Reasoning about Knowledge and Ability

Thomas Ågotnes[1,2]

[1]University of Bergen, Norway

[2]Southwest University, Chongqing, China

Workshop on Planning, Logic and Social Intelligence, DTU
4 April 2014

Based on joint work with

- Hans van Ditmarsch
- Philippe Balbiani
- Wojtek Jamroga
- Pablo Seban
- ...

Introduction

In this talk I

- Briefly review *ATL*
- Talk about interesting issues that occur when epistemic logic and ATL is combined in order to reason about strategic reasoning under imperfect information
- In particular look at the case when actions are public announcements (group announcement logic)

Contents

1 ATL
2 Strategic Reasoning under Imperfect Information
3 Group Announcement Logic

ATL

- *Alternating-time Temporal Logic (ATL)* is an agentized extension of CTL introduced by Alur and colleagues (1997)
- The language of ATL is obtained by replacing A and E with $\langle\langle C \rangle\rangle$ where $C \subseteq Ag$ and Ag is the finite set of all agents in the system
- Intuitively,

$$\langle\langle C \rangle\rangle \Diamond \phi$$

means that

  - C can cooperate to ensure that ϕ becomes true sometime in the future no matter what the other agents do (and similarly for $\bigcirc, \Box, \mathcal{U}$)
  - C has a strategy to enforce that ϕ becomes true sometime in the future
  - Is used to reason about *game-like* situations

Example

$$\langle\langle merkel, obama \rangle\rangle \Diamond \neg crisis$$

Merkel and Obama can cooperate to ensure that at some point in the future the crisis is over

Example

$$\langle\langle Ann \rangle\rangle \Box \langle\langle Bob \rangle\rangle \Diamond win$$

Definition (ATL models)

A concurrent game structure is a tuple $M = \langle \mathbb{A}\mathrm{gt}, St, \pi, Act, d, o \rangle$, where:

- $\mathbb{A}\mathrm{gt}$: a finite set of all agents
- St: a set of states
- π: a valuation of propositions
- Act: a finite set of (atomic) actions
- $d : \mathbb{A}\mathrm{gt} \times St \rightarrow \wp(Act)$ defines actions available to an agent in a state
- o: a deterministic transition function that assigns outcome states $q' = o(q, \alpha_1, \ldots, \alpha_k)$ to states and tuples of actions

Definition (ATL models)

A concurrent game structure is a tuple $M = \langle \mathbb{A}\mathrm{gt}, St, \pi, Act, d, o \rangle$, where:

- $\mathbb{A}\mathrm{gt}$: a finite set of all agents
- St: a set of states
- π: a valuation of propositions
- Act: a finite set of (atomic) actions
- $d : \mathbb{A}\mathrm{gt} \times St \to \wp(Act)$ defines actions available to an agent in a state
- o: a deterministic transition function that assigns outcome states $q' = o(q, \alpha_1, \ldots, \alpha_k)$ to states and tuples of actions

Definition (ATL models)

A concurrent game structure is a tuple $M = \langle \mathbb{A}\mathrm{gt}, St, \pi, Act, d, o \rangle$, where:

- $\mathbb{A}\mathrm{gt}$: a finite set of all agents
- St: a set of states
- π: a valuation of propositions
- Act: a finite set of (atomic) actions
- $d : \mathbb{A}\mathrm{gt} \times St \to \wp(Act)$ defines actions available to an agent in a state
- o: a deterministic transition function that assigns outcome states $q' = o(q, \alpha_1, \ldots, \alpha_k)$ to states and tuples of actions

Definition (ATL models)

A concurrent game structure is a tuple $M = \langle \mathbb{A}\mathrm{gt}, St, \pi, Act, d, o \rangle$, where:

- $\mathbb{A}\mathrm{gt}$: a finite set of all agents
- St: a set of states
- π: a valuation of propositions
- Act: a finite set of (atomic) actions
- $d : \mathbb{A}\mathrm{gt} \times St \rightarrow \wp(Act)$ defines actions available to an agent in a state
- o: a deterministic transition function that assigns outcome states $q' = o(q, \alpha_1, \ldots, \alpha_k)$ to states and tuples of actions

Definition (ATL models)

A concurrent game structure is a tuple $M = \langle \mathbb{A}\mathrm{gt}, St, \pi, Act, d, o \rangle$, where:

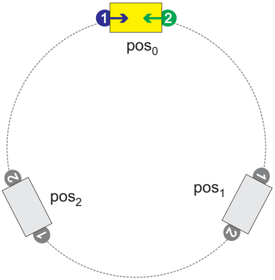
- $\mathbb{A}\mathrm{gt}$: a finite set of all agents
- St: a set of states
- π: a valuation of propositions
- Act: a finite set of (atomic) actions
- $d : \mathbb{A}\mathrm{gt} \times St \rightarrow \wp(Act)$ defines actions available to an agent in a state
- o: a deterministic transition function that assigns outcome states $q' = o(q, \alpha_1, \ldots, \alpha_k)$ to states and tuples of actions

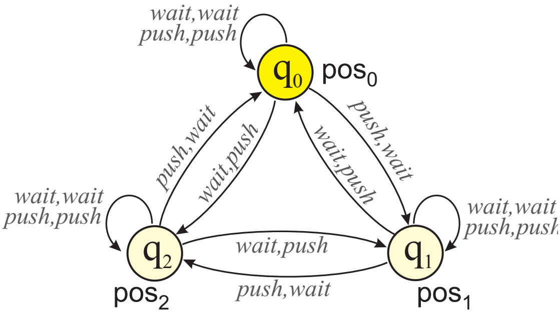
Definition (ATL models)

A concurrent game structure is a tuple $M = \langle \mathbb{A}\mathrm{gt}, St, \pi, Act, d, o \rangle$, where:

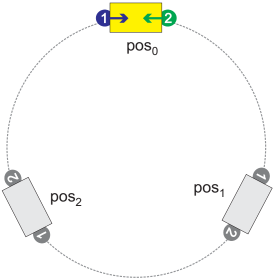
- $\mathbb{A}\mathrm{gt}$: a finite set of all agents
- St: a set of states
- π: a valuation of propositions
- Act: a finite set of (atomic) actions
- $d : \mathbb{A}\mathrm{gt} \times St \to \wp(Act)$ defines actions available to an agent in a state
- o: a deterministic transition function that assigns outcome states $q' = o(q, \alpha_1, \ldots, \alpha_k)$ to states and tuples of actions

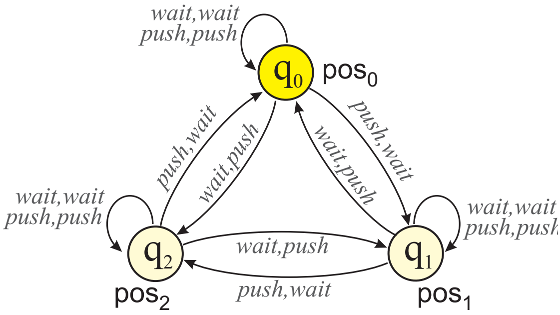
Example: Robots and Carriage





Example: Robots and Carriage





Definition (Strategy)

A strategy is a conditional plan.

We represent strategies by functions $s_a : St \to Act$.

$\leadsto$ *memoryless agents*

Alternative: perfect recall strategies $s_a : St^+ \to Act$

Function $out(q, s_A)$ returns the set of all paths that may result from agents A executing strategy s_A from state q onward.

Definition (Strategy)

A strategy is a conditional plan.

We represent strategies by functions $s_a : St \to Act$.

