Graph Games and Logic Design (Part III)

Dazhu Li

Institute of Philosophy, Chinese Academy of Sciences;

Department of Philosophy, University of Chinese Academy of Sciences

lidazhu@ucas.ac.cn

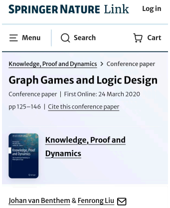
中国科学院大学
University of Chinese Academy of Sciences

CLAWS 2025, Shantou, Oct 25, 2025

Introduction

Graph game is a concise device that can capture many scenarios with interactions between agents.

The talks of Johan and Fenrong



Knowledge, Proof and Dynamics > Conference paper

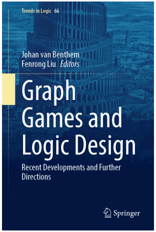
Graph Games and Logic Design

Conference paper | First Online: 24 March 2020

pp 125–146 | Cite this conference paper

Knowledge, Proof and Dynamics

Johan van Benthem & Fenrong Liu



Trends in Logic 66

Johan van Benthem
Fenrong Liu *Editors*

Graph
Games and
Logic Design

Recent Developments and Further Directions

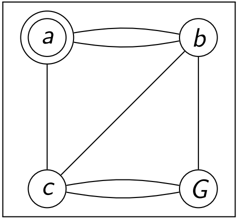
Springer

This talk:

- Applications of the sabotage-style graph game to concrete scenarios
- More on the epistemic logic for the cops and robbers game

Sabotage games as a learning model

Recall the sabotage game:



A model for the interaction between Learner (Traveler) and Teacher (Blocker):

the starting position	⇔	background knowledge
the goal state	⇔	the goal to learn
other states	⇔	possible lemmas
links	⇔	possible inferences (conjectured by Learner)
movements	⇔	inferences
link deletion	⇔	elimination of incorrect inferences

Different winning conditions, depending whether Learner is eager to learn and Teacher is helpful [Gierasimczuk et al., 2009].

Basic features of a learning scenario

- Learner may not only have wrong inferences, but also ignore correct inferences, which can be pointed out by Teacher.
- Different kinds of mistakes:

  actual mistakes made in the learning process, and potential mistakes to be avoided
- The history of the learning matters:

  correcting actual mistakes acts on the history so far (and makes all further moves on that history suspect), while eliminating potential mistakes affects the future from the current point.
- A stricter goal:

  reaching the goal with a history not containing actual mistakes, and

  (reaching the goal with actual mistakes is a Gettier case [Gettier, 1963], which does not mean winning).

A game for learning (GL)

A GL is played by Learner and Teacher on a graph with two relations R_L (visible to Learner) and R_T (correct inferences).

Actions: In each round,

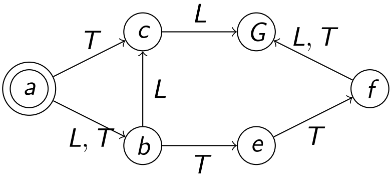
- Learner moves to a successor via an available R_L-link.
- Teacher acts out one of the following four choices:
  - Do nothing;
  - Delete a wrong inference that is not in Learner's path;
  - Delete a wrong inference (u, v) from Learner's current path S, and move Learner back $S|_{(u,v)}$;
  - Make a correct inference visible to Learner.

Winning condition:

Both of them win iff Learner moves to the goal node with an R_T-path.

Example 1 (A GL)

Consider the following graph:



where a is the start-node, and G is the goal-node. It can be shown that the players have a winning strategy.

Observation: The players can win only if there exists an R_T-sequence from the start-node to the goal node, but the existence of such a sequence cannot guarantee that they can win.

Correspondence between theorem proving and GL

- Axioms $\Leftrightarrow$ Start-node
- Theorem $\Leftrightarrow$ Goal node
- Lemmas $\Leftrightarrow$ Other states
- Correct inference from a to b $\Leftrightarrow$ R_T-edge (a, b)
- Inference from a to b conjectured by Learner $\Leftrightarrow$ R_L-edge (a, b)
- Inferring b from a $\Leftrightarrow$ Transition from a to b

Correspondence between theorem proving and GL

- Proof for a $\Leftrightarrow$ R_L-sequence beginning with the start-node and ending with a
- Correct proof for a $\Leftrightarrow$ R_L-sequence S beginning with the starting node, ending with a and $Set(S) \subseteq R_T$
- Giving a counterexample to the inference from a to b in the proof S $\Leftrightarrow$ Modifying S to $S|_{(a,b)}$ $((a, b) \in Set(S))$
- Giving a counterexample to the conjectured inference from a to b not in the proof S $\Leftrightarrow$ Removing (a, b) from R_L $((a, b) \notin Set(S))$
- Pointing out a correct inference from a to b not conjectured by Learner before $\Leftrightarrow$ Extending R_L with $(a, b) \in R_T$

A logic of the game for learning (LGL)

Definition 2 (Language $\mathcal{L}_{\mathsf{LGL}}$ of LGL)

Let P be a countable set of propositional atoms. The formulas of $\mathcal{L}_{\mathsf{LGL}}$ are recursively defined in the following way:

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \blacklozenge\varphi \mid \langle - \rangle_{on}\varphi \mid \langle - \rangle_{off}\varphi \mid \langle + \rangle\varphi$$

where $p \in \mathrm{P}$. $\top$, $\bot$, $\vee$, $\rightarrow$ and $\leftrightarrow$ are as usual. $\blacksquare$, $[-]_{on}$, $[-]_{off}$ and $[+]$ are dual operators of $\blacklozenge$, $\langle - \rangle_{on}$, $\langle - \rangle_{off}$ and $\langle + \rangle$ respectively.

