# Knowing Values and Public Inspection

Malvin Gattinger
with Jan van Eijck & Yanjing Wang

arxiv.org/abs/1609.03338
slides at w4eg.de/malvin

2016-10-14, LIRa
ILLC, Amsterdam

# Introduction

Knowing *that* $\heartsuit$ announcing *that*:

$$[!\psi]K_i(\varphi) \;\leftrightarrow\; (\psi \rightarrow K_i[!\psi]\varphi)$$

# Introduction

Knowing *that* $\heartsuit$ announcing *that*:

$$[!\psi]K_i(\varphi) \;\leftrightarrow\; (\psi \rightarrow K_i[!\psi]\varphi)$$

What about knowing *what* and announcing *what*?

# Example

| Student | Subject | Assessment |
|---|---|---|
| 1 | Mathematics | good |
| 2 | Mathematics | very good |
| 3 | Logic | good |
| 4 | Computer Science | bad |

# Example

| Student | Subject | Assessment |
|---|---|---|
| ~~1~~ | ~~Mathematics~~ | ~~good~~ |
| ~~2~~ | ~~Mathematics~~ | ~~very good~~ |
| 3 | Logic | good |
| ~~4~~ | ~~Computer Science~~ | ~~bad~~ |

$$\mathcal{M}, 3 \models [\mathit{Subject}]\mathit{Kv}(\mathit{Assessment})$$

# Knowing *what*

$$\mathsf{Kv}_i(c)$$

# Knowing *what*

$$\mathsf{Kv}_i(c)$$

Semantics:

$$\mathcal{M}, s \vDash \mathsf{Kv}_i(c) \;\Leftrightarrow\; \forall t_1, t_2 : s \sim_i t_1 \wedge s \sim_i t_2 \Rightarrow V_c(t_1) = V_c(t_2)$$

# Knowing *what*

$$\mathsf{Kv}_i(c)$$

Semantics:

$$\mathcal{M}, s \vDash \mathsf{Kv}_i(c) \;\Leftrightarrow\; \forall t_1, t_2 : s \sim_i t_1 \wedge s \sim_i t_2 \Rightarrow V_c(t_1) = V_c(t_2)$$

Considered already by (Plaza 2007) in combination with PAL, recently by (Wang and Fan 2013,Wang and Fan (2014)).
Latest news:

- we can make it normal (Gu and Wang 2016)
- generalize all the way (Baltag 2016)

# *Announcing* what

How about *announcing what*?
Other words: revealing, telling the value, *inspecting*, ...

# *Announcing* what

How about *announcing what*?
Other words: revealing, telling the value, *inspecting*, ...
PIL := Public Inspection Logic

# Contents

# Single Agent PIL

Let $c$ range over some set of variables $\mathbb{C}$.

1. Language: $\varphi ::= \top \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathsf{Kv}(c) \mid [c]\varphi$

# Single Agent PIL

Let $c$ range over some set of variables $\mathbb{C}$.

1. Language: $\varphi ::= \top \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathsf{Kv}(c) \mid [c]\varphi$
2. Models: $\mathcal{M} = \langle S, V \rangle$ where
   - $V : (\mathbb{C} \times S) \to D$ for some $D$.

   Write $s =_c t$ iff $V(c, s) = V(c, t)$.

# Single Agent PIL

Let $c$ range over some set of variables $\mathbb{C}$.

1. Language: $\varphi ::= \top \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathsf{Kv}(c) \mid [c]\varphi$

2. Models: $\mathcal{M} = \langle S, V \rangle$ where
   - $V : (\mathbb{C} \times S) \to D$ for some $D$.

   Write $s =_c t$ iff $V(c, s) = V(c, t)$.

3. Interpretation:

$$\begin{array}{lcl} \mathcal{M}, s \vDash \mathsf{Kv}(c) & \Leftrightarrow & \text{for all } t \in S : s =_c t \\ \mathcal{M}, s \vDash [c]\varphi & \Leftrightarrow & \mathcal{M}|_c^s, s \vDash \varphi \end{array}$$

   where $\mathcal{M}|_c^s$ is $\langle S', V|_{C \times S'} \rangle$ with $S' = \{t \in S \mid s =_c t\}$.