$\rightsquigarrow$ *memoryless agents*

Alternative: perfect recall strategies $s_a : St^+ \to Act$

Function $out(q, s_A)$ returns the set of all paths that may result from agents A executing strategy s_A from state q onward.

Definition (Strategy)

A strategy is a conditional plan.
We represent strategies by functions $s_a : St \to Act$.

$\rightsquigarrow$ *memoryless agents*

Alternative: perfect recall strategies $s_a : St^+ \to Act$

Function $out(q, s_A)$ returns the set of all paths that may result from agents A executing strategy s_A from state q onward.

Definition (Strategy)

A strategy is a conditional plan.
We represent strategies by functions $s_a : St \to Act$.

$\leadsto$ *memoryless agents*

Alternative: perfect recall strategies $s_a : St^+ \to Act$

Function $out(q, s_A)$ returns the set of all paths that may result from agents A executing strategy s_A from state q onward.

Definition (Strategy)

A strategy is a conditional plan.
We represent strategies by functions $s_a : St \to Act$.

$\leadsto$ *memoryless agents*

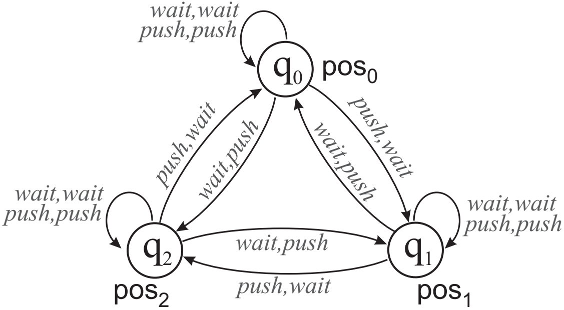
Alternative: perfect recall strategies $s_a : St^+ \to Act$

Function $out(q, s_A)$ returns the set of all paths that may result from agents A executing strategy s_A from state q onward.

Semantics

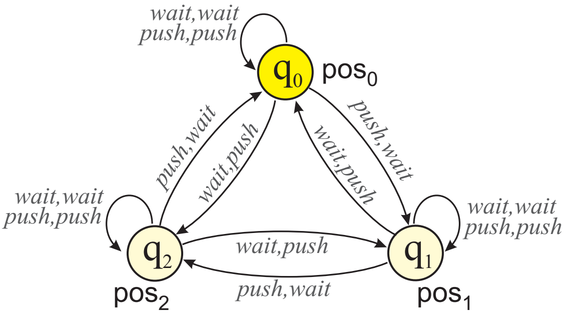
$M, q \models p$	iff p is in $\pi(q)$;
$M, q \models \neg\varphi$	iff $M, q \not\models \varphi$;
$M, q \models \varphi_1 \land \varphi_2$	iff $M, q \models \varphi_1$ and $M, q \models \varphi_2$;
$M, q \models \langle\langle A\rangle\rangle \bigcirc \varphi$	iff there is s_A such that, for every $\lambda \in out(q, s_A)$, we have $M, \lambda[1] \models \varphi$;
$M, q \models \langle\langle A\rangle\rangle \Box \varphi$	iff there is s_A such that, for every $\lambda \in out(q, s_A)$, we have $M, \lambda[i] \models \varphi$ for all $i \geq 0$;
$M, q \models \langle\langle A\rangle\rangle \varphi_1 \mathcal{U} \varphi_2$	iff there is s_A such that, for every $\lambda \in out(q, s_A)$, we have $M, \lambda[i] \models \varphi_2$ for some $i \geq 0$ and $M, \lambda[j] \models \varphi_1$ for all $0 \leq j \leq i$.

Example: Robots and Carriage



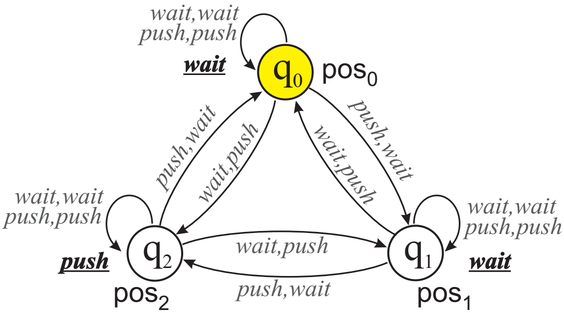
$$pos_0 \rightarrow \langle\langle 1 \rangle\rangle \Box \neg pos_1$$

Example: Robots and Carriage



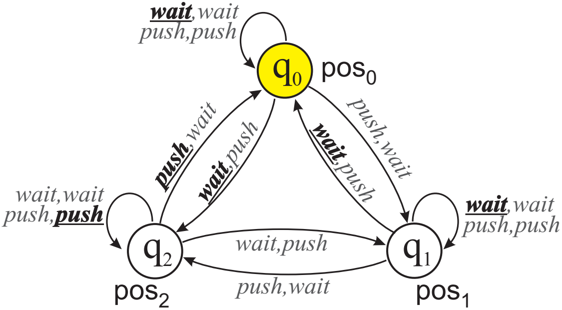
$$pos_0 \rightarrow \langle\langle 1 \rangle\rangle \Box \neg pos_1$$

Example: Robots and Carriage



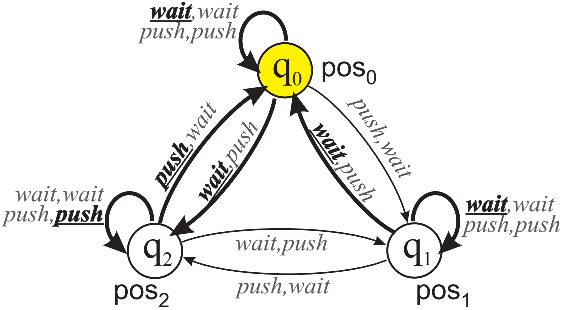
$$pos_0 \rightarrow \langle\langle 1 \rangle\rangle \Box \neg pos_1$$

Example: Robots and Carriage



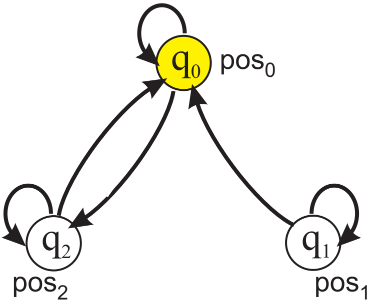
$$pos_0 \rightarrow \langle\!\langle 1 \rangle\!\rangle \Box \neg pos_1$$

Example: Robots and Carriage



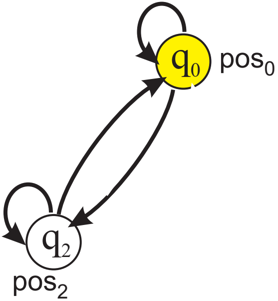
$pos_0 \to \langle\langle 1 \rangle\rangle \Box \neg pos_1$

Example: Robots and Carriage



$$\mathsf{pos}_0 \rightarrow \langle\langle 1 \rangle\rangle \Box \neg \mathsf{pos}_1$$

Example: Robots and Carriage



$$\mathsf{pos}_0 \rightarrow \langle\langle 1 \rangle\rangle \Box \neg \mathsf{pos}_1$$

Fixpoint Properties

Theorem

The following formulae are valid for ATL:

- $\langle\langle A\rangle\rangle\Box\varphi \quad\leftrightarrow\quad \varphi \wedge \langle\langle A\rangle\rangle\bigcirc\langle\langle A\rangle\rangle\Box\varphi$
- $\langle\langle A\rangle\rangle\varphi_1\,\mathcal{U}\,\varphi_2 \quad\leftrightarrow\quad \varphi_2 \vee \varphi_1 \wedge \langle\langle A\rangle\rangle\bigcirc\langle\langle A\rangle\rangle\varphi_1\,\mathcal{U}\,\varphi_2.$

Corollary

Strategy for A can be synthesized incrementally (no backtracking is necessary).