Definition 3 (Models and pointed models)

Models: $\mathcal{M} = (W, R_L, R_T, V)$, where

- $W \neq \emptyset$ is a set of possible worlds;
- $R_{i\in\{L,T\}} \subseteq W \times W$ are two binary relations;
- $V : \mathrm{P} \to 2^W$ is a valuation function.

Pointed models: $(\mathcal{M}, S)$, where $\mathcal{M}$ is a model and S is an R_L-sequence.

A logic of the game for learning (LGL)

$\mathcal{M} \ominus (u, v) := (W, R_L \setminus \{(u, v)\}, R_T, V) \quad \mathcal{M} \oplus (u, v) := (W, R_L \cup \{(u, v)\}, R_T, V)$

Definition 4 (Semantics of LGL)

Let $(\mathcal{M}, S)$ be a pointed model and $\varphi \in \mathcal{L}_{\mathsf{LGL}}$:

$$\begin{array}{rcl}
\mathcal{M}, S \vDash p & \Leftrightarrow & e(S) \in V(p) \\
\mathcal{M}, S \vDash \neg\varphi & \Leftrightarrow & \mathcal{M}, S \nvDash \varphi \\
\mathcal{M}, S \vDash \varphi \wedge \psi & \Leftrightarrow & \mathcal{M}, S \vDash \varphi \text{ and } \mathcal{M}, S \vDash \psi \\
\mathcal{M}, S \vDash \blacklozenge\varphi & \Leftrightarrow & \exists v \in W \text{ s.t. } R_L e(S) v \text{ and } \mathcal{M}, S; v \vDash \varphi \\
\mathcal{M}, S \vDash \langle - \rangle_{on}\varphi & \Leftrightarrow & \exists (u, v) \in Set(S) \setminus R_T \text{ s.t. } \mathcal{M} \ominus (u, v), S|_{(u,v)} \vDash \varphi \\
\mathcal{M}, S \vDash \langle - \rangle_{off}\varphi & \Leftrightarrow & \exists (u, v) \in (R_L \setminus R_T) \setminus Set(S) \text{ s.t. } \mathcal{M} \ominus (u, v), S \vDash \varphi \\
\mathcal{M}, S \vDash \langle + \rangle\varphi & \Leftrightarrow & \exists (u, v) \in R_T \setminus R_L \text{ s.t. } \mathcal{M} \oplus (u, v), S \vDash \varphi
\end{array}$$

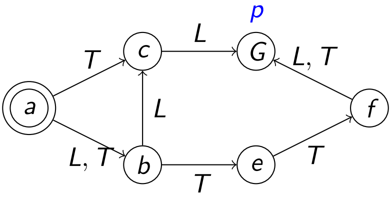
LGL: validities with respect to the pointed models whose sequences are singletons.

Applications to GL: winning positions (finite cases)

LGL is expressive enough to characterize player's actions in GL.

LGL is able to describe winning positions in finite games:

Consider the above GL example again:



Let p hold only at the goal-node G. Then the following formula holds at a:

$$\blacklozenge\langle+\rangle\blacklozenge\langle+\rangle\blacklozenge\langle-\rangle_{on}\blacklozenge\langle-\rangle_{off}\blacklozenge\blacklozenge(p \wedge [-]_{on}\bot)$$

Logical validities

LGL: validities with respect to the pointed models whose sequences are singletons.

$$[-]_{on}\bot$$

$$p \wedge \blacklozenge\top \to \blacksquare[-]_{on} p$$

$$\bigcirc(\varphi \to \psi) \to (\bigcirc\varphi \to \bigcirc\psi) \quad (\bigcirc \in \{[-]_{off}, [+]\})$$

$$\blacklozenge^{n}\langle - \rangle_{on}\varphi \to \bigvee_{m<n} \blacklozenge^{m}\langle - \rangle_{off}\varphi \quad (1 \leq n \in \mathbb{N})$$

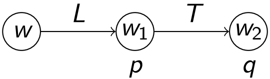
Proposition 1

Validity of LGL *is not closed under substitution.*

Example 5

$p \wedge \blacklozenge \top \to \blacksquare[-]_{on} p$ is valid.

Consider its schema $\varphi \wedge \blacklozenge \psi \to \blacksquare[-]_{on}\varphi$. Let $\varphi := \blacklozenge p$ and $\psi := \blacksquare q$.



