

# Knowing Values and Public Inspection

Malvin Gattinger  
with Jan van Eijck & Yanjing Wang

[arxiv.org/abs/1609.03338](https://arxiv.org/abs/1609.03338)  
slides at [w4eg.de/malvin](http://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* ♥ 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

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

Knowing *what*

$Kv_i(c)$

## Knowing *what*

$Kv_i(c)$

Semantics:

$$\mathcal{M}, s \models Kv_i(c) \Leftrightarrow \forall t_1, t_2 : s \sim_i t_1 \wedge s \sim_i t_2 \Rightarrow V_c(t_1) = V_c(t_2)$$

## Knowing *what*

$Kv_i(c)$

Semantics:

$$\mathcal{M}, s \models Kv_i(c) \Leftrightarrow \forall t_1, t_2 : s \sim_i t_1 \wedge s \sim_i t_2 \Rightarrow V_c(t_1) = V_c(t_2)$$

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

## 1. Single Agent PIL

- ▶ Syntax
- ▶ Semantics
- ▶ Example
- ▶ Proof System
- ▶ Armstrong Axioms
- ▶ Completeness via Canonical Graph

## 2. Multi-Agent PIL

## 3. Comparison with Dependence Logic

## 4. Related and Future Work

# Single Agent PIL

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

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

# Single Agent PIL

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

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

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

3. Interpretation:

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

where  $\mathcal{M}|_c^s$  is  $\langle S', V|_{\mathbb{C} \times S'} \rangle$  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

$$\mathcal{M}, s \models K_v(t)$$

$$\mathcal{M} \models K_v(t)$$

$$\mathcal{M}, s \models [e]K_v(d)$$

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

# Proof system for PIL

---

Tautologies	propositional tautologies
Distribution	$[c](\varphi \rightarrow \psi) \rightarrow ([c]\varphi \rightarrow [c]\psi)$
Learning	$[c]Kv(c)$
No Forgetting	$Kv(c) \rightarrow [d]Kv(c)$
Determinacy	$\langle c \rangle \varphi \leftrightarrow [c]\varphi$
Commutativity	$[c][d]\varphi \leftrightarrow [d][c]\varphi$
Irrelevance	$Kv(c) \rightarrow ([c]\varphi \rightarrow \varphi)$

---

$$\text{Modus Ponens: } \frac{\varphi, \varphi \rightarrow \psi}{\psi}$$

$$\text{Necessitation: } \frac{\varphi}{[c]\varphi}$$



## Dependency

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

$$\text{Kv}(C, D) := [c_1] \dots [c_n](\text{Kvd}_1 \wedge \dots \wedge \text{Kvd}_m)$$

We get:

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

## Dependency

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

$$\text{Kv}(C, D) := [c_1] \dots [c_n](\text{Kvd}_1 \wedge \dots \wedge \text{Kvd}_m)$$

We get:

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

### Lemma

$\varphi ::= \top \mid \neg\varphi \mid \varphi \wedge \varphi \mid \text{Kv}(C, C)$  has the same expressive power.

*Proof.* Push sensing operator through negations and conjunctions:

$$\begin{aligned} & [c](\neg\text{Kvd} \wedge [e]\text{Kvf}) \\ \equiv & [c]\neg\text{Kvd} \wedge [c][e]\text{Kvf} \\ \equiv & \neg\text{Kv}(c, d) \wedge \text{Kv}(\{c, e\}, f) \end{aligned}$$

# Hints from Database Theory: Armstrong's Axioms

## Lemma

The (Armstrong 1974) axioms are valid and provable:

---

Projectivity  $K_v(C, D)$  for any  $D \subseteq C$

Transitivity  $K_v(C, D) \wedge K_v(D, E) \rightarrow K_v(C, E)$

Additivity  $K_v(C, D) \wedge K_v(C, E) \rightarrow K_v(C, D \cup E)$

---

# Hints from Database Theory: Armstrong's Axioms

## Lemma

The (Armstrong 1974) axioms are valid and provable:

---

Projectivity	$Kv(C, D)$ for any $D \subseteq C$
Transitivity	$Kv(C, D) \wedge Kv(D, E) \rightarrow Kv(C, E)$
Additivity	$Kv(C, D) \wedge Kv(C, E) \rightarrow Kv(C, D \cup E)$

---

[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  $\Delta \not\vdash \varphi$ .
- ▶ 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  $\Delta \not\models \varphi$ .