# Example Model

$\mathcal{M} = \langle S, V \rangle$

- $S = \{s, t, u\}$
- $V$ as follows, e.g. $V(s, d) = 1$.

| | c | d | e |
|---|---|---|---|
| s | 1 | 1 | 3 |
| t | 2 | 1 | 2 |
| u | 3 | 3 | 1 |

$\mathcal{M}, s \models \mathsf{Kv}(t)$
$\mathcal{M} \models \mathsf{Kv}(t)$
$\mathcal{M}, s \models [e]\mathsf{Kv}(d)$
$\mathcal{M}, s \models \neg[d]\mathsf{Kv}(e)$

# Proof system for PIL

| | |
|---|---|
| Tautologies | propositional tautologies |
| Distribution | $[c](\varphi \to \psi) \to ([c]\varphi \to [c]\psi)$ |
| Learning | $[c]\mathsf{Kv}(c)$ |
| No Forgetting | $\mathsf{Kv}(c) \to [d]\mathsf{Kv}(c)$ |
| Determinacy | $\langle c\rangle\varphi \leftrightarrow [c]\varphi$ |
| Commutativity | $[c][d]\varphi \leftrightarrow [d][c]\varphi$ |
| Irrelevance | $\mathsf{Kv}(c) \to ([c]\varphi \to \varphi)$ |

Modus Ponens: $\dfrac{\varphi, \varphi \to \psi}{\psi}$

Necessitation: $\dfrac{\varphi}{[c]\varphi}$

# Dependency

For any two finite sets $C, D \subseteq \mathbb{C}$, let

$$\mathrm{Kv}(C, D) := [c_1] \ldots [c_n](\mathrm{Kv}d_1 \wedge \ldots \wedge \mathrm{Kv}d_m)$$

We get:

$$\mathcal{M}, s \vDash \mathrm{Kv}(C, D) \Leftrightarrow \text{for all } t \in S : \text{if } s =_C t \text{ then } s =_D t$$

# Dependency

For any two finite sets $C, D \subseteq \mathbb{C}$, let

$$\mathrm{Kv}(C, D) := [c_1] \ldots [c_n](\mathrm{Kv}d_1 \wedge \ldots \wedge \mathrm{Kv}d_m)$$

We get:

$$\mathcal{M}, s \vDash \mathrm{Kv}(C, D) \Leftrightarrow \text{for all } t \in S : \text{if } s =_C t \text{ then } s =_D t$$

**Lemma**
$\varphi ::= \top \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathrm{Kv}(C, C)$ has the same expressive power.
*Proof.* Push sensing operator through negations and conjunctions:

$$\begin{array}{rl} & [c](\neg \mathrm{Kv}d \wedge [e]\mathrm{Kv}f) \\ \equiv & [c]\neg \mathrm{Kv}d \wedge [c][e]\mathrm{Kv}f \\ \equiv & \neg \mathrm{Kv}(c, d) \wedge \mathrm{Kv}(\{c, e\}, f) \end{array}$$

# Hints from Database Theory: Armstrong's Axioms

**Lemma**

The (Armstrong 1974) axioms are valid and provable:

| | |
|---|---|
| Projectivity | $\mathrm{Kv}(C, D)$ for any $D \subseteq C$ |
| Transitivity | $\mathrm{Kv}(C, D) \wedge \mathrm{Kv}(D, E) \rightarrow \mathrm{Kv}(C, E)$ |
| Additivity | $\mathrm{Kv}(C, D) \wedge \mathrm{Kv}(C, E) \rightarrow \mathrm{Kv}(C, D \cup E)$ |