It holds that $\mathcal{M}, w \vDash \blacklozenge p \wedge \blacklozenge \blacksquare q$. But, since w has exactly one R_L-successor w_1 and $(w, w_1) \notin R_T$, we have $\mathcal{M}, w \nvDash \blacksquare[-]_{on} \blacklozenge p$.

Expressivity of LGL

- First-order translation
- Bisimulation
- Characterization theorem

Bisimulation

For any two models $\mathcal{M} = (W, R_L, R_T, V)$ and $\mathcal{M}' = (W', R'_L, R'_T, V')$, a non-empty relation Z_l is an l-bisimulation between the two pointed models $(\mathcal{M}, S)$ and $(\mathcal{M}, S')$ (notation: $(\mathcal{M}, S) Z_l (\mathcal{M}', S')$) if the following conditions are satisfied:

Bisimulation

Atom: $\mathcal{M}, S \vDash p$ iff $\mathcal{M}', S' \vDash p$, for each $p \in \mathsf{P}$.

Zig$_{\blacklozenge}$: If there exists $v \in W$ such that $R_L e(S) v$, then there exists $v' \in W'$ such that $R'_L e(S') v'$ and $(\mathcal{M}, S; v) Z_l (\mathcal{M}', S'; v')$.

Zig$_{\langle - \rangle_{on}}$: If there exists $(u, v) \in Set(S) \setminus R_T$, then there exists $(u', v') \in Set(S') \setminus R'_T$ such that

$$(\mathcal{M} \ominus (u, v), S|_{(u,v)}) Z_l (\mathcal{M}' \ominus (u', v'), S'|_{(u',v')}).$$

Zig$_{\langle - \rangle_{off}}$: If there exists $(u, v) \in (R_L \setminus R_T) \setminus Set(S)$, then there exists $(u', v') \in (R'_L \setminus R'_T) \setminus Set(S')$ such that

$$(\mathcal{M} \ominus (u, v), S) Z_l (\mathcal{M}' \ominus (u', v'), S').$$

Zig$_{\langle + \rangle}$: If there exists $(u, v) \in R_T \setminus R_L$, then there exists $(u', v') \in R'_T \setminus R'_L$ such that $(\mathcal{M} \oplus (u, v), S) Z_l (\mathcal{M}' \oplus (u', v'), S')$.

Zag$_{\blacklozenge}$, Zag$_{\langle - \rangle_{on}}$, Zag$_{\langle - \rangle_{off}}$ and **Zag$_{\langle + \rangle}$:** Analogous clauses in the converse of **Zig$_{\blacklozenge}$, Zig$_{\langle - \rangle_{on}}$, Zig$_{\langle - \rangle_{off}}$** and **Zig$_{\langle + \rangle}$** respectively.

Bisimulation and characterization theorem

Theorem 6

For any two pointed models, if they are l-bisimilar, then they are modally equivalent.

Theorem 7

For any two ω-saturated pointed models, if they are modally equivalent, then they are l-bisimilar.

Theorem 8

For any $\alpha(x) \in \mathcal{L}_1$ with only one free variable, $\alpha(x)$ is equivalent to the translation of some $\mathcal{L}_{\mathsf{LGL}}$-formula φ iff $\alpha(x)$ is invariant under l-bisimulation.

Therefore, in terms of the expressiveness, LGL is as powerful as the one free variable fragment of first-order logic that is invariant for l-bisimulation.

Model checking and satisfiability for LGL

Theorem 9

LGL *enjoys neither the tree model property nor the finite model property.*

Theorem 10

The satisfiability problem for LGL *is undecidable.*

Theorem 11

Model checking for LGL *is PSPACE-complete.*

Upper bound: first-order translation

Lower bound: bridge modal logic [Areces et al., 2015]

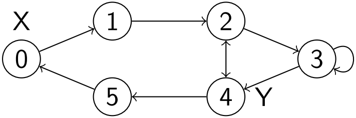
Let $\mathcal{M} = (W, R_L, W \times W, V)$ be a model and $\varphi \in \mathcal{L}_{\blacklozenge\langle+\rangle}$:

$$\mathcal{M}, S \vDash \varphi \Leftrightarrow (W, R_L, V), e(S) \vDash \varphi^\star,$$

where $\varphi^\star$ is a formula obtained by replacing $\blacklozenge$ in φ with $\Diamond$.

A scenario with restricted observation

Initially, Cop X (female) is at 0, and Robber Y (male) is at 4. They know the graph structure and their own positions. A player can see the other if they are at the same position or at a vertex reachable by an arrow in either direction.

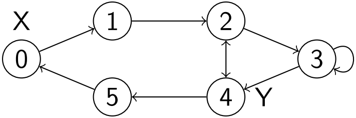


- Thus, X and Y do not know each other's exact positions.
- X knows that Y must be at 2, 3, or 4, while Y knows that X must be at 0 or 1.

Logic design: basic ideas

- We use values for nodes in a graph, which together with the binary relation give us a FOL structure.
- We use variables for players, then the current position of a player gives us the value of that variable.
- So, positions of all players can give us an assignment σ that assigns values to variables, say, if player x is at a, then $\sigma(x) = a$.

Revisiting the example, we have $\sigma(x) = 0$, $\sigma(y) = 4$.



