Model Checking for Coalition Announcement Logic

Rustam GALIMULLIN [1]
Natasha ALECHINA [1]
Hans VAN DITMARSCH [2]

[1]University of Nottingham, Nottingham, UK

[2]CNRS, LORIA, University of Lorraine, France & ReLaX, Chennai, India

What this talk is about

- What agents know and don't know (through the lens of epistemic logic[1])
- The effect of public announcements[2] on agent's knowledge
- How agents can achieve certain goals by teaming up in coalitions and making joint announcements[3]
- Model checking for such a framework

[1] Hans van Ditmarsch et al., eds. *Handbook of Epistemic Logic*. College Publications, 2015.

[2] Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. *Dynamic Epistemic Logic*. Vol. 337. Synthese Library. Springer, 2008.

[3] Thomas Ågotnes and Hans van Ditmarsch. "Coalitions and Announcements". In: *Proceedings of AAMAS 2008*. Ed. by Lin Padgham et al. IFAAMAS, 2008, pp. 673–680.

What this talk is about

- What agents know and don't know (through the lens of epistemic logic[1])
- The effect of public announcements[2] on agent's knowledge
- How agents can achieve certain goals by teaming up in coalitions and making joint announcements[3]
- Model checking for such a framework

---

[1] Hans van Ditmarsch et al., eds. *Handbook of Epistemic Logic*. College Publications, 2015.

[2] Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. *Dynamic Epistemic Logic*. Vol. 337. Synthese Library. Springer, 2008.

[3] Thomas Ågotnes and Hans van Ditmarsch. "Coalitions and Announcements". In: *Proceedings of AAMAS 2008*. Ed. by Lin Padgham et al. IFAAMAS, 2008, pp. 673–680.

Example

There are two households, a and b, and an electricity substation c that requires information about how many households consume power. Moreover, it is imperative that individual household's consumption remains unknown.

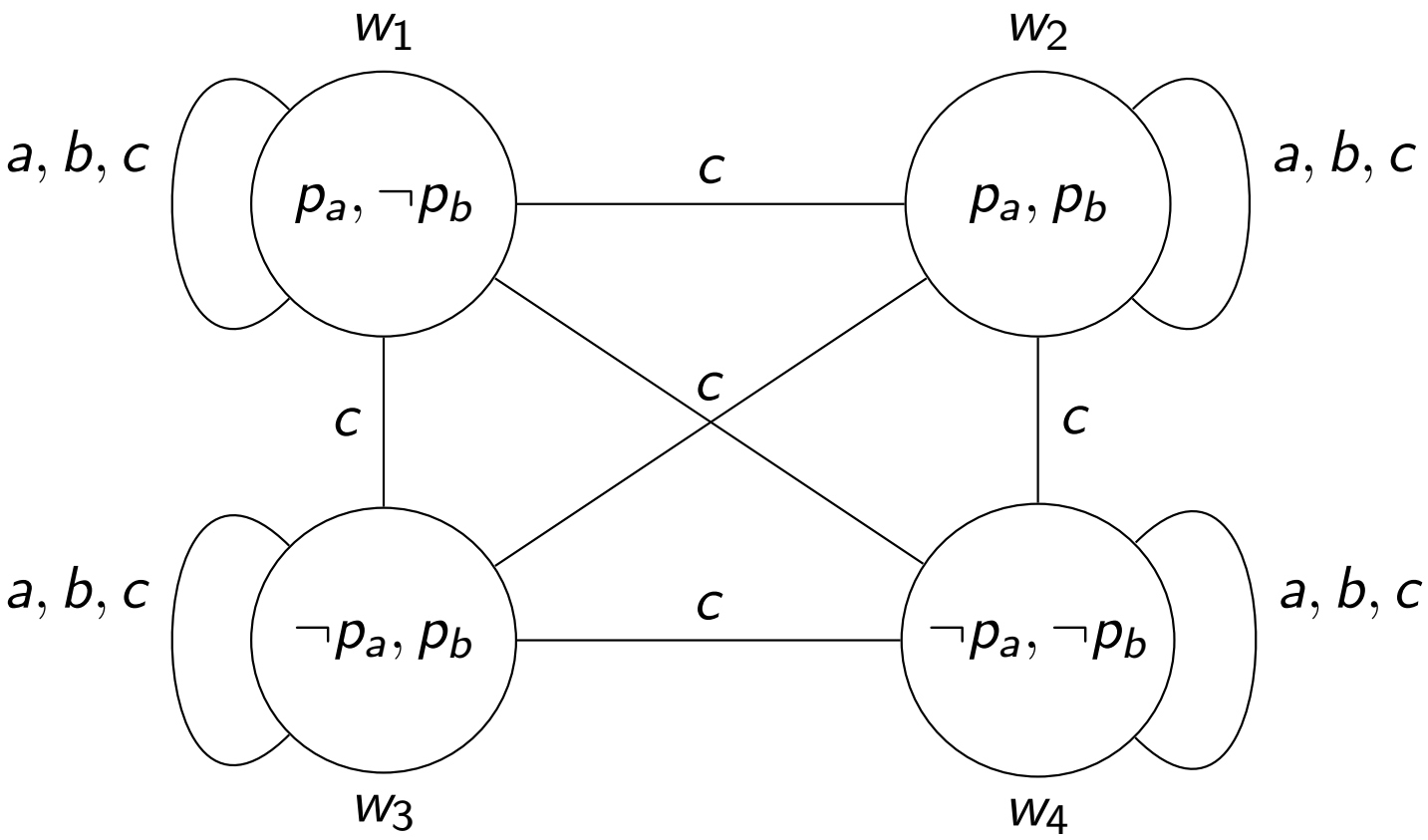
Example

There are two households, a and b, and an electricity substation c that requires information about how many households consume power. Moreover, it is imperative that individual household's consumption remains unknown.