# Hints from Database Theory: Armstrong's Axioms

**Lemma**

The (Armstrong 1974) axioms are valid and provable:

| | |
|---|---|
| Projectivity | $\mathsf{Kv}(C, D)$ for any $D \subseteq C$ |
| Transitivity | $\mathsf{Kv}(C, D) \wedge \mathsf{Kv}(D, E) \rightarrow \mathsf{Kv}(C, E)$ |
| Additivity | $\mathsf{Kv}(C, D) \wedge \mathsf{Kv}(C, E) \rightarrow \mathsf{Kv}(C, D \cup E)$ |

[CITAAT] Dependency Structures of Data Base Relationships.
WW Armstrong - IFIP congress, 1974 - Geneva, Switzerland
Geciteerd door 1132 Verwante artikelen Citeren Opslaan

# Completeness

**Theorem** (Strong Completeness)
For all $\Delta \subseteq \mathcal{L}_1$ and all $\varphi \in \mathcal{L}_1$: If $\Delta \models \varphi$, then also $\Delta \vdash \varphi$.

# Completeness

**Theorem** (Strong Completeness)
For all $\Delta \subseteq \mathcal{L}_1$ and all $\varphi \in \mathcal{L}_1$: If $\Delta \models \varphi$, then also $\Delta \vdash \varphi$.

*Proof Strategy*:

- Suppose $\Delta \nvdash \varphi$.
- Then $\Delta \cup \{\neg\varphi\}$ is consistent
- Take a maximally consistent set $\Gamma \supseteq \Delta \cup \{\neg\varphi\}$.
- Build a model $\mathcal{M}_\Gamma$ such that for some world $\mathbb{C}$ we have $\mathcal{M}_\Gamma, \mathbb{C} \models \Gamma$ which implies $\Delta \not\models \varphi$.

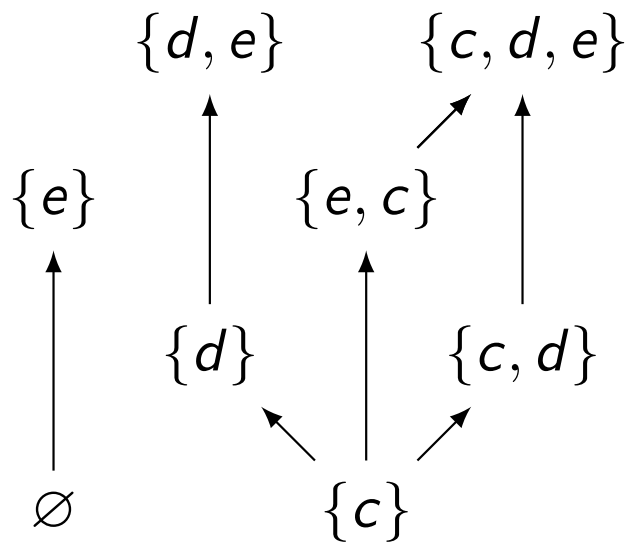
# Completeness via the Canonical Dependency Graph

**Canonical Graph**

Let $G_\Gamma := (\mathcal{P}(\mathbb{C}), \to)$ where $A \to B$ iff $\mathsf{Kv}(A, B) \in \Gamma$.

**Example**

$\Gamma = \{\mathsf{Kv}(c, d), \mathsf{Kv}(e), \dots\}$.



**Lemma**

$G_\Gamma$ is projective, transitive and additive.

Call $s \subseteq \mathbb{C}$ *closed* under $G_\Gamma$ iff whenever $A \subseteq s$ and $A \to B$ according to $G_\Gamma$, then also $B \subseteq s$.

**Lemma**
$G_\Gamma$ is projective, transitive and additive.

Call $s \subseteq \mathbb{C}$ *closed* under $G_\Gamma$ iff whenever $A \subseteq s$ and $A \to B$ according to $G_\Gamma$, then also $B \subseteq s$.