Epistemic logic for cops and robbers (ELCR)

We fix a vocabulary $Voc = (Pred, Cons, Var)$:

- $Pred$ is a set of predicate symbols, containing a special binary relation R describing the edges of the graph of a game.
- $Cons$ is a non-empty, finite set of constants.
- $Var = \{x, y\}$ are the two variables, meaning the players.

Definition 12

Formulas in *the language $\mathcal{L}$ for* ELCR are defined as follows:

$$\begin{gathered}\mathcal{L}_{\mathrm{B}} \ni \alpha ::= P\boldsymbol{t} \mid t_1 \equiv t_2 \mid \neg\alpha \mid (\alpha \wedge \alpha) \\ \mathcal{L}_{\mathrm{BD}} \ni \psi ::= \alpha \mid \neg\psi \mid (\psi \wedge \psi) \mid [z]\psi \\ \mathcal{L} \ni \varphi ::= \psi \mid K_z t \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_z\psi \mid [z]\varphi\end{gathered}$$

where $t, t_1, t_2 \in Cons \cup Var$ are terms, $\boldsymbol{t}$ is a tuple of terms, $P \in Pred$ is a predicate symbol, and $z \in Var = \{x, y\}$ is a variable.

$\langle K_z\rangle\varphi := \neg K_z\neg\varphi, \quad \langle z\rangle\varphi := \neg[z]\neg\varphi, \quad K_z\mathcal{T} := \bigwedge_{t\in\mathcal{T}} K_z t \;\; (\mathcal{T} \subseteq Cons \cup Var)$.

Models

Definition 13

A *k-sight model for* ELCR is a tuple $M = (\mathbf{D}, \mathbf{I}, \Sigma, \sim)$, where

- $\mathbf{D}$ is a non-empty, finite set of values (also called vertices or positions).
- $\mathbf{I}$ is the interpretation function s.t.
    - $\mathbf{I}(P) \subseteq \mathbf{D}^m$ is an m-ary relation on $\mathbf{D}$. $\mathbf{I}(R)$ is a binary relation on $\mathbf{D}$ s.t. for any $s \in \mathbf{D}$, there is some $t \in \mathbf{D}$ s.t. $(s, t) \in \mathbf{I}(R)$.
    - $\mathbf{I}(c) \in \mathbf{D}$, for $c \in Cons$. Moreover, for each $s \in \mathbf{D}$, there is $c \in Cons$ s.t. $\mathbf{I}(c) = s$.
- $\Sigma \subseteq \mathbf{D}^{Var}$ is a non-empty set of *situations* of the players' positions, also called *assignments*.
- $\sim_z \subseteq \Sigma \times \Sigma$ is an equivalence relation s.t. for all $\sigma, \sigma' \in \Sigma$, if $\sigma \sim_z \sigma'$, then for all $z' \in Var$ with $\sigma(z') \in \mathbb{D}^k(\sigma(z))$, $\sigma(z') = \sigma'(z')$.

Intuitively, $u \in \mathbb{D}^k(v)$ means that from v one can reach vertex u within k steps via $\mathbf{I}(R)$ or its converse relation, which can be defined precisely.

Truth conditions for the static part

Given a model $M = (\mathbf{D}, \mathbf{I}, \Sigma, \sim)$ and a situation $\sigma \in \Sigma$, we use $t^{(\mathbf{I},\sigma)}$ for the value of $t \in \mathit{Var} \cup \mathit{Cons}$.

Truth conditions (without $[z]\varphi$):

$$\begin{aligned}
M, \sigma \vDash P(t_1, \ldots, t_n) \quad &\text{iff} \quad (t_1^{(\mathbf{I},\sigma)}, \ldots, t_n^{(\mathbf{I},\sigma)}) \in \mathbf{I}(P) \\
M, \sigma \vDash t_1 \equiv t_2 \quad &\text{iff} \quad t_1^{(\mathbf{I},\sigma)} = t_2^{(\mathbf{I},\sigma)} \\
M, \sigma \vDash K_z t \quad &\text{iff} \quad \text{for all } \sigma' \sim_z \sigma,\ t^{(\mathbf{I},\sigma)} = t^{(\mathbf{I},\sigma')} \\
M, \sigma \vDash K_z \varphi \quad &\text{iff} \quad \text{for all } \sigma' \sim_z \sigma,\ M, \sigma' \vDash \varphi
\end{aligned}$$

Some validities

(1) $t_1 \equiv t_2 \to (\alpha \leftrightarrow \alpha[t_1/t_2])$, given $\alpha \in \mathcal{L}_\mathrm{B}$ — Closed for B-formulas

(2) $\bigvee_{c \in Cons} z \equiv c$, $z \in \{x, y\}$ — At some where

(3) $t \equiv c \to (K_z t \leftrightarrow K_z t \equiv c)$, $z \in \{x, y\}$, t is a term. — De re knowledge

(4) $(K_z t_1 \wedge K_z t_2 \wedge R t_1 t_2) \to K_z R t_1 t_2$ — Knowing the graph