Fixpoint Properties

Theorem

The following formulae are valid for ATL:

- $\langle\langle A\rangle\rangle\Box\varphi \quad\leftrightarrow\quad \varphi\wedge\langle\langle A\rangle\rangle\bigcirc\langle\langle A\rangle\rangle\Box\varphi$
- $\langle\langle A\rangle\rangle\varphi_1\,\mathcal{U}\,\varphi_2 \quad\leftrightarrow\quad \varphi_2\vee\varphi_1\wedge\langle\langle A\rangle\rangle\bigcirc\langle\langle A\rangle\rangle\varphi_1\,\mathcal{U}\,\varphi_2.$

Corollary

Strategy for A can be synthesized incrementally (no backtracking is necessary).

Contents

1 ATL

2 Strategic Reasoning under Imperfect Information
- Combining Dimensions
- Constructive Strategic Logic
- Constructive Knowledge
- Between Perception and Recall

3 Group Announcement Logic

ATL and epistemic logic can be combined to allow strategic reasoning under imperfect information

- We extend CGSs with indistinguishability relations $\sim_a$, one per agent
- We add epistemic operators to ATL

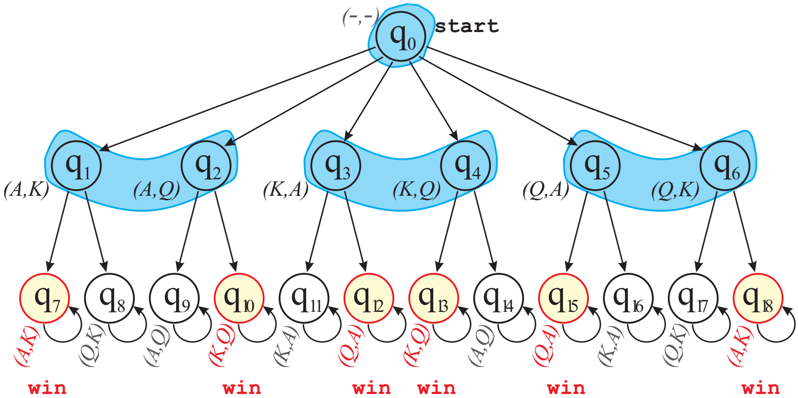
$\rightsquigarrow$ Problems!

ATL and epistemic logic can be combined to allow strategic reasoning under imperfect information

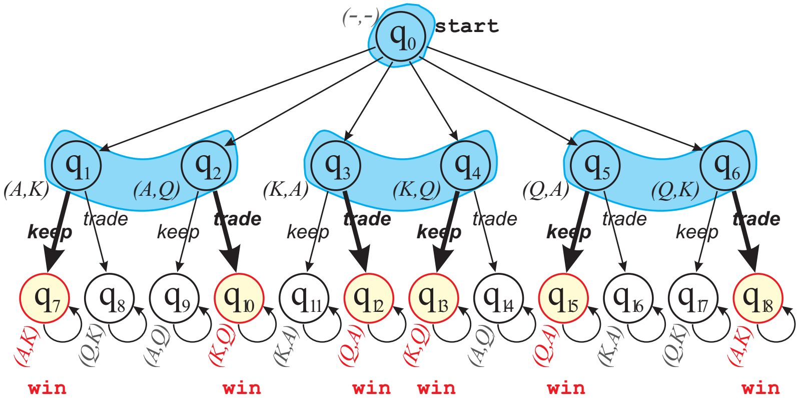
- We extend CGSs with indistinguishability relations $\sim_a$, one per agent
- We add epistemic operators to ATL

$\leadsto$ Problems!

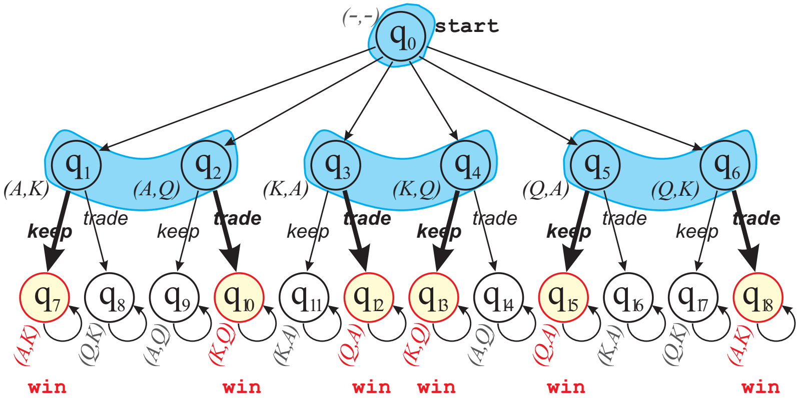
Combining Dimensions



Combining Dimensions

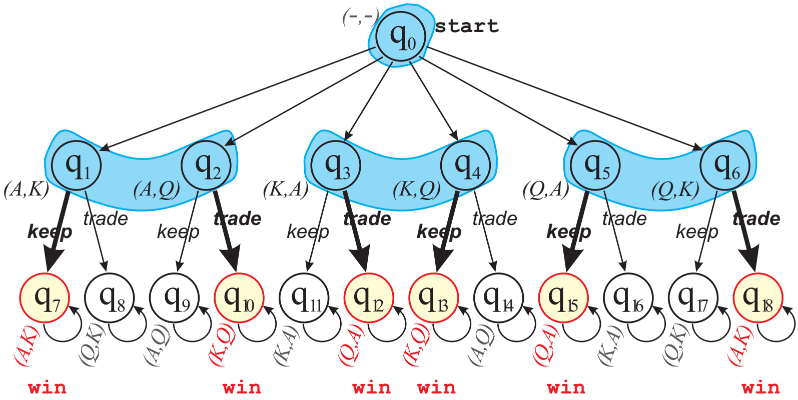


Combining Dimensions



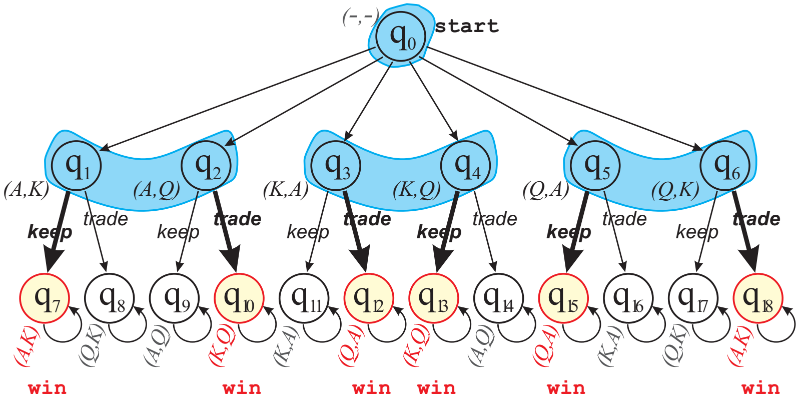
$$start \rightarrow \langle\langle a \rangle\rangle \Diamond win$$

Combining Dimensions



$$start \rightarrow \langle\langle a \rangle\rangle \Diamond win$$

$$start \rightarrow K_a \langle\langle a \rangle\rangle \Diamond win$$



$$start \rightarrow \langle\langle a \rangle\rangle \Diamond win$$

$$start \rightarrow K_a \langle\langle a \rangle\rangle \Diamond win$$

Does it make sense?