Example

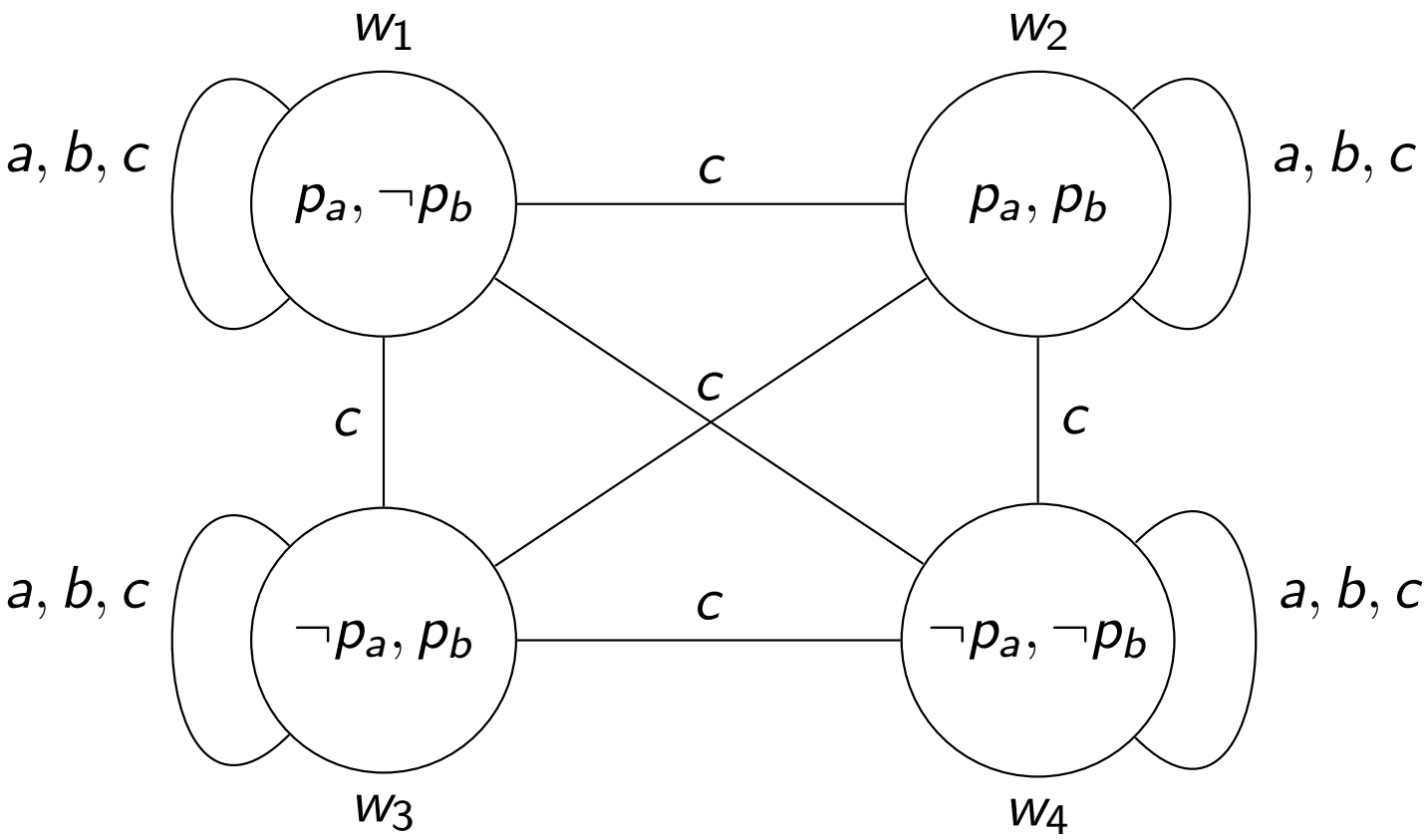
$(M, w_1) \models p_a \wedge \neg p_b$, $(M, w_1) \models K_a p_a$, $(M, w_1) \models K_a p_b$, $(M, w_1) \models \neg K_c p_a$.

Example



$(M, w_1) \models p_a \wedge \neg p_b$, $(M, w_1) \models K_a p_a$, $(M, w_1) \models K_a p_b$, $(M, w_1) \models \neg K_c p_a$.

Example



$$(M, w_1) \models p_a \wedge \neg p_b,\ (M, w_1) \models K_a p_a,\ (M, w_1) \models K_a p_b,$$
$$(M, w_1) \models \neg K_c p_a.$$

Public Announcements

In the continuation of the example, suppose that a announces that

Exactly one of us, a and b, uses electricity, i.e.
$(p_a \wedge \neg p_b) \vee (\neg p_a \wedge p_b)$.

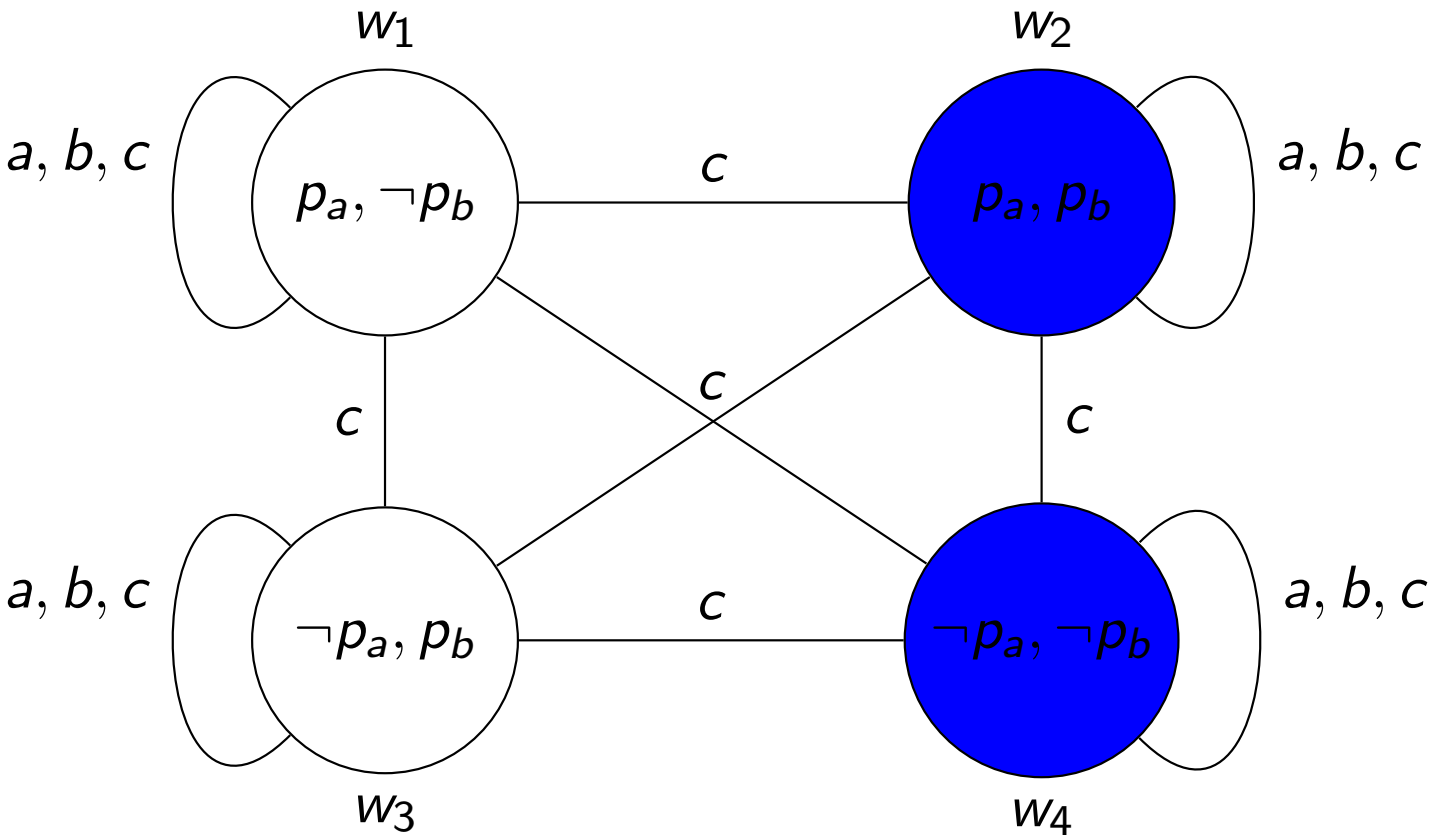
Public Announcements

In the continuation of the example, suppose that a announces that

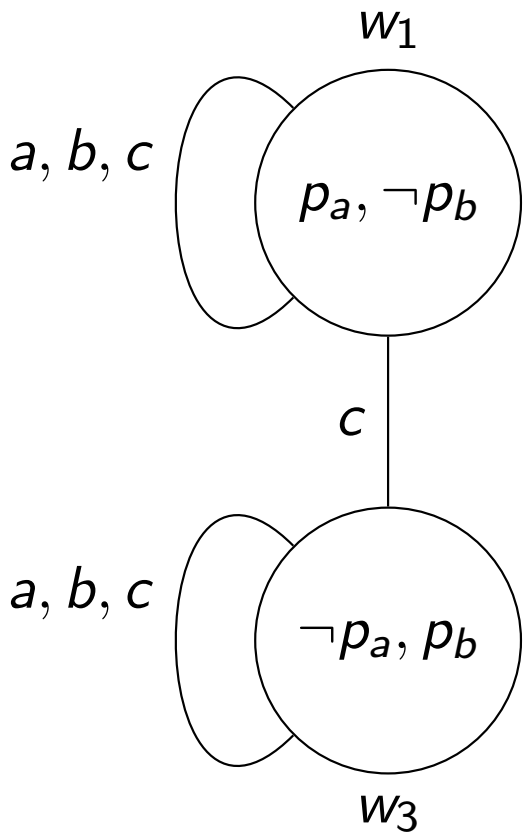
> Exactly one of us, a and b, uses electricity, i.e.
> $(p_a \wedge \neg p_b) \vee (\neg p_a \wedge p_b)$.