(5) $\mathrm{D}^k z t \to K_z t$ — k-sight ability

(6) $K_x y = \mathcal{T} \to (K_x \alpha \leftrightarrow \bigwedge_{c \in \mathcal{T}} \alpha[c/y])$, where $\mathcal{T} \subseteq Cons$ and $\alpha \in \mathcal{L}_\mathrm{B}$. — Grounded knowledge

$\mathrm{D}^k z t$ is a predicate symbol for $\mathbb{D}^k$, which is definable with R, due to the finiteness of $Var \cup Cons$:

$$\mathrm{D}^0 t_1 t_2 := t_1 \equiv t_2$$

$$\mathrm{D}^{n+1} t_1 t_2 := \mathrm{D}^n t_1 t_2 \vee \bigvee_{t \in Cons \cup Var} (\mathrm{D}^n t_1 t \wedge (R t t_2 \vee R t_2 t))$$

$K_x y = \mathcal{T}$: $\mathcal{T}$ is the collection of y's possible positions considered by x. It is also definable in the language since $Cons$ is finite.

Dynamic updates

Consider $[z]\varphi$, we need to capture the following:

(1) Player z moves, so the situation of the game changes.

(2) The movements affect all players' knowledge.

One assumption: when a player acts, the other would know.

Dynamic updates

For (1), we define relations $\mathsf{R}^{z \in Var}$ on assignments $\mathbf{D}^{Var}$:

For any $\sigma, \sigma' \in \mathbf{D}^{Var}$, we write $\mathsf{R}^z\sigma\sigma'$ if $(\sigma(z), \sigma'(z)) \in \mathbf{R}$ and for $z' \in Var \setminus \{z\}$, $\sigma(z') = \sigma'(z')$.

When $\mathsf{R}^z\sigma\sigma'$, the only difference between σ and σ' is the values of z: z value in σ' is an **R**-successor of z value in σ, which describes that after the movement of z from $\sigma(z)$ to $\sigma'(z)$, σ becomes σ'.

Given a class $\Sigma \subseteq \mathbf{D}^{Var}$ of situations and a variable z, let

$$\mathsf{R}^z(\Sigma) := \{\sigma' \mid \textit{there is } \sigma \in \Sigma \textit{ s.t. } \mathsf{R}^z\sigma\sigma'\}.$$

When Σ is a singleton $\{\sigma\}$, we write $\mathsf{R}^z(\sigma)$ for $\mathsf{R}^z(\{\sigma\})$.

Truth condition for dynamic operators

Definition 14

Assuming both players have the same sight k,

$$(\mathbf{D}, \mathbf{I}, \Sigma, \sim), \sigma_1 \models [z]\varphi \quad \text{iff} \quad \text{for all } \sigma_2 \in \mathrm{R}^z(\sigma_1),\ (\mathbf{D}, \mathbf{I}, \Sigma', \sim'), \sigma_2 \models \varphi,$$

where Σ' is given by the following:

(a) if $\sigma_2(x) \in \mathbb{D}^k(\sigma_2(y))$, then $\Sigma' = \{\sigma_2\}$,
[for the case that they are in the sight of each other]

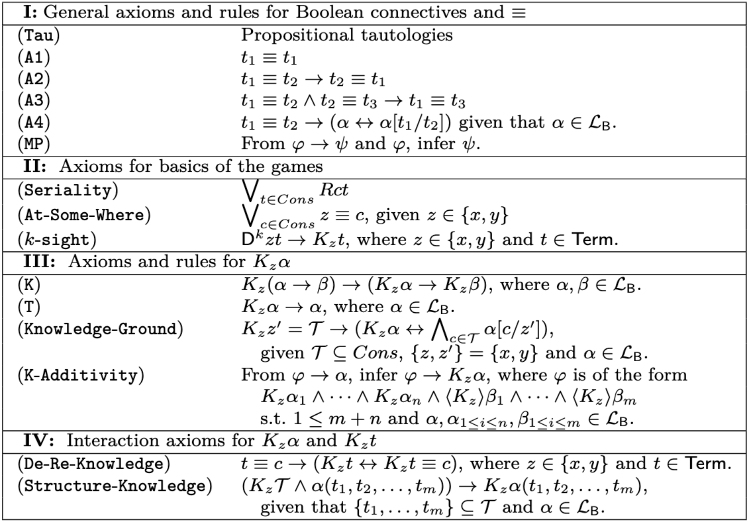
(b) if $\sigma_2(x) \notin \mathbb{D}^k(\sigma_2(y))$, then $\Sigma' = \{\sigma' \in \mathrm{R}^z(\Sigma) \mid \sigma'(x) \notin \mathbb{D}^k(\sigma'(y))\}$,
[for the case that they are not in the sight of each other]

and the new relations $\sim'_{z \in \{x,y\}}$ on Σ' are obtained by:

(c) $\sigma'_1 \sim'_z \sigma'_2$ iff $\sigma'_1(z) = \sigma'_2(z)$.

The resulting model is also a k-sight model.