Problem:

Strategic and epistemic abilities are *not* independent!

$\langle\langle A\rangle\rangle\Phi$ = A can enforce Φ

It should at least mean that A are able to identify and execute the right strategy!

Executable strategies = uniform strategies

Problem:

Strategic and epistemic abilities are *not* independent!

$\langle\langle A\rangle\rangle\Phi$ = A can enforce Φ

It should at least mean that A are able to identify and execute the right strategy!

Executable strategies = uniform strategies

Problem:

Strategic and epistemic abilities are *not* independent!

$\langle\langle A\rangle\rangle\Phi$ = A can enforce Φ

It should at least mean that A are able to identify and execute the right strategy!

Executable strategies = uniform strategies

Problem:

Strategic and epistemic abilities are *not* independent!

$\langle\langle A \rangle\rangle \Phi$ = A can enforce Φ

It should at least mean that A are able to identify and execute the right strategy!

Executable strategies = uniform strategies

Definition (Uniform strategy)

Strategy s_a is uniform iff it specifies the same choices for indistinguishable situations:

- (no recall:) if $q \sim_a q'$ then $s_a(q) = s_a(q')$
- (perfect recall:) if $\lambda \approx_a \lambda'$ then $\Rightarrow s_a(\lambda) = s_a(\lambda)$, where $\lambda \approx_a \lambda'$ iff $\lambda[i] \sim_a \lambda'[i]$ for every i.

A collective strategy is uniform iff it consists only of uniform individual strategies.

Definition (Uniform strategy)

Strategy s_a is uniform iff it specifies the same choices for indistinguishable situations:

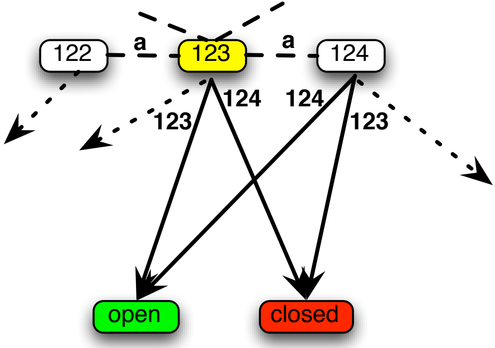
- (no recall:) if $q \sim_a q'$ then $s_a(q) = s_a(q')$
- (perfect recall:) if $\lambda \approx_a \lambda'$ then $\Rightarrow s_a(\lambda) = s_a(\lambda)$, where $\lambda \approx_a \lambda'$ iff $\lambda[i] \sim_a \lambda'[i]$ for every i.

A collective strategy is uniform iff it consists only of uniform individual strategies.

Note:

Having a successful strategy does not imply knowing that we have it!

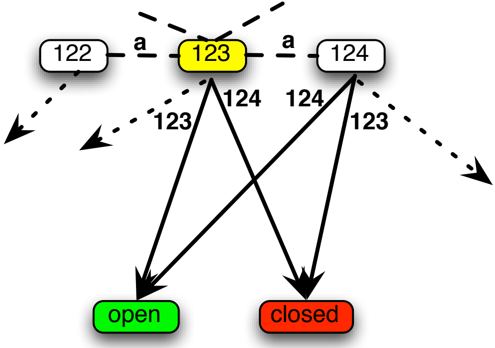
Example



$\langle\langle a \rangle\rangle \bigcirc open$

$K_a \langle\langle a \rangle\rangle \bigcirc open$

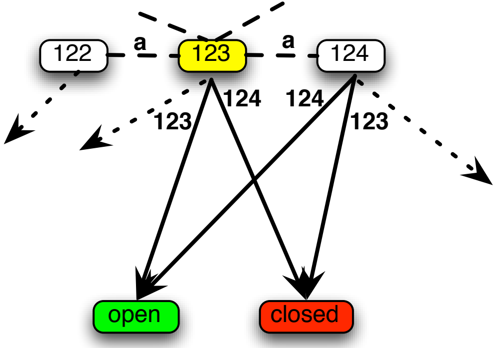
Example



$$\langle\langle a \rangle\rangle \bigcirc open$$

$$K_a \langle\langle a \rangle\rangle \bigcirc open$$

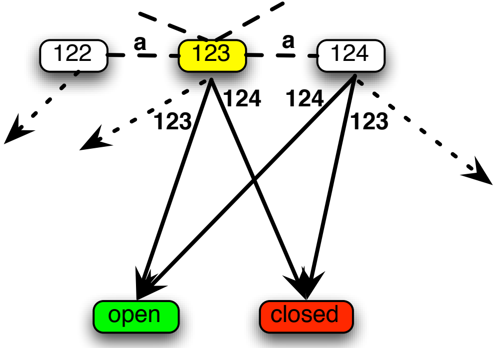
Example



$$\langle\langle a\rangle\rangle \bigcirc open$$

$$K_a\langle\langle a\rangle\rangle \bigcirc open$$

Example



$$\langle\langle a\rangle\rangle \bigcirc open$$

$$K_a\langle\langle a\rangle\rangle \bigcirc open$$

Note:

Knowing that a successful strategy exists does not imply knowing the strategy itself!

Levels of Strategic Ability

Our cases for $\langle\langle A \rangle\rangle \Phi$ under imperfect information:

1. There is σ (not necessarily executable!) such that, for every execution of σ, Φ holds
2. There is a uniform σ such that, for every execution of σ, Φ holds
3. A know that there is a uniform σ such that, for every execution of σ, Φ holds
4. There is a uniform σ such that A know that, for every execution of σ, Φ holds

From now on, we restrict our discussion to uniform memoryless strategies (unless explicitly stated otherwise).

Levels of Strategic Ability

Our cases for $\langle\langle A \rangle\rangle \Phi$ under imperfect information:

1. There is σ (not necessarily executable!) such that, for every execution of σ, Φ holds
2. There is a uniform σ such that, for every execution of σ, Φ holds
3. A know that there is a uniform σ such that, for every execution of σ, Φ holds
4. There is a uniform σ such that A know that, for every execution of σ, Φ holds

From now on, we restrict our discussion to uniform memoryless strategies (unless explicitly stated otherwise).

Levels of Strategic Ability

Our cases for $\langle\langle A\rangle\rangle\Phi$ under imperfect information:

1. There is σ (not necessarily executable!) such that, for every execution of σ, Φ holds
2. There is a uniform σ such that, for every execution of σ, Φ holds
3. *A* know that there is a uniform σ such that, for every execution of σ, Φ holds
4. There is a uniform σ such that *A* know that, for every execution of σ, Φ holds

From now on, we restrict our discussion to uniform memoryless strategies (unless explicitly stated otherwise).

Levels of Strategic Ability

Our cases for $\langle\langle A \rangle\rangle \Phi$ under imperfect information:

1. There is σ (not necessarily executable!) such that, for every execution of σ, Φ holds
2. There is a uniform σ such that, for every execution of σ, Φ holds
3. *A* know that there is a uniform σ such that, for every execution of σ, Φ holds
4. There is a uniform σ such that *A* know that, for every execution of σ, Φ holds

From now on, we restrict our discussion to uniform memoryless strategies (unless explicitly stated otherwise).

Levels of Strategic Ability

Our cases for $\langle\langle A \rangle\rangle \Phi$ under imperfect information:

1. There is σ (not necessarily executable!) such that, for every execution of σ, Φ holds
2. There is a uniform σ such that, for every execution of σ, Φ holds
3. *A* know that there is a uniform σ such that, for every execution of σ, Φ holds
4. There is a uniform σ such that *A* know that, for every execution of σ, Φ holds

From now on, we restrict our discussion to uniform memoryless strategies (unless explicitly stated otherwise).