Example Updated

Example Updated

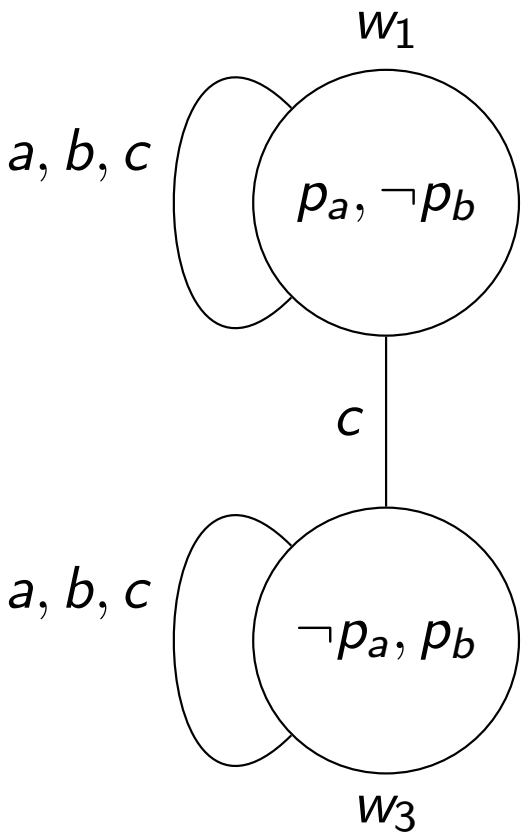


Example Updated



$(M, w_1)^{ann} \models p_a \wedge \neg p_b$, $(M, w_1)^{ann} \models K_a p_a$, $(M, w_1)^{ann} \models K_a p_b$, $(M, w_1)^{ann} \models K_c((p_a \wedge \neg p_b) \vee (\neg p_a \wedge p_b))$.

Example Updated



$$(M, w_1)^{ann} \models p_a \wedge \neg p_b,\ (M, w_1)^{ann} \models K_a p_a,\ (M, w_1)^{ann} \models K_a p_b,$$
$$(M, w_1)^{ann} \models K_c((p_a \wedge \neg p_b) \vee (\neg p_a \wedge p_b)).$$

Models

Definition (Epistemic Model)

An epistemic model is a triple $M = (W, \sim, V)$, where

- W is a non-empty set of states,
- $\sim\colon A \to \mathcal{P}(W \times W)$ assigns an equivalence relation to each agent,
- $V : P \to \mathcal{P}(W)$ is the valuation function.

A pair (M, w) with $w \in W$ is called a pointed model.
An announcement in a pointed model (M, w) results in an updated pointed model $(M, w)^\varphi$ with $W^\varphi = [\![\varphi]\!]_M$, $\sim_a^\varphi = \sim_a \cap ([\![\varphi]\!]_M \times [\![\varphi]\!]_M)$, and $V^\varphi(p) = V(p) \cap [\![\varphi]\!]_M$.

Models

Definition (Epistemic Model)

An epistemic model is a triple $M = (W, \sim, V)$, where

- W is a non-empty set of states,
- $\sim$: $A \to \mathcal{P}(W \times W)$ assigns an equivalence relation to each agent,
- $V : P \to \mathcal{P}(W)$ is the valuation function.

A pair (M, w) with $w \in W$ is called a pointed model.
An announcement in a pointed model (M, w) results in an updated pointed model $(M, w)^\varphi$ with $W^\varphi = [\![\varphi]\!]_M$, $\sim_a^\varphi = \sim_a \cap ([\![\varphi]\!]_M \times [\![\varphi]\!]_M)$, and $V^\varphi(p) = V(p) \cap [\![\varphi]\!]_M$.

Semantics

Definition (Semantics)

$$
\begin{array}{lll}
(M,w) \models p & \text{iff} & w \in V(p) \\
(M,w) \models \neg\varphi & \text{iff} & (M,w) \not\models \varphi \\
(M,w) \models \varphi \wedge \psi & \text{iff} & (M,w) \models \varphi \text{ and } (M,w) \models \psi \\
(M,w) \models K_a\varphi & \text{iff} & \forall v \in W : w \sim_a v \text{ implies } (M,v) \models \varphi \\
(M,w) \models [\varphi]\psi & \text{iff} & (M,w) \models \varphi \text{ implies } (M,w)^{\varphi} \models \psi
\end{array}
$$

Formula $[\varphi]\psi$ is read as

after a public announcement of φ, ψ holds in the resulting model.

Dual of []

$$(M,w) \models \langle\varphi\rangle\psi \quad \text{iff} \quad (M,w) \models \varphi \text{ and } (M,w)^{\varphi} \models \psi$$

Semantics

Definition (Semantics)

$$
\begin{array}{lll}
(M,w) \models p & \text{iff} & w \in V(p) \\
(M,w) \models \neg\varphi & \text{iff} & (M,w) \not\models \varphi \\
(M,w) \models \varphi \wedge \psi & \text{iff} & (M,w) \models \varphi \text{ and } (M,w) \models \psi \\
(M,w) \models K_a\varphi & \text{iff} & \forall v \in W : w \sim_a v \text{ implies } (M,v) \models \varphi \\
(M,w) \models [\varphi]\psi & \text{iff} & (M,w) \models \varphi \text{ implies } (M,w)^{\varphi} \models \psi
\end{array}
$$

Formula $[\varphi]\psi$ is read as

after a public announcement of φ, ψ holds in the resulting model.

Dual of []

$$(M,w) \models \langle\varphi\rangle\psi \quad \text{iff} \quad (M,w) \models \varphi \text{ and } (M,w)^{\varphi} \models \psi$$

Announcements by Coalitions

We are interested in the following restrictions on announcements:

- Announcements are made by agents
- Agents can only announce what they know
- Coalitions of agents can announce conjunctions of formulas, where each conjunct is a formula known by an agent in the coalition
- Agents outside of the coalitions also make an announcement that can preclude coalition to reach its goal

Announcements by Coalitions

We are interested in the following restrictions on announcements:

- Announcements are made by agents
- Agents can only announce what they know
- Coalitions of agents can announce conjunctions of formulas, where each conjunct is a formula known by an agent in the coalition
- Agents outside of the coalitions also make an announcement that can preclude coalition to reach its goal

Announcements by Coalitions

Coalition Announcement Logic (CAL) allows us to reason about announcements by coalition of agents. This is Public Announcement Logic (PAL) with the following added operators:

$\langle\!\langle G \rangle\!\rangle \varphi$: 'there is an announcement by agents from G such that whatever agents $A \setminus G$ outside of the coalition announce, φ holds,'

or

$[\![G]\!] \varphi$: 'whatever agents from G announce, there is an announcement by the agents from the outside of the coalition, such that φ holds.'

Announcements by Coalitions

Coalition Announcement Logic (CAL) allows us to reason about announcements by coalition of agents. This is Public Announcement Logic (PAL) with the following added operators:

$\langle\!\langle G \rangle\!\rangle \varphi$: 'there is an announcement by agents from G such that whatever agents $A \setminus G$ outside of the coalition announce, φ holds,'

or

$[\![G]\!] \varphi$: 'whatever agents from G announce, there is an announcement by the agents from the outside of the coalition, such that φ holds.'