Axiomatization: the static part



I: General axioms and rules for Boolean connectives and $\equiv$	
(Tau)	Propositional tautologies
(A1)	$t_1 \equiv t_1$
(A2)	$t_1 \equiv t_2 \to t_2 \equiv t_1$
(A3)	$t_1 \equiv t_2 \wedge t_2 \equiv t_3 \to t_1 \equiv t_3$
(A4)	$t_1 \equiv t_2 \to (\alpha \leftrightarrow \alpha[t_1/t_2])$ given that $\alpha \in \mathcal{L}_{\mathsf{B}}$.
(MP)	From $\varphi \to \psi$ and φ, infer ψ.
II: Axioms for basics of the games	
(Seriality)	$\bigvee_{t \in Cons} Rct$
(At-Some-Where)	$\bigvee_{c \in Cons} z \equiv c$, given $z \in \{x, y\}$
(k-sight)	$\mathsf{D}^k zt \to K_z t$, where $z \in \{x, y\}$ and $t \in \mathsf{Term}$.
III: Axioms and rules for $K_z \alpha$	
(K)	$K_z(\alpha \to \beta) \to (K_z \alpha \to K_z \beta)$, where $\alpha, \beta \in \mathcal{L}_{\mathsf{B}}$.
(T)	$K_z \alpha \to \alpha$, where $\alpha \in \mathcal{L}_{\mathsf{B}}$.
(Knowledge-Ground)	$K_z z' = \mathcal{T} \to (K_z \alpha \leftrightarrow \bigwedge_{c \in \mathcal{T}} \alpha[c/z'])$, given $\mathcal{T} \subseteq Cons$, $\{z, z'\} = \{x, y\}$ and $\alpha \in \mathcal{L}_{\mathsf{B}}$.
(K-Additivity)	From $\varphi \to \alpha$, infer $\varphi \to K_z \alpha$, where φ is of the form $K_z \alpha_1 \wedge \cdots \wedge K_z \alpha_n \wedge \langle K_z \rangle \beta_1 \wedge \cdots \wedge \langle K_z \rangle \beta_m$ s.t. $1 \leq m + n$ and $\alpha, \alpha_{1 \leq i \leq n}, \beta_{1 \leq i \leq m} \in \mathcal{L}_{\mathsf{B}}$.
IV: Interaction axioms for $K_z \alpha$ and $K_z t$	
(De-Re-Knowledge)	$t \equiv c \to (K_z t \leftrightarrow K_z t \equiv c)$, where $z \in \{x, y\}$ and $t \in \mathsf{Term}$.
(Structure-Knowledge)	$(K_z \mathcal{T} \wedge \alpha(t_1, t_2, \ldots, t_m)) \to K_z \alpha(t_1, t_2, \ldots, t_m)$, given that $\{t_1, \ldots, t_m\} \subseteq \mathcal{T}$ and $\alpha \in \mathcal{L}_{\mathsf{B}}$.

Theorem 15

The calculus is sound and strongly complete with respect to the static part of ELCR*.*

Key idea of the completeness proof:

- We use canonical construction, with maximal consistent sets serving as assignments.
- A crucial aspect is to ensure that the resulting canonical model is a *k*-sight model. For instance,
  - The players should have the *k*-sight ability.
  - The resulting class of maximal consistent sets Δ should be a set of assignments, e.g.,

$$(x \equiv c_1 \wedge y \equiv c_2) \in \Delta_1 \cap \Delta_2 \Rightarrow \Delta_1 = \Delta_2.$$

Axiomatization for the full language

Key idea to deal with dynamic operators:

Encode their information with static formulas.

- Similar to the case of "recursion axioms" in DEL
- A crucial difference: updates induced by the dynamic operators are not functional, and there is no general equivalence transferring $[z]\neg\varphi$ into the form $\neg[z]\varphi$.

Our strategy:

> *Given $[z]\varphi$, we can characterize the region where z can move to and the uncertainty scopes of the players, which help us to change $[z]\varphi$ into an equivalent ψ without dynamic operators.*

Axiomatization for the full language

An example:

$$[x]K_xy \leftrightarrow \bigwedge_{\mathcal{T},\mathcal{T}_1 \subseteq Cons} (K_xy = \mathcal{T} \wedge Rx = \mathcal{T}_1 \rightarrow$$

Assuming that x knows that y is in the region $\mathcal{T}$, and the set of successors of x is $\mathcal{T}_1$,

$$\bigwedge_{a \in \mathcal{T}_1} (\mathsf{D}^k ay \vee \bigwedge_{b \in \mathcal{T}} (b \not\equiv y \rightarrow \mathsf{D}^k ab)))$$

for any new position a of x,

either y is in the sight of x from a (so x knows where y is), or

all other possibilities of y's position are in the sight of x from a

(so x can rule out those possibilities and then knows where y is).