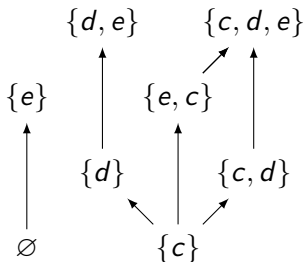
# Completeness via the Canonical Dependency Graph

## Canonical Graph

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

## Example

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



## Lemma

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

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

## Canonical model:

$\mathcal{M}_\Gamma = (S, V)$  where

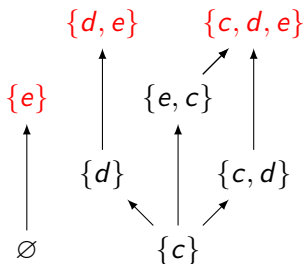
$$S := \{s \subseteq \mathbb{C} \mid s \text{ is closed under } G_\Gamma\}$$

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

## Truth Lemma

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

## 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

---

$\mathcal{M}, u \models \text{Kv}(c, d) \wedge \text{Kv}(e) \wedge \dots$

# Multi-Agent PIL

Language:

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

Add to the models:

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

Interpretation:

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

Proof system with restricted Irrelevance:

$$\mathbf{Kv}_i c \rightarrow ([c]\varphi \rightarrow \varphi)$$

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:

$$[c](Kvd \vee Kve) \rightarrow ([c]Kvd \vee [c]Kve)$$

## Comparison with Dependence Logic

Our following validity does not translate to the Armstrong framework:

$$[c](Kvd \vee Kve) \rightarrow ([c]Kvd \vee [c]Kve)$$

*Pointed* models convey more information than a team:

<i>c</i>	<i>d</i>	<i>e</i>
1	1	3
1	1	2
2	2	1
2	3	1

- ▶ “After inspecting *c* we know *d* or we know *e*.”  $[c](Kvd \vee Kve)$
- ▶ “After inspecting *c* we know *d* or after inspecting *c* we know *e*.”  $([c]Kvd \vee [c]Kve)$

## Comparison with Dependence Logic

Our following validity does not translate to the Armstrong framework:

$$[c](Kvd \vee Kve) \rightarrow ([c]Kvd \vee [c]Kve)$$

*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  $e$ .”  $[c](Kvd \vee Kve)$
- ▶ “After inspecting  $c$  we know  $d$  or after inspecting  $c$  we know  $e$ .”  $([c]Kvd \vee [c]Kve)$
- ▶ False: “ $c$  globally determines  $d$  or  $c$  globally determines  $e$ ”

Armstrong’s system can not express  $[c](Kvd \vee Kve)$ .

## Related and Future Work

---

$p$	$K\varphi$			$[!\varphi]\varphi$	(Plaza 2007)
$p$	$K\varphi$	$Kv(c)$		$[!\varphi]\varphi$	(Plaza 2007)
$p$	$K\varphi$	$Kv(c)$	$Kv(\varphi, c)$	$[!\varphi]\varphi$	(Wang and Fan 2013) (Wang and Fan 2014) (Gu and Wang 2016)
		$Kv(c)$		$[c]\varphi$	this
$p$	$K\varphi$	$Kv(c)$	$Kv(\varphi, c)$	$[c]\varphi$	$[!\varphi]\varphi$ (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)
- ▶  $Kv(c) \rightarrow (\varphi \rightarrow [c]\varphi)$  (Irrelevance\*)
- ▶  $[c](\varphi \wedge \psi) \leftrightarrow [c]\varphi \wedge [c]\psi$  (Distribution\*)
- ▶  $[c_1] \dots [c_n](\varphi \rightarrow \psi) \rightarrow ([c_1] \dots [c_n]\varphi \rightarrow [c_1] \dots [c_n]\psi)$   
(multi-Distribution)
- ▶  $[c_1] \dots [c_n](\varphi \wedge \psi) \leftrightarrow [c_1] \dots [c_n]\varphi \wedge [c_1] \dots [c_n]\psi$   
(multi-Distribution\*)
- ▶  $[c_1] \dots [c_n](Kv(c_1) \wedge \dots \wedge Kv(c_n))$  (multi-Learning)
- ▶  $(Kv(c_1) \wedge \dots \wedge Kv(c_n)) \rightarrow [d_1] \dots [d_n](Kv(c_1) \wedge \dots \wedge Kv(c_n))$   
(multi-No Forgetting)
- ▶  $(Kv(c_1) \wedge \dots \wedge Kv(c_n)) \rightarrow ([c_1] \dots [c_n]\varphi \rightarrow \varphi)$   
(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>.