**Canonical model**:
$\mathcal{M}_\Gamma = (S, V)$ where

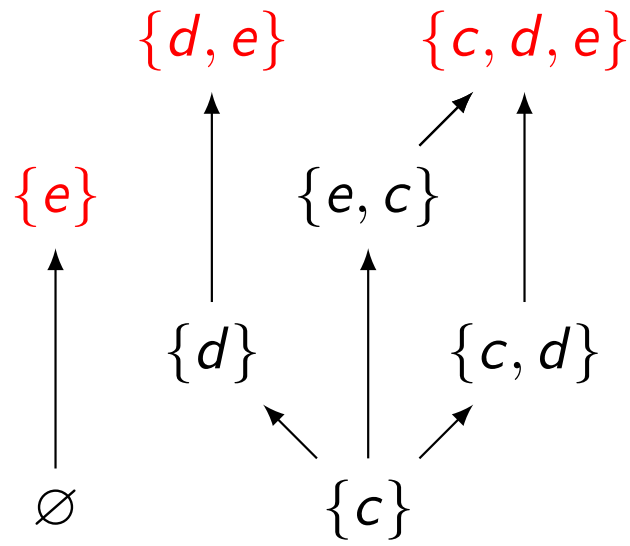
$$S := \{s \subseteq \mathbb{C} \mid s \text{ is closed under } G_\Gamma\}$$

$$\text{and } V_s(c) = \begin{cases} 0 & \text{if } c \in s \\ 1 & \text{otherwise} \end{cases}$$

**Truth Lemma**

$$\mathcal{M}_\Gamma, \mathbb{C} \vDash \varphi \text{ iff } \varphi \in \Gamma$$

# Example continued




| V | c | d | e |
|---|---|---|---|
| $s = \{e\}$ | 1 | 1 | 0 |
| $t = \{d, e\}$ | 1 | 0 | 0 |
| $u = \mathbb{C} = \{c, d, e\}$ | 0 | 0 | 0 |

$$\mathcal{M}, u \vDash \mathsf{Kv}(c, d) \wedge \mathsf{Kv}(e) \wedge \ldots$$

# Multi-Agent PIL

Language:

$$\varphi ::= \top \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathsf{Kv}_i c \mid [c]\varphi$$

Add to the models:

$$\sim_i \subseteq S \times S$$

Interpretation:

$$\mathcal{M}, s \vDash \mathsf{Kv}_i c \iff \forall t \in S : s \sim_i t \Rightarrow s =_c t$$

Proof system with restricted Irrelevance:

$$\mathsf{Kv}_i c \to ([c]\varphi \to \varphi)$$

where $\varphi$ does not mention any agent besides $i$

Strong completeness proof: $S = \mathcal{P}(\mathbb{C})$, use a canonical graph for each agent to define $\sim_i$.

# Comparison with Dependence Logic

Our following validity does not translate to the Armstrong framework:

$$[c](\mathrm{Kv}d \vee \mathrm{Kv}e) \to ([c]\mathrm{Kv}d \vee [c]\mathrm{Kv}e)$$

# Comparison with Dependence Logic

Our following validity does not translate to the Armstrong framework:

$$[c](\mathrm{Kv}d \vee \mathrm{Kv}e) \to ([c]\mathrm{Kv}d \vee [c]\mathrm{Kv}e)$$

*Pointed* models convey more information than a team:

| $c$ | $d$ | $e$ |
|---|---|---|
| 1 | 1 | 3 |
| 1 | 1 | 2 |
| 2 | 2 | 1 |
| 2 | 3 | 1 |

- "After inspecting $c$ we know $d$ or we know $e$." $[c](\mathrm{Kv}d \vee \mathrm{Kv}e)$
- "After inspecting $c$ we know $d$ or after inspecting $c$ we know $e$." $([c]\mathrm{Kv}d \vee [c]\mathrm{Kv}e)$

# Comparison with Dependence Logic

Our following validity does not translate to the Armstrong framework:

$$[c](\mathrm{Kv}d \vee \mathrm{Kv}e) \to ([c]\mathrm{Kv}d \vee [c]\mathrm{Kv}e)$$

*Pointed* models convey more information than a team:

| $c$ | $d$ | $e$ |
|---|---|---|
| 1 | 1 | 3 |
| 1 | 1 | 2 |
| 2 | 2 | 1 |
| 2 | 3 | 1 |

- "After inspecting $c$ we know $d$ or we know $e$." $[c](\mathrm{Kv}d \vee \mathrm{Kv}e)$
- "After inspecting $c$ we know $d$ or after inspecting $c$ we know $e$." $([c]\mathrm{Kv}d \vee [c]\mathrm{Kv}e)$
- False: "$c$ globally determines $d$ or $c$ globally determines $e$"

Armstrong's system can not express $[c](\mathrm{Kv}d \vee \mathrm{Kv}e)$.