Semantics of CAL

Let ψ_G be a shorthand for a formula of the type $K_a\varphi_a \wedge \ldots \wedge K_b\varphi_b$, where $a, \ldots, b \in G$, and $\varphi_a, \ldots, \varphi_b$ are formulas of epistemic logic. For example, ψ_G may be $K_a p \wedge K_b(p \rightarrow q)$ for $G = \{a, b\}$.

Definition (Semantics)

$$(M, w) \models [\![G]\!]\varphi \quad \text{iff} \quad \forall \psi_G \exists \chi_{A \setminus G} : (M, w) \models \psi_G \rightarrow \langle \psi_G \wedge \chi_{A \setminus G} \rangle \varphi$$
$$(M, w) \models \langle\!\langle G \rangle\!\rangle \varphi \quad \text{iff} \quad \exists \psi_G \forall \chi_{A \setminus G} : (M, w) \models \psi_G \wedge [\psi_G \wedge \chi_{A \setminus G}] \varphi$$

Semantics of CAL

Let ψ_G be a shorthand for a formula of the type $K_a\varphi_a \wedge \ldots \wedge K_b\varphi_b$, where $a, \ldots, b \in G$, and $\varphi_a, \ldots, \varphi_b$ are formulas of epistemic logic. For example, ψ_G may be $K_a p \wedge K_b(p \to q)$ for $G = \{a, b\}$.

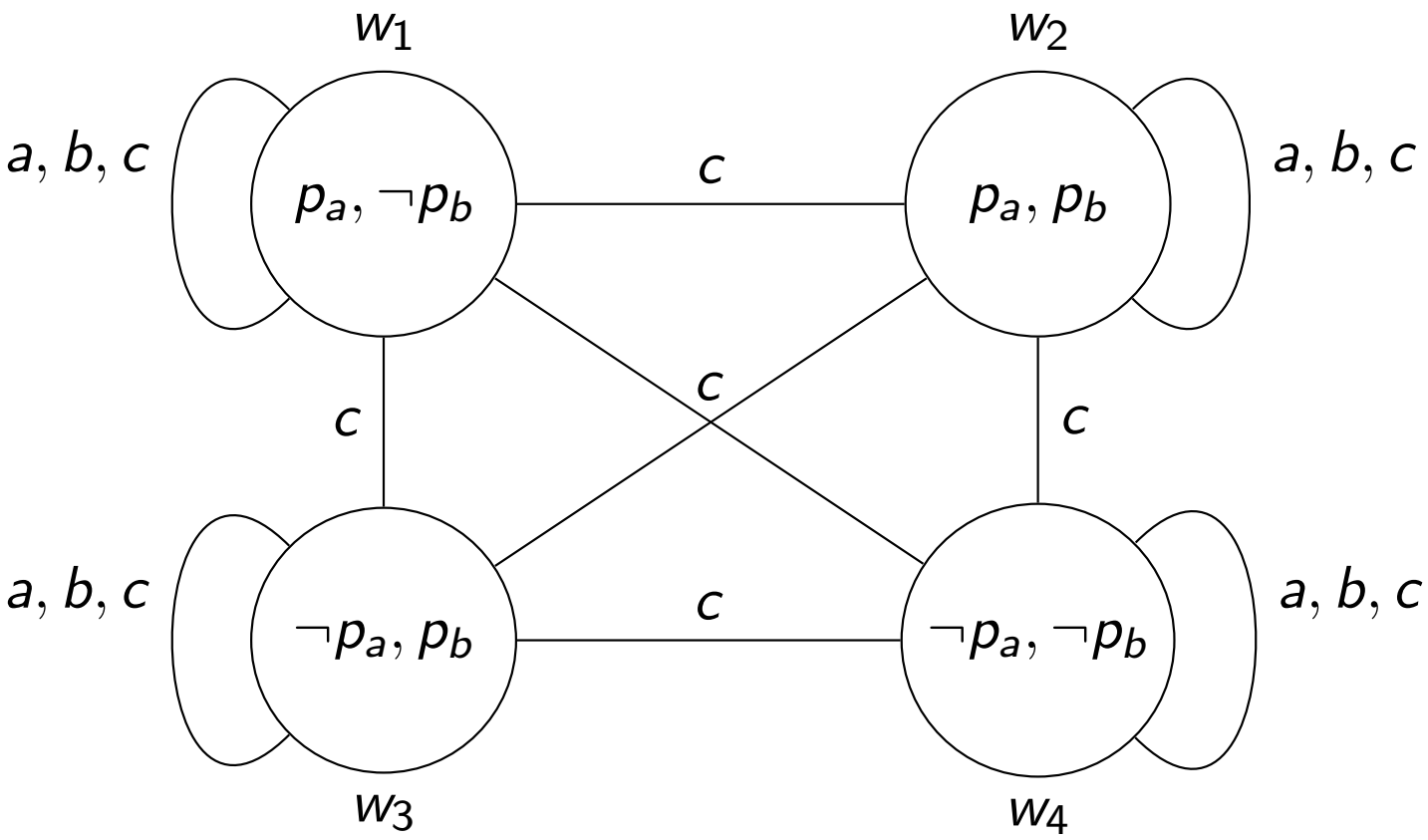
Definition (Semantics)

$$(M, w) \models [\![G\rangle\!]\varphi \quad \text{iff} \quad \forall\psi_G \exists\chi_{A\setminus G} : (M, w) \models \psi_G \to \langle\psi_G \wedge \chi_{A\setminus G}\rangle\varphi$$
$$(M, w) \models \langle\!\![G]\!\!\rangle\varphi \quad \text{iff} \quad \exists\psi_G \forall\chi_{A\setminus G} : (M, w) \models \psi_G \wedge [\psi_G \wedge \chi_{A\setminus G}]\varphi$$