Levels of Strategic Ability

Our cases for $\langle\langle A \rangle\rangle \Phi$ under imperfect information:

1. There is σ (not necessarily executable!) such that, for every execution of σ, Φ holds
2. There is a uniform σ such that, for every execution of σ, Φ holds
3. A know that there is a uniform σ such that, for every execution of σ, Φ holds
4. There is a uniform σ such that A know that, for every execution of σ, Φ holds

From now on, we restrict our discussion to uniform memoryless strategies (unless explicitly stated otherwise).

Knowing how to play

- It turns out that knowledge of ability *de re* is not expressible in the language
- In *Constructive strategic logic (CSL)* (Jamroga and Ågotnes, 2007) ATL is extended with *constructive knowledge operators* such that

$$\mathbb{K}_a \langle\langle a \rangle\rangle \phi$$

  means that a knows *de re* that she can achieve the goal

Constructive Strategic Logic: key idea

1. Interpret ability modalities in *sets* of states:
   - $M, Q \models \langle\langle a \rangle\rangle \phi$: there exists some strategy such that if a follows it *from any of the states in the set Q*, ϕ is guaranteed to be true
2. Introduce new *constructive knowledge* operators:
   - $M, q \models \mathbb{K}_a \phi \Leftrightarrow M, [q]_{\sim_a} \models \phi$

We get that:

$$M, q \models \mathbb{K}_a \langle\langle a \rangle\rangle \phi \Leftrightarrow M, [q]_{\sim_a} \models \langle\langle a \rangle\rangle \phi \Leftrightarrow$$

there exists some strategy such that if a follows it *from any of the states she considers possible*, ϕ is guaranteed to be true

Constructive Strategic Logic: key idea

1. Interpret ability modalities in *sets* of states:
   - $M, Q \models \langle\langle a \rangle\rangle \phi$: there exists some strategy such that if a follows it *from any of the states in the set Q*, ϕ is guaranteed to be true
2. Introduce new *constructive knowledge* operators:
   - $M, q \models \mathbb{K}_a \phi \Leftrightarrow M, [q]_{\sim_a} \models \phi$

We get that:

$$M, q \models \mathbb{K}_a \langle\langle a \rangle\rangle \phi \Leftrightarrow M, [q]_{\sim_a} \models \langle\langle a \rangle\rangle \phi \Leftrightarrow$$

there exists some strategy such that if a follows it *from any of the states she considers possible*, ϕ is guaranteed to be true

Constructive Strategic Logic: key idea

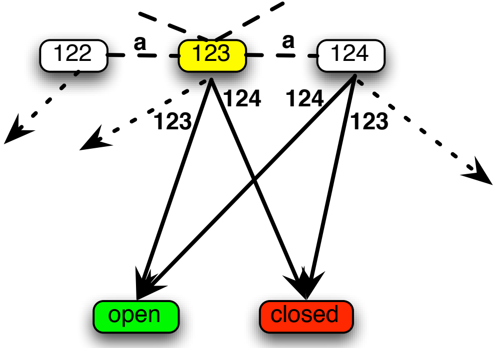
1. Interpret ability modalities in *sets* of states:
   - $M, Q \models \langle\langle a \rangle\rangle \phi$: there exists some strategy such that if a follows it *from any of the states in the set Q*, ϕ is guaranteed to be true
2. Introduce new *constructive knowledge* operators:
   - $M, q \models \mathbb{K}_a \phi \Leftrightarrow M, [q]_{\sim_a} \models \phi$

We get that:

$$M, q \models \mathbb{K}_a \langle\langle a \rangle\rangle \phi \Leftrightarrow M, [q]_{\sim_a} \models \langle\langle a \rangle\rangle \phi \Leftrightarrow$$

there exists some strategy such that if a follows it *from any of the states she considers possible*, ϕ is guaranteed to be true

Example

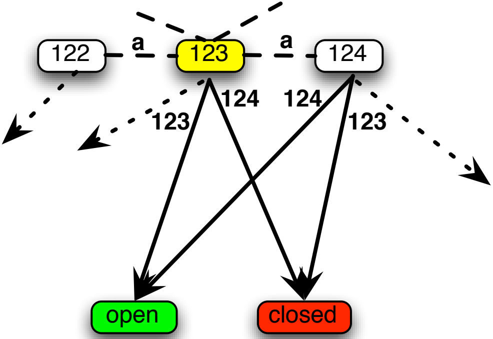


$$\langle\langle a\rangle\rangle \bigcirc open$$

$$K_a\langle\langle a\rangle\rangle \bigcirc open$$

$$\neg ck_a\langle\langle a\rangle\rangle \bigcirc open$$

Example



$$\langle\langle a \rangle\rangle \bigcirc \mathit{open}$$

$$K_a \langle\langle a \rangle\rangle \bigcirc \mathit{open}$$

$$\neg ck_a \langle\langle a \rangle\rangle \bigcirc \mathit{open}$$

Knowing how to Play

- Single agent case: we take into account the paths starting from indistinguishable states
- What about coalitions? In what sense should they know the strategy? Common knowledge (C_A), mutual knowledge (E_A), distributed knowledge (D_A)...?
- Other options also make sense!

Given strategy σ, agents A can have:

- Common knowledge that σ is a winning strategy. This requires the least amount of additional communication (agents from A may agree upon a total order over their collective strategies at the beginning of the game and that they will always choose the maximal winning strategy with respect to this order)
- Mutual knowledge that σ is a winning strategy: everybody in A knows that σ is winning

Given strategy σ, agents A can have:

- Common knowledge that σ is a winning strategy. This requires the least amount of additional communication (agents from A may agree upon a total order over their collective strategies at the beginning of the game and that they will always choose the maximal winning strategy with respect to this order)
- Mutual knowledge that σ is a winning strategy: everybody in A knows that σ is winning

- Distributed knowledge that σ is a winning strategy: if the agents share their knowledge at the current state, they can identify the strategy as winning
- "The leader": the strategy can be identified by agent $a \in A$
- "Headquarters' committee": the strategy can be identified by subgroup $A' \subseteq A$
- "Consulting company": the strategy can be identified by some other group B

- Distributed knowledge that σ is a winning strategy: if the agents share their knowledge at the current state, they can identify the strategy as winning
- "The leader": the strategy can be identified by agent $a \in A$
- "Headquarters' committee": the strategy can be identified by subgroup $A' \subseteq A$
- "Consulting company": the strategy can be identified by some other group B

- Distributed knowledge that σ is a winning strategy: if the agents share their knowledge at the current state, they can identify the strategy as winning
- "The leader": the strategy can be identified by agent $a \in A$
- "Headquarters' committee": the strategy can be identified by subgroup $A' \subseteq A$
- "Consulting company": the strategy can be identified by some other group B

- Distributed knowledge that σ is a winning strategy: if the agents share their knowledge at the current state, they can identify the strategy as winning
- "The leader": the strategy can be identified by agent $a \in A$
- "Headquarters' committee": the strategy can be identified by subgroup $A' \subseteq A$
- "Consulting company": the strategy can be identified by some other group B

Many subtle cases...

$\rightsquigarrow$ Solution: (general) constructive knowledge operators

Many subtle cases...

