**Tableaux**: tree with **labelled formulas** and **constraints** at each node, constructed according to tableaux rules:

**Condition on branch**

\[ \begin{align*}
F\phi &\quad \triangleright \quad \psi : x \in \mathcal{F} \\
&\quad \quad \quad \quad yz \preceq x \in \mathcal{C} \\
\langle \{F\phi : y\}, \emptyset \rangle &\quad \quad \quad \quad \quad \langle \{F\psi : z\}, \emptyset \rangle \\
\hline
\end{align*} \]

Expand with sets to create new branches
Labelled Tableaux Proof System

**Tableaux:** tree with *labelled formulas* and *constraints* at each node, constructed according to tableaux rules:

Condition on branch

\[ F \phi \implies \psi : x \in \mathcal{F} \]
\[ yz \preceq x \in \overline{C} \]

\[ \frac{\langle \{ F \phi : y \}, \emptyset \rangle}{\langle \{ F \psi : z \}, \emptyset \rangle} \]

Expand with sets to create new branches

**Proof of** \( \phi \): tableaux with root \( F \phi : c_0 \) s.t. every branch inconsistent.
Theorem

\( \phi \) is tableaux-provable iff \( \phi \) is valid in layered graph semantics.

- **Idea**: nonexistence of tableau proof of \( \phi \) entails infinite tableau. Data on infinite branch determines graph countermodel of \( \phi \).

---

Metatheory

Theorem
\( \phi \) is tableaux-provable iff \( \phi \) is valid in layered graph semantics.

- **Idea**: nonexistence of tableau proof of \( \phi \) entails infinite tableau. Data on infinite branch determines graph countermodel of \( \phi \).
- Only a semi-decision procedure: terminates if provable, does not terminate if unprovable. However...

---

\(^8\)S. Docherty and D. Pym. Intuitionistic Layered Graph Logic: Semantics and Proof Theory. Forthcoming.
Metatheory

Theorem

\( \phi \) is tableaux-provable iff \( \phi \) is valid in layered graph semantics.

- **Idea:** nonexistence of tableau proof of \( \phi \) entails infinite tableau. Data on infinite branch determines graph countermodel of \( \phi \).
- **Only a semi-decision procedure:** terminates if provable, does not terminate if unprovable. However...

Theorem

\( ILGL \) is decidable.\(^8\)

- **Idea:** by analysis of the algebraic structure of the logical syntax.

---

\(^8\)S. Docherty and D. Pym. Intuitionistic Layered Graph Logic: Semantics and Proof Theory. Forthcoming.
Conclusions
Conclusions

- ILGL is a decidable logic for reasoning about structure of complex systems with a view to describing emergent behaviour.
- Sound and complete for graph theoretic semantics and labelled tableaux system.
Further Work

- Adapt completeness argument to classical LGL.
- Dynamic extensions to the logic for specific modelling purposes:
  - Eg. Agent logics (epistemic, PAL, deontic)
  - Eg. Dynamic logics for cyber-physical systems
- Test modelling capability with real systems.
- Tool implementation (eg: provers and simulation modelling).
- Other applications of graphs?

---