Example

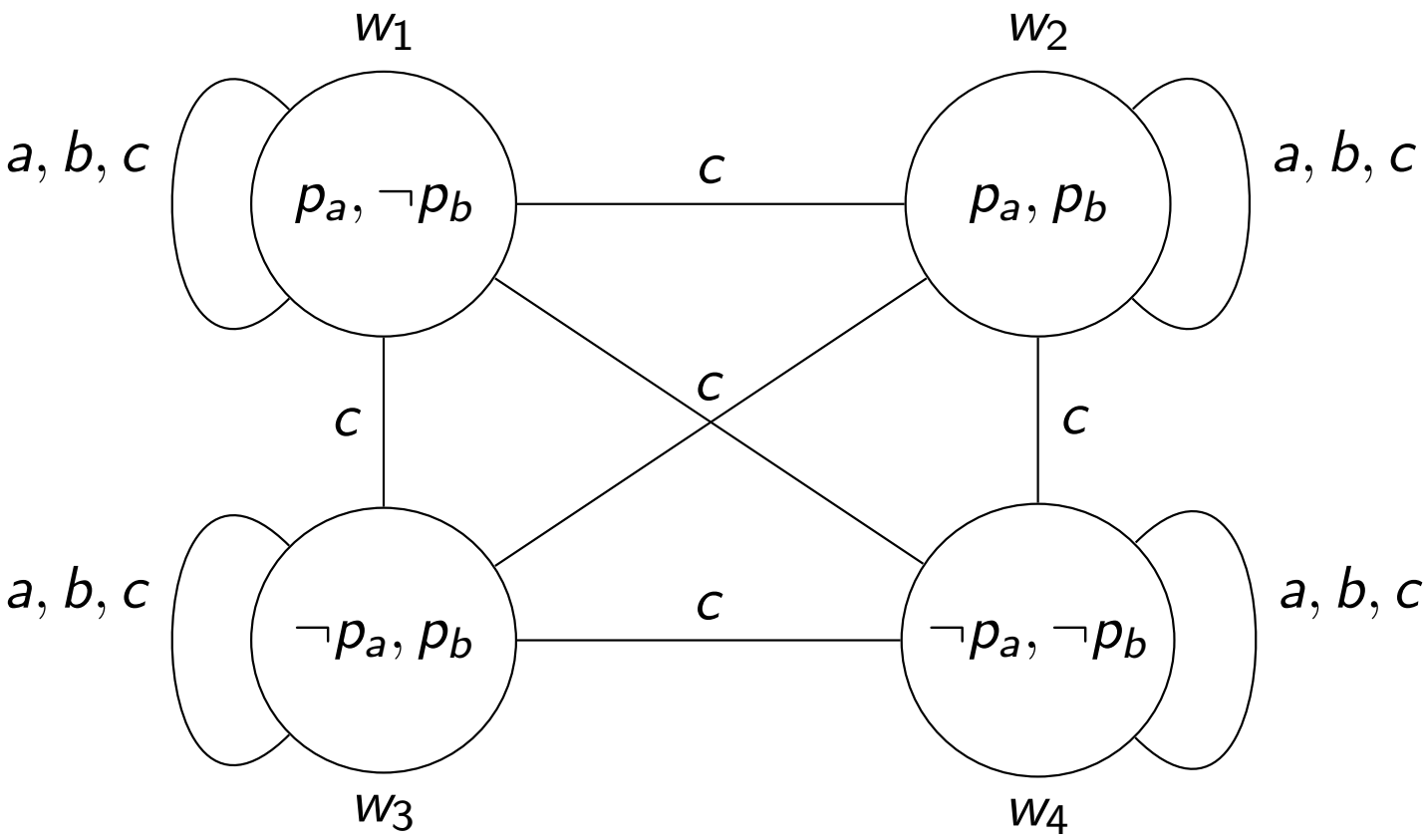
$$(M, w_1) \models \langle\!\langle \{a, b\} \rangle\!\rangle (K_c((p_a \wedge \neg p_b) \vee (\neg p_a \wedge p_b)) \wedge \neg(K_c(p_a \wedge \neg p_b) \vee K_c(\neg p_a \wedge p_b))), (M, w_1) \models \neg \langle\!\langle \{a\} \rangle\!\rangle \varphi.$$

Example



$$(M, w_1) \models \langle\!\langle \{a, b\} \rangle\!\rangle (K_c((p_a \wedge \neg p_b) \vee (\neg p_a \wedge p_b)) \wedge \neg(K_c(p_a \wedge \neg p_b) \vee K_c(\neg p_a \wedge p_b))), (M, w_1) \models \neg\langle\!\langle \{a\} \rangle\!\rangle \varphi.$$

Example



$$(M, w_1) \models \langle\!\langle \{a, b\} \rangle\!\rangle (K_c((p_a \wedge \neg p_b) \vee (\neg p_a \wedge p_b)) \wedge \neg(K_c(p_a \wedge \neg p_b) \vee K_c(\neg p_a \wedge p_b))), (M, w_1) \models \neg \langle\!\langle \{a\} \rangle\!\rangle \varphi.$$

Model Checking for CAL: Introduction

Definition (Model Checking for CAL)

Model-checking problem for CAL: given a pointed epistemic model (M, w) and a formula φ, determine whether $(M, w) \models \varphi$.

- Epistemic Planning: is there a strategy for a coalition such that some information is made known without revealing too much?
- Verification of Distributed Protocols: communication over a channel with an eavesdropper
- Producing the announcement that guarantees a certain output for a coalition (if possible)

Model Checking for CAL: Introduction

Definition (Model Checking for CAL)

Model-checking problem for CAL: given a pointed epistemic model (M, w) and a formula φ, determine whether $(M, w) \models \varphi$.

- Epistemic Planning: is there a strategy for a coalition such that some information is made known without revealing too much?
- Verification of Distributed Protocols: communication over a channel with an eavesdropper
- Producing the announcement that guarantees a certain output for a coalition (if possible)

Model Checking for CAL: Introduction

Definition (Model Checking for CAL)

Model-checking problem for CAL: given a pointed epistemic model (M, w) and a formula φ, determine whether $(M, w) \models \varphi$.

- Epistemic Planning: is there a strategy for a coalition such that some information is made known without revealing too much?
- Verification of Distributed Protocols: communication over a channel with an eavesdropper
- Producing the announcement that guarantees a certain output for a coalition (if possible)

Model Checking for CAL: General Idea

- Problem: implementing the truth definition directly requires checking results of announcing infinitely many formulas
- Hint: models for model checking are finite; for every agent there are finitely many ways to modify a model, and the same model update is a result of infinitely many announcements
- Solution: systematically define 'exemplar' announcements by an agent for every possible model update[4]

[4] Similar ideas to model checking of GAL: Thomas Ågotnes et al. "Group announcement logic". In: *Journal of Applied Logic* 8.1 (2010), pp. 62–81

Model Checking for CAL: General Idea

- Problem: implementing the truth definition directly requires checking results of announcing infinitely many formulas
- Hint: models for model checking are finite; for every agent there are finitely many ways to modify a model, and the same model update is a result of infinitely many announcements
- Solution: systematically define 'exemplar' announcements by an agent for every possible model update[4]

[4] Similar ideas to model checking of GAL: Thomas Ågotnes et al. "Group announcement logic". In: *Journal of Applied Logic* 8.1 (2010), pp. 62–81

Model Checking for CAL: Machinery I

Given a finite model (M, w), distinguishing formula δ_w is constructed recursively as follows:

$$\delta_w^{k+1} ::= \delta_w^0 \wedge \bigwedge_{a \in A} \Big(\bigwedge_{w \sim_a v} \widehat{K}_a \delta_v^k \wedge K_a \bigvee_{w \sim_a v} \delta_v^k \Big),$$

where $0 \leq k < |W|$, and δ_w^0 is the conjunction of all literals that are true in w, i.e. $\delta_w^0 ::= \bigwedge_{w \in V(p)} p \wedge \bigwedge_{w \notin V(p)} \neg p$.

Theorem

Every pointed model (M, w) is distinguished from all other non-bisimilar pointed models (M, v) by some distinguishing formula $\delta_w \in \mathcal{L}_{EL}$.

A *distinguishing formula for a set of states S* is

$$\delta_S ::= \bigvee_{w \in S} \delta_w.$$

Model Checking for CAL: Machinery I

Given a finite model (M, w), distinguishing formula δ_w is constructed recursively as follows:

$$\delta_w^{k+1} ::= \delta_w^0 \wedge \bigwedge_{a \in A} \Big(\bigwedge_{w \sim_a v} \widehat{K}_a \delta_v^k \wedge K_a \bigvee_{w \sim_a v} \delta_v^k \Big),$$

where $0 \leq k < |W|$, and δ_w^0 is the conjunction of all literals that are true in w, i.e. $\delta_w^0 ::= \bigwedge_{w \in V(p)} p \wedge \bigwedge_{w \notin V(p)} \neg p$.

Theorem

Every pointed model (M, w) is distinguished from all other non-bisimilar pointed models (M, v) by some distinguishing formula $\delta_w \in \mathcal{L}_{EL}$.