$\rightsquigarrow$ Solution: (general) constructive knowledge operators

Constructive Strategic Logic (CSL)

- $\langle\langle A\rangle\rangle\Phi$: A have a uniform memoryless strategy to enforce Φ
- $K_a\langle\langle a\rangle\rangle\Phi$: *a has a strategy to enforce Φ, and knows that he has one*
- For groups of agents: $C_A, E_A, D_A, \ldots$
- $\mathbb{K}_a\langle\langle a\rangle\rangle\Phi$: *a has a strategy to enforce Φ, and knows that this is a winning strategy*
- For groups of agents: $\mathbb{C}_A, \mathbb{E}_A, \mathbb{D}_A, \ldots$

Constructive Strategic Logic (CSL)

- $\langle\langle A\rangle\rangle\Phi$: A have a uniform memoryless strategy to enforce Φ
- $K_a\langle\langle a\rangle\rangle\Phi$: *a has a strategy to enforce Φ, and knows that he has one*
- For groups of agents: $C_A, E_A, D_A, \ldots$
- $\mathbb{K}_a\langle\langle a\rangle\rangle\Phi$: *a has a strategy to enforce Φ, and knows that this is a winning strategy*
- For groups of agents: $\mathbb{C}_A, \mathbb{E}_A, \mathbb{D}_A, \ldots$

Constructive Strategic Logic (CSL)

- $\langle\langle A\rangle\rangle\Phi$: *A* have a uniform memoryless strategy to enforce Φ
- $K_a\langle\langle a\rangle\rangle\Phi$: *a has a strategy to enforce Φ, and knows that he has one*
- For groups of agents: $C_A, E_A, D_A, \ldots$
- $\mathbb{K}_a\langle\langle a\rangle\rangle\Phi$: *a has a strategy to enforce Φ, and knows that this is a winning strategy*
- For groups of agents: $\mathbb{C}_A, \mathbb{E}_A, \mathbb{D}_A, \ldots$

Non-standard semantics:

- Formulae are evaluated in sets of states
- $M, Q \models \langle\langle A \rangle\rangle\gamma$: A have a single strategy to enforce γ from all states in Q

Additionally:

- $out(Q, s_A) = \bigcup_{q \in Q} out(q, s_A)$
- $\mathrm{img}(Q, \mathcal{R}) = \bigcup_{q \in Q} \mathrm{img}(q, \mathcal{R})$

- $M, q \models \varphi$ iff $M, \{q\} \models \varphi$

Non-standard semantics:

- Formulae are evaluated in sets of states
- $M, Q \models \langle\langle A \rangle\rangle \gamma$: A have a single strategy to enforce γ from all states in Q

Additionally:

- $out(Q, s_A) = \bigcup_{q \in Q} out(q, s_A)$
- $\mathrm{img}(Q, \mathcal{R}) = \bigcup_{q \in Q} \mathrm{img}(q, \mathcal{R})$

- $M, q \models \varphi$ iff $M, \{q\} \models \varphi$

Non-standard semantics:

- Formulae are evaluated in sets of states
- $M, Q \models \langle\langle A \rangle\rangle \gamma$: A have a single strategy to enforce γ from all states in Q

Additionally:

- $out(Q, s_A) = \bigcup_{q \in Q} out(q, s_A)$
- $\mathrm{img}(Q, \mathcal{R}) = \bigcup_{q \in Q} \mathrm{img}(q, \mathcal{R})$

- $M, q \models \varphi$ iff $M, \{q\} \models \varphi$

Definition (Semantics of CSL)

$M, Q \models p$ iff $p \in \pi(q)$ for every $q \in Q$;

$M, Q \models \neg\varphi$ iff not $M, Q \models \varphi$;

$M, Q \models \varphi \wedge \psi$ iff $M, Q \models \varphi$ and $M, Q \models \psi$;

$M, Q \models \langle\langle A \rangle\rangle \gamma$ iff there exists s_A such that, for every $\lambda \in out(Q, s_A)$, we have that $M, \lambda \models \gamma$;

$M, Q \models \mathcal{K}_A \varphi$ iff $M, q \models \varphi$ for every $q \in \mathrm{img}(Q, \sim_A^{\mathcal{K}})$ (where $\mathcal{K} = C, E, D$);

$M, Q \models \hat{\mathcal{K}}_A \varphi$ iff $M, \mathrm{img}(Q, \sim_A^{\mathcal{K}}) \models \varphi$ (where $\hat{\mathcal{K}} = \mathbb{C}, \mathbb{E}, \mathbb{D}$ and $\mathcal{K} = C, E, D$, respectively).

Definition (Semantics of CSL)

$M, Q \models p$ iff $p \in \pi(q)$ for every $q \in Q$;

$M, Q \models \neg\varphi$ iff not $M, Q \models \varphi$;

$M, Q \models \varphi \wedge \psi$ iff $M, Q \models \varphi$ and $M, Q \models \psi$;

$M, Q \models \langle\langle A \rangle\rangle \gamma$ iff there exists s_A such that, for every $\lambda \in out(Q, s_A)$, we have that $M, \lambda \models \gamma$;

$M, Q \models \mathcal{K}_A \varphi$ iff $M, q \models \varphi$ for every $q \in \mathrm{img}(Q, \sim_A^{\mathcal{K}})$ (where $\mathcal{K} = C, E, D$);

$M, Q \models \hat{\mathcal{K}}_A \varphi$ iff $M, \mathrm{img}(Q, \sim_A^{\mathcal{K}}) \models \varphi$ (where $\hat{\mathcal{K}} = \mathbb{C}, \mathbb{E}, \mathbb{D}$ and $\mathcal{K} = C, E, D$, respectively).

Definition (Semantics of CSL)

$M, Q \models p$ iff $p \in \pi(q)$ for every $q \in Q$;

$M, Q \models \neg\varphi$ iff not $M, Q \models \varphi$;

$M, Q \models \varphi \wedge \psi$ iff $M, Q \models \varphi$ and $M, Q \models \psi$;

$M, Q \models \langle\langle A \rangle\rangle \gamma$ iff there exists s_A such that, for every $\lambda \in out(Q, s_A)$, we have that $M, \lambda \models \gamma$;

$M, Q \models \mathcal{K}_A \varphi$ iff $M, q \models \varphi$ for every $q \in \mathrm{img}(Q, \sim_A^{\mathcal{K}})$ (where $\mathcal{K} = C, E, D$);

$M, Q \models \hat{\mathcal{K}}_A \varphi$ iff $M, \mathrm{img}(Q, \sim_A^{\mathcal{K}}) \models \varphi$ (where $\hat{\mathcal{K}} = \mathbb{C}, \mathbb{E}, \mathbb{D}$ and $\mathcal{K} = C, E, D$, respectively).

Definition (Semantics of CSL)

$M, Q \models p$ iff $p \in \pi(q)$ for every $q \in Q$;

$M, Q \models \neg\varphi$ iff not $M, Q \models \varphi$;

$M, Q \models \varphi \wedge \psi$ iff $M, Q \models \varphi$ and $M, Q \models \psi$;

$M, Q \models \langle\langle A \rangle\rangle\gamma$ iff there exists s_A such that, for every $\lambda \in out(Q, s_A)$, we have that $M, \lambda \models \gamma$;

$M, Q \models \mathcal{K}_A\varphi$ iff $M, q \models \varphi$ for every $q \in \mathrm{img}(Q, \sim_A^{\mathcal{K}})$ (where $\mathcal{K} = C, E, D$);

$M, Q \models \hat{\mathcal{K}}_A\varphi$ iff $M, \mathrm{img}(Q, \sim_A^{\mathcal{K}}) \models \varphi$ (where $\hat{\mathcal{K}} = \mathbb{C}, \mathbb{E}, \mathbb{D}$ and $\mathcal{K} = C, E, D$, respectively).