# Related and Future Work

| | | | | | | |
|---|---|---|---|---|---|---|
| $p$ | $K\varphi$ | | | | $[!\varphi]\varphi$ | (Plaza 2007) |
| $p$ | $K\varphi$ | $\mathrm{Kv}(c)$ | | | $[!\varphi]\varphi$ | (Plaza 2007) |
| $p$ | $K\varphi$ | $\mathrm{Kv}(c)$ | $\mathrm{Kv}(\varphi, c)$ | | $[!\varphi]\varphi$ | (Wang and Fan 2013)<br>(Wang and Fan 2014)<br>(Gu and Wang 2016) |
| | | $\mathrm{Kv}(c)$ | | $[c]\varphi$ | | this |
| $p$ | $K\varphi$ | $\mathrm{Kv}(c)$ | $\mathrm{Kv}(\varphi, c)$ | $[c]\varphi$ | $[!\varphi]\varphi$ | (Baltag 2016) |

- What does it mean to know a *function*?
- Application to Security Protocols?

# Summary

Knowing what ♡ Announcing what

# Bonus Slide: Provable Stuff

- $\langle c \rangle \top$ (seriality)
- $\mathsf{Kv}(c) \to (\varphi \to [c]\varphi)$ (Irrelevance*)
- $[c](\varphi \wedge \psi) \leftrightarrow [c]\varphi \wedge [c]\psi$ (Distribution*)
- $[c_1] \ldots [c_n](\varphi \to \psi) \to ([c_1] \ldots [c_n]\varphi \to [c_1] \ldots [c_n]\psi)$ (multi-Distribution)
- $[c_1] \ldots [c_n](\varphi \wedge \psi) \leftrightarrow [c_1] \ldots [c_n]\varphi \wedge [c_1] \ldots [c_n]\psi$ (multi-Distribution*)
- $[c_1] \ldots [c_n](\mathsf{Kv}(c_1) \wedge \ldots \mathsf{Kv}(c_n))$ (multi-Learning)
- $(\mathsf{Kv}(c_1) \wedge \cdots \wedge \mathsf{Kv}(c_n)) \to [d_1] \ldots [d_n](\mathsf{Kv}(c_1) \wedge \cdots \wedge \mathsf{Kv}(c_n))$ (multi-No Forgetting)
- $(\mathsf{Kv}(c_1) \wedge \cdots \wedge \mathsf{Kv}(c_n)) \to ([c_1] \ldots [c_n]\varphi \to \varphi)$ (multi-Irrelevance)

# References


Armstrong, William Ward. 1974. "Dependency Structures of Data Base Relationships." In *IFIP Congress*, 74:580–83. Geneva, Switzerland.

Baltag, Alexandru. 2016. "To Know Is to Know the Value of a Variable." *Advances in Modal Logic* 11: 135–55.

Gu, Tau, and Yanjing Wang. 2016. "'Knowing Value' Logic as a Normal Modal Logic." *Advances in Modal Logic* 11: 362–81. `http://arxiv.org/abs/1604.08709`.

Plaza, Jan. 2007. "Logics of Public Communications." *Synthese* 158 (2): 165–79. doi:10.1007/s11229-007-9168-7.

Wang, Yanjing, and Jie Fan. 2013. "Knowing That, Knowing What, and Public Communication: Public Announcement Logic with Kv Operators." In *IJCAI '13*, 1147–54.

———. 2014. "Conditionally Knowing What." *Advances in Modal Logic* 10: 569–87. `http://www.aiml.net/volumes/volume10/Wang-Fan.pdf`.