A *distinguishing formula for a set of states S* is

$$\delta_S ::= \bigvee_{w \in S} \delta_w.$$

Model Checking for CAL: Machinery I

Given a finite model (M, w), distinguishing formula δ_w is constructed recursively as follows:

$$\delta_w^{k+1} ::= \delta_w^0 \wedge \bigwedge_{a \in A} \Big(\bigwedge_{w \sim_a v} \widehat{K}_a \delta_v^k \wedge K_a \bigvee_{w \sim_a v} \delta_v^k \Big),$$

where $0 \leq k < |W|$, and δ_w^0 is the conjunction of all literals that are true in w, i.e. $\delta_w^0 ::= \bigwedge_{w \in V(p)} p \wedge \bigwedge_{w \notin V(p)} \neg p$.

Theorem

Every pointed model (M, w) is distinguished from all other non-bisimilar pointed models (M, v) by some distinguishing formula $\delta_w \in \mathcal{L}_{EL}$.

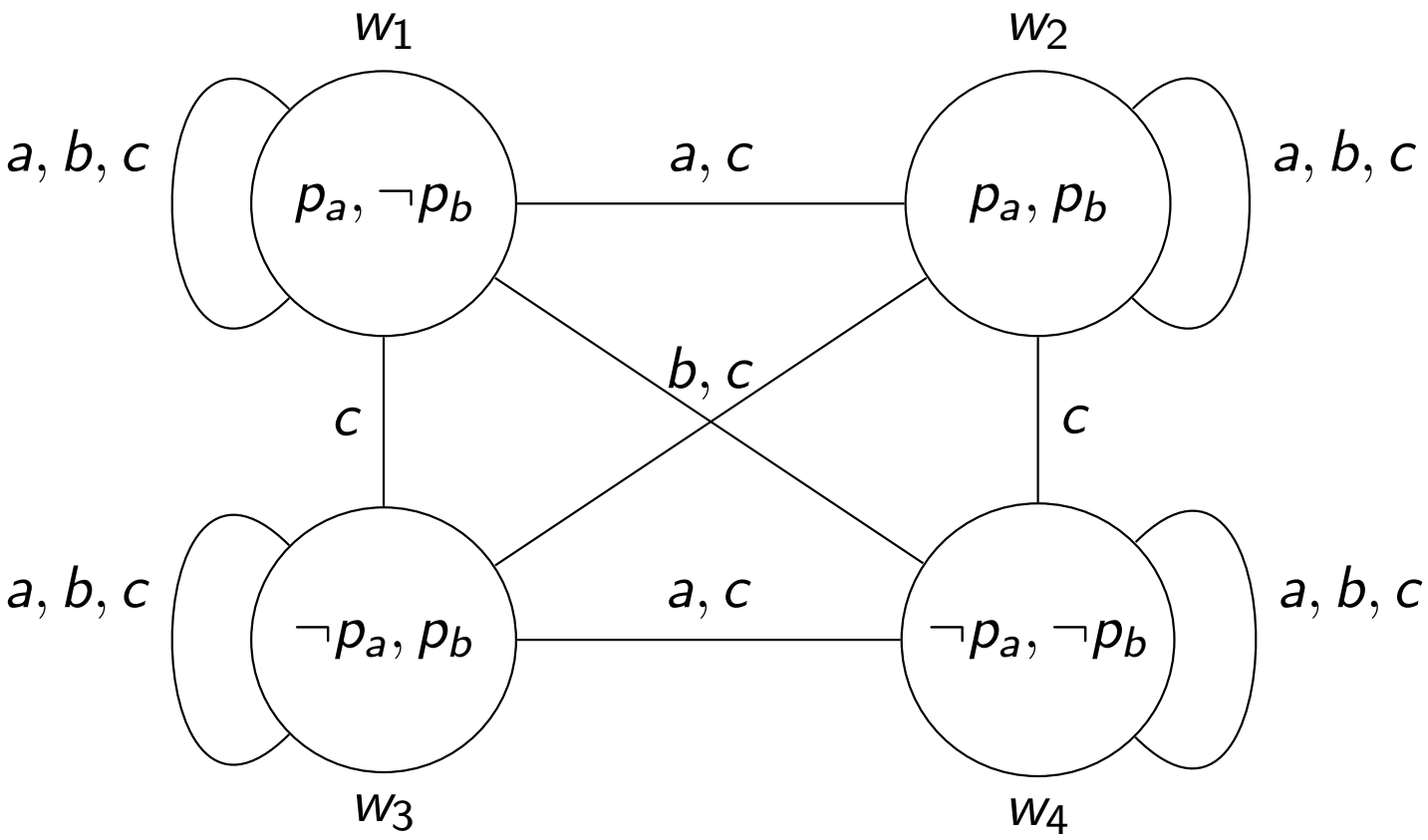
A *distinguishing formula for a set of states S* is

$$\delta_S ::= \bigvee_{w \in S} \delta_w.$$

A Modified Example

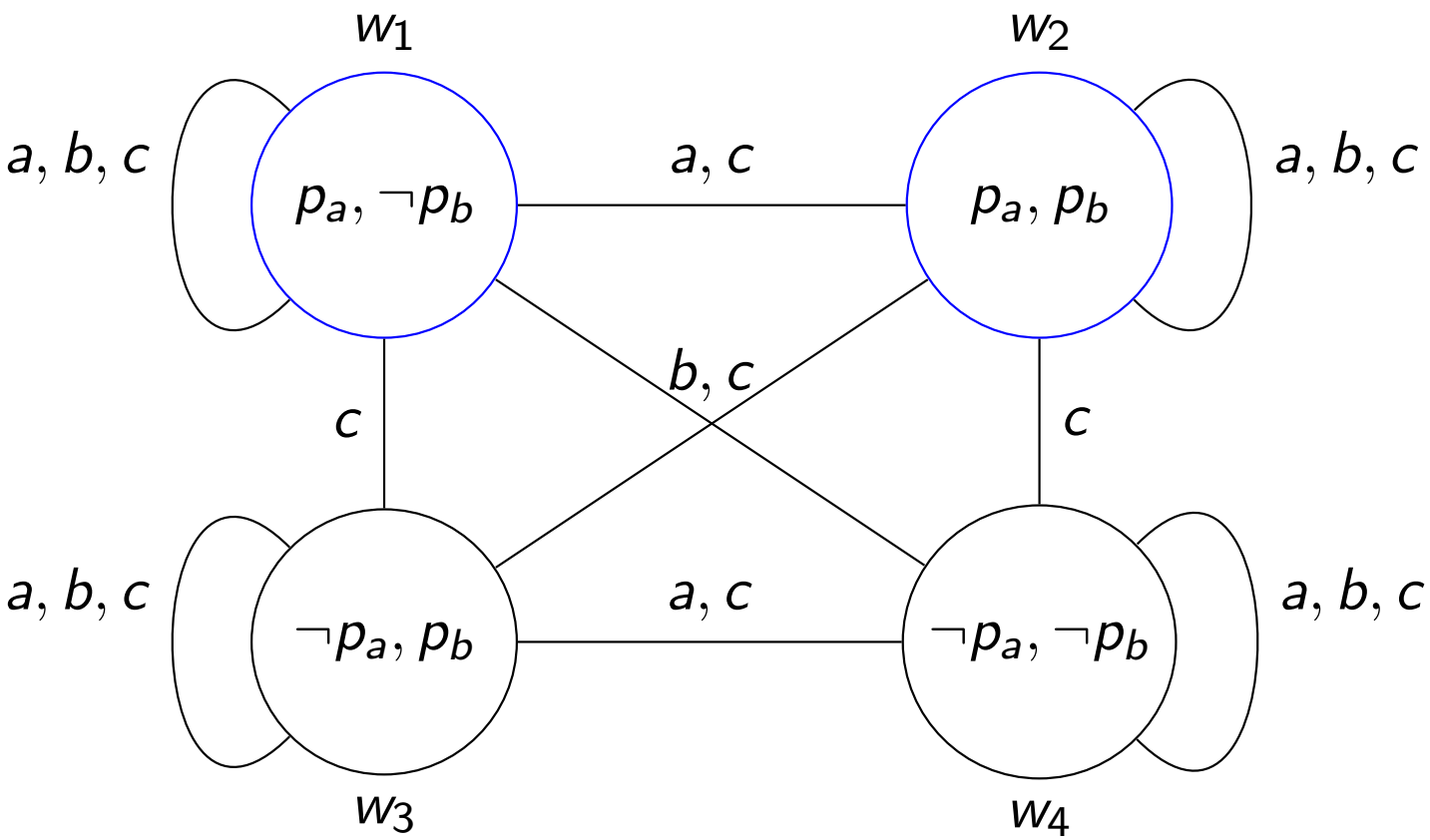
Agents a and b are unaware of each others states.