Definition (Semantics of CSL)

$M, Q \models p$ iff $p \in \pi(q)$ for every $q \in Q$;

$M, Q \models \neg\varphi$ iff not $M, Q \models \varphi$;

$M, Q \models \varphi \wedge \psi$ iff $M, Q \models \varphi$ and $M, Q \models \psi$;

$M, Q \models \langle\langle A \rangle\rangle \gamma$ iff there exists s_A such that, for every $\lambda \in out(Q, s_A)$, we have that $M, \lambda \models \gamma$;

$M, Q \models \mathcal{K}_A \varphi$ iff $M, q \models \varphi$ for every $q \in \mathrm{img}(Q, \sim_A^{\mathcal{K}})$ (where $\mathcal{K} = C, E, D$);

$M, Q \models \hat{\mathcal{K}}_A \varphi$ iff $M, \mathrm{img}(Q, \sim_A^{\mathcal{K}}) \models \varphi$ (where $\hat{\mathcal{K}} = \mathbb{C}, \mathbb{E}, \mathbb{D}$ and $\mathcal{K} = C, E, D$, respectively).

Definition (Semantics of CSL)

$M, Q \models p$ iff $p \in \pi(q)$ for every $q \in Q$;

$M, Q \models \neg\varphi$ iff not $M, Q \models \varphi$;

$M, Q \models \varphi \wedge \psi$ iff $M, Q \models \varphi$ and $M, Q \models \psi$;

$M, Q \models \langle\langle A \rangle\rangle \gamma$ iff there exists s_A such that, for every $\lambda \in out(Q, s_A)$, we have that $M, \lambda \models \gamma$;

$M, Q \models \mathcal{K}_A \varphi$ iff $M, q \models \varphi$ for every $q \in \mathrm{img}(Q, \sim_A^{\mathcal{K}})$ (where $\mathcal{K} = C, E, D$);

$M, Q \models \hat{\mathcal{K}}_A \varphi$ iff $M, \mathrm{img}(Q, \sim_A^{\mathcal{K}}) \models \varphi$ (where $\hat{\mathcal{K}} = \mathbb{C}, \mathbb{E}, \mathbb{D}$ and $\mathcal{K} = C, E, D$, respectively).

Validity in CSL

- Formula φ is valid iff $M, q \models \varphi$ for all models M and states q
- Formula φ is strongly valid iff for each M and every non-empty set of states Q it is the case that $M, Q \models \varphi$

Theorem

1. *Strong validity implies validity.*
2. *Validity does not imply strong validity.*

Validity in CSL

- Formula φ is valid iff $M, q \models \varphi$ for all models M and states q
- Formula φ is strongly valid iff for each M and every non-empty set of states Q it is the case that $M, Q \models \varphi$

Theorem

1. *Strong validity implies validity.*
2. *Validity does not imply strong validity.*

Validity in CSL

- We are ultimately interested in simple validity
- The importance of strong validity, on the other hand, lies in the fact that strong validity of $\varphi \leftrightarrow \psi$ makes φ and ψ completely interchangeable

Theorem

If $\varphi_1 \leftrightarrow \varphi_2$ is strongly valid, and ψ' is obtained from ψ through replacing an occurrence of φ_1 by φ_2, then $M, Q \models \psi$ iff $M, Q \models \psi'$.

Validity in CSL

- We are ultimately interested in simple validity
- The importance of strong validity, on the other hand, lies in the fact that strong validity of $\varphi \leftrightarrow \psi$ makes φ and ψ completely interchangeable

Theorem

If $\varphi_1 \leftrightarrow \varphi_2$ is strongly valid, and ψ' is obtained from ψ through replacing an occurrence of φ_1 by φ_2, then $M, Q \models \psi$ iff $M, Q \models \psi'$.

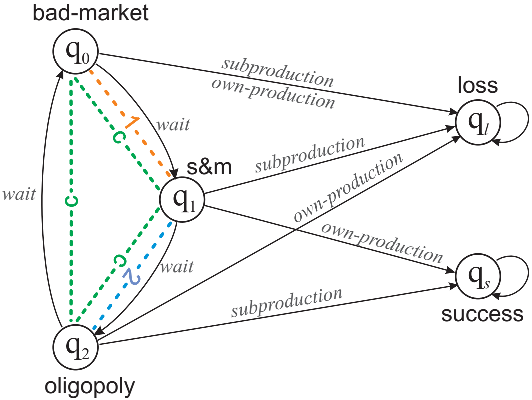
Validity in CSL

- We are ultimately interested in simple validity
- The importance of strong validity, on the other hand, lies in the fact that strong validity of $\varphi \leftrightarrow \psi$ makes φ and ψ completely interchangeable

Theorem

If $\varphi_1 \leftrightarrow \varphi_2$ is strongly valid, and ψ' is obtained from ψ through replacing an occurrence of φ_1 by φ_2, then $M, Q \models \psi$ iff $M, Q \models \psi'$.

Example: Simple Market



@ q_1 :

$\neg\mathbb{K}_c\langle\langle c\rangle\rangle\Diamond\text{success}$

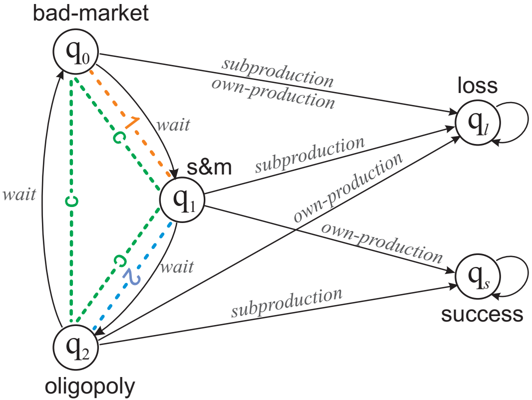
$\neg\mathbb{E}_{\{1,2\}}\langle\langle c\rangle\rangle\Diamond\text{success}$

$\neg\mathbb{K}_1\langle\langle c\rangle\rangle\Diamond\text{success}$

$\neg\mathbb{K}_2\langle\langle c\rangle\rangle\Diamond\text{success}$

$\mathbb{D}_{\{1,2\}}\langle\langle c\rangle\rangle\Diamond\text{success}$

Example: Simple Market



$@\ q_1$:

$\neg\mathbb{K}_c\langle\langle c\rangle\rangle\Diamond$success

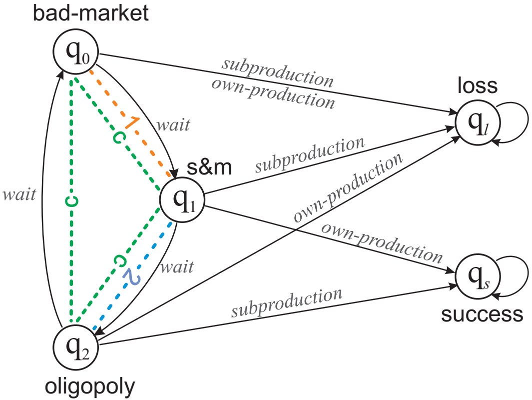
$\neg\mathbb{E}_{\{1,2\}}\langle\langle c\rangle\rangle\Diamond$success

$\neg\mathbb{K}_1\langle\langle c\rangle\rangle\Diamond$success

$\neg\mathbb{K}_2\langle\langle c\rangle\rangle\Diamond$success

$\mathbb{D}_{\{1,2\}}\langle\langle c\rangle\rangle\Diamond$success

Example: Simple Market



@ q_1 :