Axiomatization for the Full Language

Let $z \in \{x, y\}$, $c \in \mathit{Cons}$ and $\mathcal{T} \subseteq \mathit{Cons}$.

$$\begin{aligned}(\alpha)_{z\Rightarrow c} :=& \alpha[c/z]\\ (\alpha)^{(\mathcal{T},x)}_{x\Rightarrow c} :=& (\mathrm{D}^k cy \to \alpha[c/x]) \wedge \\ & (\neg \mathrm{D}^k cy \to \bigwedge_{c' \in \mathcal{T}} (\neg \mathrm{D}^k cc' \to \alpha[c/x, c'/y]))\end{aligned}$$

Similarly we can define $(\alpha)^{(\mathcal{T},y)}_{y\Rightarrow c}$.

$$\begin{aligned}(\alpha)^{(\mathcal{T},x)}_{y\Rightarrow c} :=& \alpha[c/y] \wedge (\mathrm{D}^k xc \vee \\ & (\neg \mathrm{D}^k xc \wedge \bigwedge_{t_1 \in \mathcal{T}, c_1 \in \mathit{Cons}} (Rt_1 c_1 \wedge \neg \mathrm{D}^k xc_1 \to \alpha[c_1/y])))\end{aligned}$$

Similarly, we can define $(\alpha)^{(\mathcal{T},y)}_{x\Rightarrow c}$.

$$\begin{aligned}(K_y x)^{(\mathcal{T},y)}_{x\Rightarrow c} :=& \mathrm{D}yc \vee \\ & (\neg \mathrm{D}yc \rightarrow \bigwedge_{t_1,t_2\in\mathcal{T},c_1,c_2\in Cons} (Rt_1c_1 \wedge Rt_2c_2 \rightarrow c_1 \equiv c_2))\end{aligned}$$

Similarly, we can define $(K_x y)^{(\mathcal{T},x)}_{y\Rightarrow c}$.

$$\begin{aligned}(K_x y)^{(\mathcal{T},x)}_{x\Rightarrow c} :=& \mathrm{D}^k cy \vee \\ & (\neg \mathrm{D}^k cy \wedge \bigwedge_{t_1\in\mathcal{T}} (\neg \mathrm{D}^k ct_1 \rightarrow t_1 \equiv y))\end{aligned}$$

Similarly, we can define $(K_y x)^{(\mathcal{T},y)}_{y\Rightarrow c}$.

The ‘reduction’

$$[z]\varphi \leftrightarrow \bigwedge_{\mathcal{R},\mathcal{T}_x,\mathcal{T}_y \subseteq Cons} (Rz = \mathcal{R} \wedge K_x y = \mathcal{T}_x \wedge K_y x = \mathcal{T}_y \to \bigwedge_{c \in \mathcal{R}} \varphi^{z \Rightarrow c})$$

where $Rz = \mathcal{R}$ means $\mathcal{R}$ is the set of successors of z (i.e., the possible new positions of z), and $\varphi^{z \Rightarrow c}$ is obtained by the following substitutions:

- Substitute each atomic subformula α of φ not in the knowledge operator with $(\alpha)_{z \Rightarrow c}$.
- Substitute subformulas $K_z\psi$ with $(\psi)^{(\mathcal{T}_z,z)}_{z \Rightarrow c}$.
- Substitute subformulas $K_{z'}\psi$ with $K_{z'}\psi$ with $(\psi)^{(\mathcal{T}_{z'},z')}_{z \Rightarrow c}$, where $\{z, z'\} = \{x, y\}$.
- Replace subformulas $K_x x$ and $K_y y$ with $\top$.
- Replace $K_z z'$ with $(K_z z')^{(\mathcal{T}_z,z)}_{z \Rightarrow c}$ and $K_{z'} z$ with $(K_{z'} z)^{(\mathcal{T}_{z'},z')}_{z \Rightarrow c}$, where $\{z, z'\} = \{x, y\}$.

Technical Results

Theorem 16

The proof system is sound and strongly complete with respect to the logic.

Theorem 17

The satisfiability problem for the logic is decidable.

A product model approach

van Benthem [2025]: one can study the update with DEL product models, and basic ideas for event models:

- We need event models for movements of both Robber and Cop.
- Events can be given by edges $(s, t) \in \mathbf{I}(R)$ of the graph: (s, t) means the movement of, e.g., Cop, from s to t.
- Uncertainty relations among these movements are based on what regions can be observed by players (so their positions and observation ability are crucial).

A product model approach

Let $M = (\mathbf{D}, \mathbf{I}, \Sigma, \sim)$ be a k-sight model and $\sigma \in \Sigma$. We only present the event models for Cop, and those for Robber are analogous.

An event model for Cop x, associated with σ, is the following:

$$E_\sigma^x = (\mathbf{I}(R), \approx, \{pre_{(s,t)} \mid (s,t) \in \mathbf{I}(R)\}, \{post_{(s,t)} \mid (s,t) \in \mathbf{I}(R)\})$$

where

- $\mathbf{I}(R)$ are arrows in the graph. [They are ‘moves’ (or ‘actions’).]
- For each $(s,t) \in \mathbf{I}(R)$,
  its *pre-condition* $pre_{(s,t)}$ is that Cop is at s, and
  its *post-condition* $post_{(s,t)}$ is that Cop is at t.

  [This means the change of the position is caused by the action.]