A Modified Example



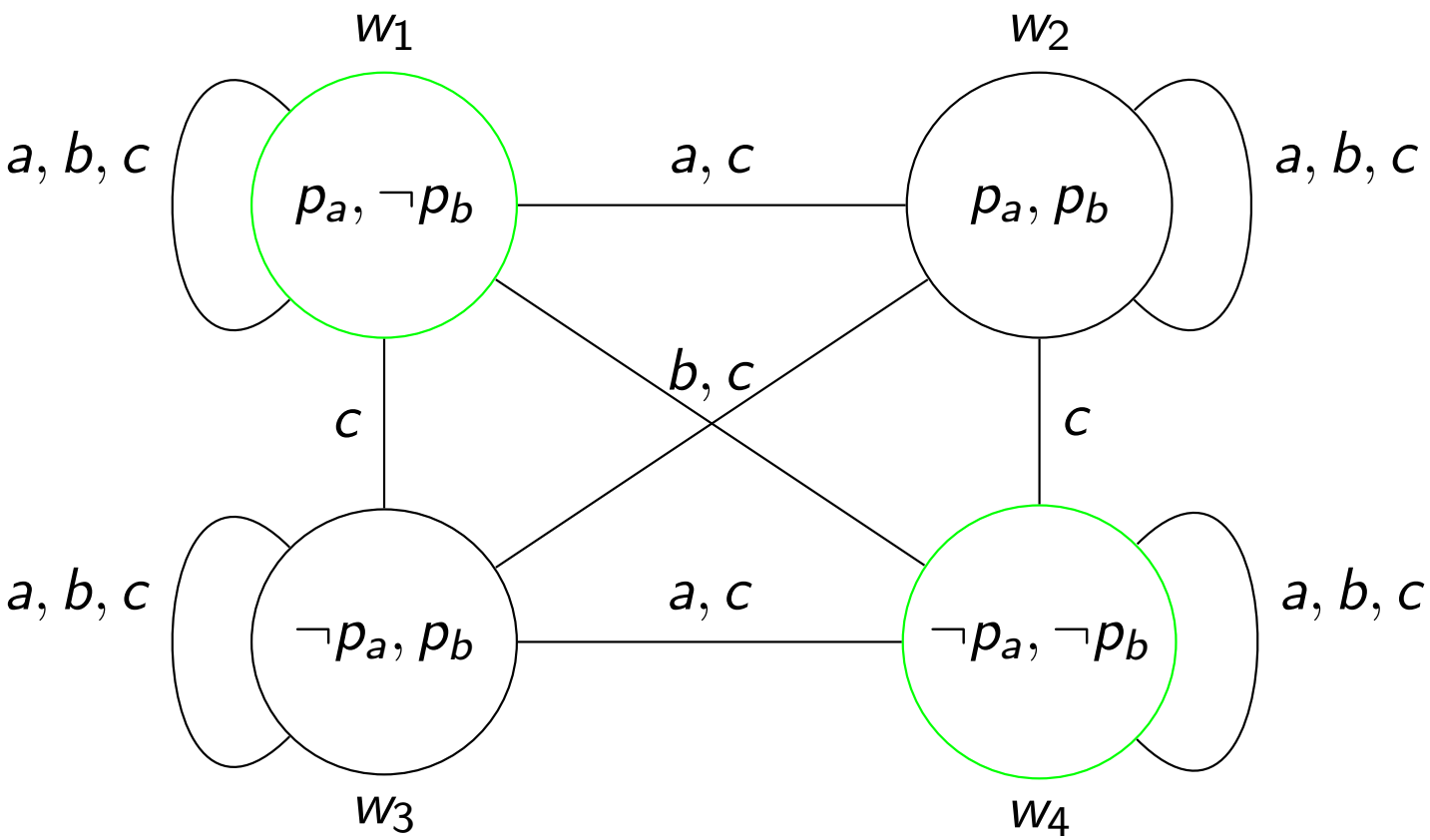
Agents a and b are unaware of each others states.

A Modified Example



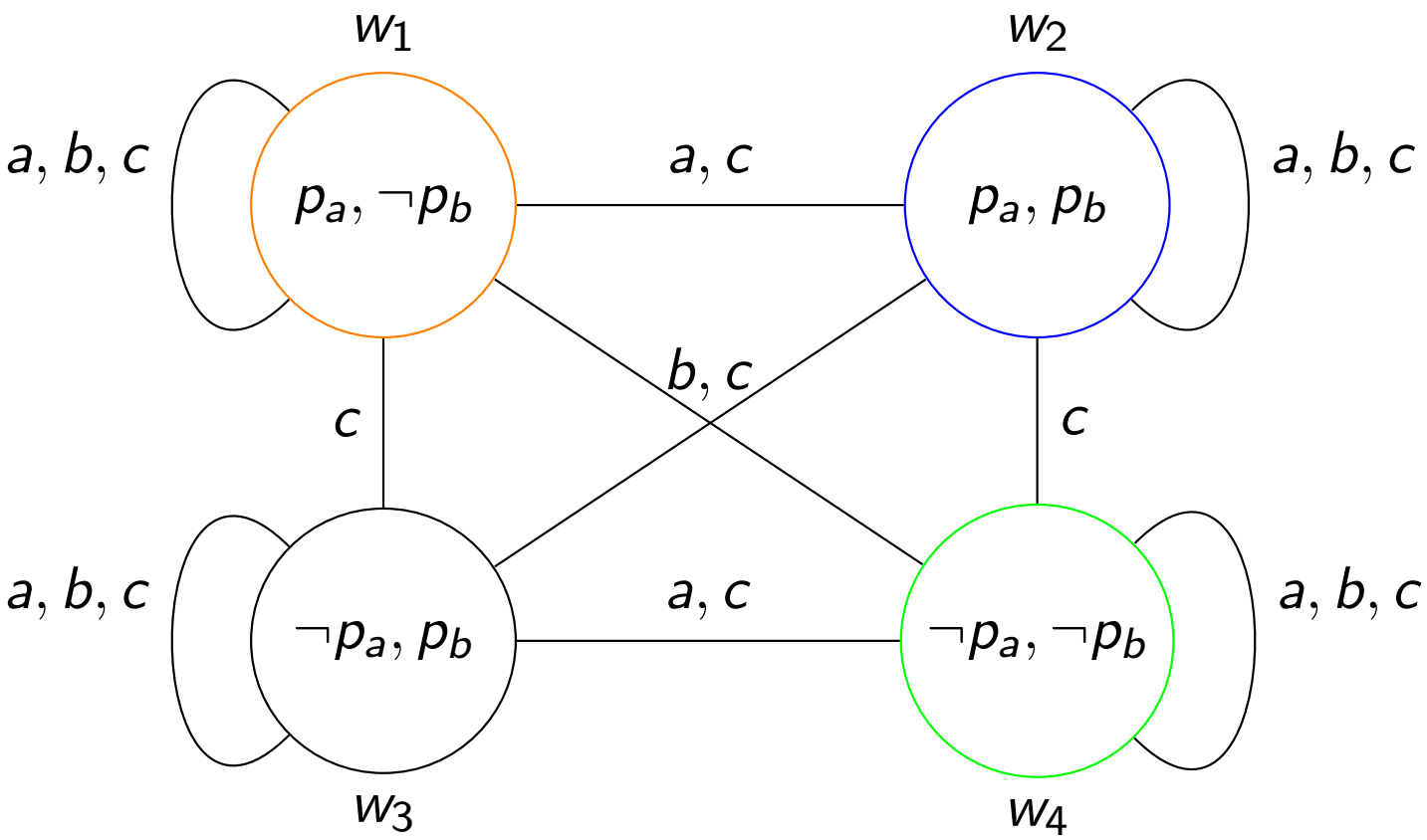
Agent's a equivalence class (announcement $K_a p_a$)

A Modified Example



Agent's b equivalence class (announcement $K_b \neg p_b$)

A Modified Example



equivalence class $\cap$ equivalence class $=$ equivalence class

(announcement $K_a p_a \wedge K_b \neg p_b$)

Model Checking for CAL: Machinery II

Definition (Strategies)

A strategy X_a for an agent a in a finite model (M, w) is a union of equivalence classes of a including $[w]_a$. Let denote the set of all available strategies of a as $S(a, w)$. Coalition strategy X_G is defined as $\cap_{a \in G} X_a$ for all $a \in G$. The set of available strategies for a coalition of agents G is denoted as $S(G, w)$.

A distinguishing formula δ_{X_G} for X_G is $\bigvee_{w \in X_G} \delta_w$.
Given that the number of strategies is always finite, we can a new finitary definition to the coalition announcement operator:

$$(M, w) \models \langle\!\langle G \rangle\!\rangle \varphi \text{ iff}$$
$$\exists X_G \in S(G, w)\ \forall X_{A \setminus G} \in S(A \setminus G, w) : (M, w)^{X_G \cap X_{A \setminus G}} \models \varphi$$

Model Checking for CAL: Machinery II

Definition (Strategies)

A strategy X_a for an agent a in a finite model (M, w) is a union of equivalence classes of a including $[w]_a$. Let denote the set of all available strategies of a as $S(a, w)$. Coalition strategy X_G is defined as $\cap_{a \in G} X_a$ for all $a \in G$. The set of available strategies for a coalition of agents G is denoted as $S(G, w)$.

A distinguishing formula δ_{X_G} for X_G is $\bigvee_{w \in X_G} \delta_w$.
Given that the number of strategies is always finite, we can a new finitary definition to the coalition announcement operator:

$$(M, w) \models \langle\!\langle G \rangle\!\rangle \varphi \text{ iff}$$
$$\exists X_G \in S(G, w)\ \forall X_{A \setminus G} \in S(A \setminus G, w) : (M, w)^{X_G \cap X_{A \setminus G}} \models \varphi$$

Model Checking for CAL: The Algorithm

Algorithm $mc(M, w, \varphi_0)$

case φ_0

p : **if** $w \in V(p)$ **then return** *true* **else return** *false*;

$\neg\varphi$: **if** $\neg mc(M, w, \varphi)$ **then return** *true* **else return** *false*;

$\varphi \wedge \psi$: **if** $mc(M, w, \varphi)$ and $mc(M, w, \psi)$ **then return** *true*
else return *false*;

$K_a\varphi$: **for all** $v \sim_a w$
if $\neg mc(M, v, \varphi)$ **then return** *false*;
return *true*

Model Checking for CAL: The Algorithm

Algorithm $mc(M, w, \varphi_0)$

case φ_0

p : **if** $w \in V(p)$ **then** **return** *true* **else return** *false*;

$\neg\varphi$: **if** $\neg mc(M, w, \varphi)$ **then return** *true* **else return** *false*;

$\varphi \wedge \psi$: **if** $mc(M, w, \varphi)$ and $mc(M, w, \psi)$ **then** **return** *true*
else return *false*;

$K_a\varphi$: **for all** $v \sim_a w$
if $\neg mc(M, v, \varphi)$ **then return** *false*;
return *true*

Model Checking for CAL: The Algorithm Continued

$\langle\psi\rangle\varphi$ **if** $\neg mc(M, w, \psi)$ **then return** *false*, **else** compute the ψ-submodel of M and **return** $mc(M^{\psi}, w, \varphi)$.

$\langle\!\langle G \rangle\!\rangle\varphi$: compute $(\|M\|, w)$ and sets of strategies $S(G, w)$ and $S(A \setminus G, w)$

for all $X_G \in S(G, w)$

check $=$ *true*;

for all $X_{A\setminus G} \in S(A \setminus G, w)$

if $\neg mc(\|M\|^{X_G \cap X_{A\setminus G}}, w, \varphi)$ **then**

check $=$ *false*

if *check* **then return** *true*

return *false*.

Complexity of The Model Checking Problem

Theorem

The model checking problem for CAL is PSPACE-complete.

Complexity of The Model Checking Problem

Theorem

The model checking problem for CAL is PSPACE-complete.

Conclusions and Further Research

- PSPACE is a manageable complexity
- We can use model checking to verify consequences of coalition announcements (for example, communication protocols, or data collection)
- We can also use it to produce strategies (the right announcements to make) given the properties that should hold after the announcement
- The satisfiability problem for CAL is undecidable[5]. Finding its decidable fragments is an open problem

[5] Thomas Ågotnes, Hans van Ditmarsch, and Timothy Stewart French. "The Undecidability of Quantified Announcements". In: *Studia Logica* 104.4 (2016), pp. 597–640.

Conclusions and Further Research

- PSPACE is a manageable complexity
- We can use model checking to verify consequences of coalition announcements (for example, communication protocols, or data collection)
- We can also use it to produce strategies (the right announcements to make) given the properties that should hold after the announcement
- The satisfiability problem for CAL is undecidable[5]. Finding its decidable fragments is an open problem

[5] Thomas Ågotnes, Hans van Ditmarsch, and Timothy Stewart French. "The Undecidability of Quantified Announcements". In: *Studia Logica* 104.4 (2016), pp. 597–640.

Thank you for attention!

Encore: Bisimulation

Let two models $M = (W, \sim V)$ and $M' = (W', \sim', V')$ be given. A non-empty binary relation $Z \subseteq W \times W'$ is called a *bisimulation* if and only if for all $w \in W$ and $w' \in W'$ with $(w, w') \in Z$:

- w and w' satisfy the same propositional variables;
- for all $a \in A$ and all $v \in W$: if $w \sim_a v$, then there is a v' such that $w' \sim_a v'$ and $(v, v') \in Z$;
- for all $a \in A$ and all $v' \in W'$: if $w' \sim_a v'$, then there is a v such that $w \sim_a v$ and $(v, v') \in Z$.

Encore: Bisimulation Contraction

- The *quotient model* of M with respect to some relation R is $M^R = (W^R, \sim^R, V^R)$, where $W^R = \{[w] \mid w \in W\}$ and $[w] = \{v \mid wRv\}$, $[w] \sim_a^R [v]$ iff $\exists w' \in [w]$, $\exists v' \in [v]$ such that $w' \sim_a v'$ in M, and $[w] \in V^R(p)$ iff $\forall w' \in [w] : w' \in V(p)$.
- *Bisimulation contraction* of M (written $\|M\|$) is the quotient model of M with respect to the maximal bisimulation of M with itself.
- A model M is *bisimulation contracted* if M is isomorphic to $\|M\|$.
- $(\|M\|, w) \models \varphi$ iff $(M, w) \models \varphi$ for all $\varphi \in \mathcal{L}_{CAL}$.