The uncertainty relations $\approx_x$ and $\approx_y$ are as follows:

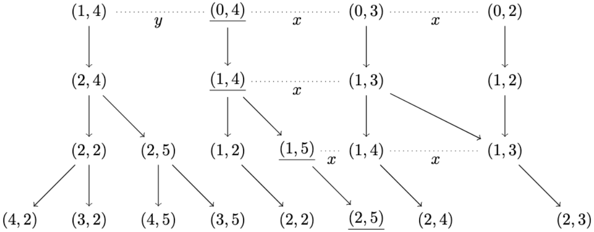
- For Cop x, the case is easy: $\approx_x$ is the identity relation on pairs, since x knows which move she is making.
- For Robber y, we need to consider whether Robber can see the states involved in movements:
    - For those (s, t) s.t. $s, t \in \mathbb{D}^k(\sigma(y))$, $\approx_y$ is the identity relation.
    - For those (s, t) s.t. $s \in \mathbb{D}^k(\sigma(y))$ and $t \notin \mathbb{D}^k(\sigma(y))$,
    $$(s, t) \approx_y (s', t') \quad \text{iff} \quad s = s' \text{ and } t' \notin \mathbb{D}^k(\sigma(y)).$$
    - For those (s, t) s.t. $s \notin \mathbb{D}^k(\sigma(y))$ and $t \in \mathbb{D}^k(\sigma(y))$,
    $$(s, t) \approx_y (s', t') \quad \text{iff} \quad t = t' \text{ and } s' \notin \mathbb{D}^k(\sigma(y)).$$
    - For those (s, t) s.t. $s \notin \mathbb{D}^k(\sigma(y))$ and $t \notin \mathbb{D}^k(\sigma(y))$,
    $$(s, t) \approx_y (s', t') \quad \text{iff} \quad s, t, s', t' \notin \mathbb{D}^k\sigma(y)).$$

Definition 18

Let $M = (\mathbf{D}, \mathbf{I}, \Sigma, \sim)$ be a k-sight model, $\sigma \in \Sigma$ and E^x_σ be the event model associated with σ. The *product update* $(M, \sigma) \times E^x_\sigma$ is a new k-sight model $(\mathbf{D}, \mathbf{I}, \Sigma', \sim')$:

- $\Sigma' = \{(\sigma_1, x, (s, t)) \mid \sigma_1(x) = s,\ \sigma_1 \in \Sigma\}$, where $(\sigma_1, x, (s, t))$ is a new situation such that $(\sigma_1, x, (s, t))(x) = t$ and $(\sigma_1, x, (s, t))(y) = \sigma_1(y)$.
- For any variable $z \in Var$ and situations $(\sigma_1, x, (s_1, t_1)), (\sigma_2, x, (s_2, t_2)) \in \Sigma'$, $(\sigma_1, x, (s_1, t_1)) \sim'_z (\sigma_2, x, (s_2, t_2))$ if, and only if,
  - (*i*) $\sigma_1 \sim_z \sigma_2$, $(s_1, t_1) \approx_z (s_2, t_2)$.
  - (*ii*) If $\sigma_1(x) \in \mathbb{D}^k(\sigma_1(y))$, then $\sigma_1 = \sigma_2$, and if $\sigma_1(x) \notin \mathbb{D}^k(\sigma_1(y))$, then $\sigma_2(x) \notin \mathbb{D}^k(\sigma_2(y))$.

Example revisited (based on product models)



- Dotted links labeled with *x* and *y* are indistinguishability relations
- Solid arrows are successive product updates: Each layer is an epistemic model The first layer is the original M, the second is $M_1 = (M, (0, 4)) \times E^x_{(0,1)}$, the third is $M_2 = (M_1, (1, 4)) \times E^y_{(4,5)}$, and the last layer is $(M_2, (1, 5)) \times E^x_{(1,2)}$.

Comparing the two approaches

- In the resulting model of a product update, the new situations are not necessarily those in our ELCR-approach, and might be larger.
- The direct update shows succinctly how players reason about the situations after facts change.

- Graph games are powerful tools, which can serve as models for interesting scenarios and provide a device for theoretical study in relevant fields.
- Study graph game with a logical approach motivates new designs of logics, which in turn deepens our understanding of graph games.
- Existing logical exploration on graph game usually focuses on the perfect information setting, but the study on the imperfect information version of cops and robbers provides ideas to study other games.

Thanks for your attention!

C. Areces, R. Fervari, and G. Hoffmann. Relation-changing modal operators. *Journal of the IGPL*, 23(4):601–627, August 2015.

E. Gettier. Is justified true belief knowledge? *Analysis*, 23(6):121–123, 1963.

N. Gierasimczuk, L. Kurzen, and F. R. Velázquez-Quesada. Learning and teaching as a game: A sabotage approach. In X. He, J. Horty, and E. Pacuit, editors, *Proceedings of LORI 2009*, volume 5834 of *Lecture Notes in Computer Science*, pages 119–132. Springer, 2009. URL `https://doi.org/10.1007/978-3-642-04893-7_10`.

J. van Benthem. Dynamic-epistemic logic for games. In J. van Benthem and F. Liu, editors, *Graph Games and Logic Design: Recent Developments and Future Directions* (To Appear). Springer, Cham, 2025.