$\neg\mathbb{K}_c\langle\langle c\rangle\rangle\Diamond$success

$\neg\mathbb{E}_{\{1,2\}}\langle\langle c\rangle\rangle\Diamond$success

$\neg\mathbb{K}_1\langle\langle c\rangle\rangle\Diamond$success

$\neg\mathbb{K}_2\langle\langle c\rangle\rangle\Diamond$success

$\mathbb{D}_{\{1,2\}}\langle\langle c\rangle\rangle\Diamond$success

Onion Soup Robbery

A virtual safe contains the recipe for the best onion soup in the world. The safe can only be opened by a k-digit binary code, where each digit c_i is sent from a prescribed location i ($1 \leq i \leq k$). To open the safe and download the recipe it is enough that at least $n \leq k$ correct digits are sent at the same moment. However, if a wrong value is sent from one of the locations, or if an insufficient number (i.e., between 1 and $n - 1$) of digits is submitted, then the safe locks up and activates an alarm.

k agents are connected at the right locations; each of them can send 0, send 1, or do nothing (*nop*). Moreover, individual agents have only partial information about the code: agent i (connected to location i) knows the values of c_{i-1} XOR c_i and c_i XOR c_{i+1} (we take $c_0 = c_{k+1} = 0$). This implies that only agents 1 and k *know the values of "their" digits*. Still, *every agent knows whether his neighbors' digits are the same as his.*

Onion Soup Robbery: Some Properties

For OSR_k^n and the initial state, we have:

- $\neg\mathbb{E}_{\mathbb{A}\mathrm{gt}}\langle\langle\mathbb{A}\mathrm{gt}\rangle\rangle\Diamond\mathsf{open}$: the team cannot identify a winning strategy;
- $\mathbb{D}_{\mathbb{A}\mathrm{gt}}\langle\langle\mathbb{A}\mathrm{gt}\rangle\rangle\Diamond\mathsf{open}$: if the agents share information they can recognize who should send what;
- $\mathbb{D}_{\{1,\ldots,n-1\}}\langle\langle\mathbb{A}\mathrm{gt}\rangle\rangle\Diamond\mathsf{open}$: it is enough that the first $n-1$ agents devise the strategy. Note that the same holds for the last $n-1$ agents, i.e., the subteam $\{k-n+2,\ldots,k\}$.

Onion Soup Robbery: Some Properties

For OSR_k^n and the initial state, we have:

- $\neg\mathbb{E}_{\mathbb{A}\mathrm{gt}}\langle\langle\mathbb{A}\mathrm{gt}\rangle\rangle\Diamond\mathrm{open}$: the team cannot identify a winning strategy;
- $\mathbb{D}_{\mathbb{A}\mathrm{gt}}\langle\langle\mathbb{A}\mathrm{gt}\rangle\rangle\Diamond\mathrm{open}$: if the agents share information they can recognize who should send what;
- $\mathbb{D}_{\{1,\ldots,n-1\}}\langle\langle\mathbb{A}\mathrm{gt}\rangle\rangle\Diamond\mathrm{open}$: it is enough that the first $n-1$ agents devise the strategy. Note that the same holds for the last $n-1$ agents, i.e., the subteam $\{k-n+2,\ldots,k\}$.

Onion Soup Robbery: Some Properties

For OSR_k^n and the initial state, we have:

- $\neg\mathbb{E}_{\mathbb{A}\mathrm{gt}}\langle\langle\mathbb{A}\mathrm{gt}\rangle\rangle\Diamond$open: the team cannot identify a winning strategy;
- $\mathbb{D}_{\mathbb{A}\mathrm{gt}}\langle\langle\mathbb{A}\mathrm{gt}\rangle\rangle\Diamond$open: if the agents share information they can recognize who should send what;
- $\mathbb{D}_{\{1,\ldots,n-1\}}\langle\langle\mathbb{A}\mathrm{gt}\rangle\rangle\Diamond$open: it is enough that the first $n-1$ agents devise the strategy. Note that the same holds for the last $n-1$ agents, i.e., the subteam $\{k-n+2,\ldots,k\}$.

Properties of Constructive Knowledge

Non-standard semantics raises some natural questions:

- Is constructive knowledge... em, well, knowledge?

  $\leadsto$ semantic vs. syntactic analysis
- Is constructive knowledge a special kind of standard knowledge? Or the other way around?
- Is there a relevant subset of the language for whom a more standard semantics can be given?

Properties of Constructive Knowledge

Non-standard semantics raises some natural questions:

- Is constructive knowledge... em, well, knowledge?
  $\rightsquigarrow$ semantic vs. syntactic analysis
- Is constructive knowledge a special kind of standard knowledge? Or the other way around?
- Is there a relevant subset of the language for whom a more standard semantics can be given?

Properties of Constructive Knowledge

Non-standard semantics raises some natural questions:

- Is constructive knowledge... em, well, knowledge?
  $\rightsquigarrow$ semantic vs. syntactic analysis
- Is constructive knowledge a special kind of standard knowledge? Or the other way around?
- Is there a relevant subset of the language for whom a more standard semantics can be given?

Properties of Constructive Knowledge

Non-standard semantics raises some natural questions:

- Is constructive knowledge... em, well, knowledge?
  $\rightsquigarrow$ semantic vs. syntactic analysis
- Is constructive knowledge a special kind of standard knowledge? Or the other way around?
- Is there a relevant subset of the language for whom a more standard semantics can be given?

Is $\mathbb{K}_a$ an Epistemic Operator?

Theorem

Below, we list the constructive knowledge versions of some of the S5 properties for individual agents. "Yes" means that the schema is strongly valid; "No" means that it is not even weakly valid (incidentally, none of the properties turns out to be weakly but not strongly valid).

K	$\mathbb{K}_a(\varphi \to \psi) \to (\mathbb{K}_a\varphi \to \mathbb{K}_a\psi)$	*Yes*
D	$\neg\mathbb{K}_a\bot$	*Yes*
T	$\mathbb{K}_a\varphi \to \varphi$	*No*
4	$\mathbb{K}_a\varphi \to \mathbb{K}_a\mathbb{K}_a\varphi$	*Yes*
4$^+$	$\mathbb{K}_a\varphi \leftrightarrow \mathbb{K}_a\mathbb{K}_a\varphi$	*Yes*
5	$\neg\mathbb{K}_a\varphi \to \mathbb{K}_a\neg\mathbb{K}_a\varphi$	*Yes*
5$^+$	$\neg\mathbb{K}_a\varphi \leftrightarrow \mathbb{K}_a\neg\mathbb{K}_a\varphi$	*Yes*
B	$\varphi \to \mathbb{K}_a\neg\mathbb{K}_a\neg\varphi$	*No*

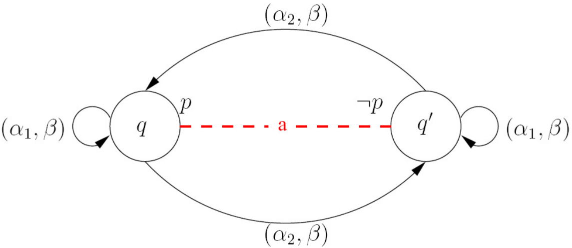
Is $\mathbb{K}_a$ an Epistemic Operator?

Theorem

Below, we list the constructive knowledge versions of some of the S5 properties for individual agents. "Yes" means that the schema is strongly valid; "No" means that it is not even weakly valid (incidentally, none of the properties turns out to be weakly but not strongly valid).

K	$\mathbb{K}_a(\varphi \to \psi) \to (\mathbb{K}_a\varphi \to \mathbb{K}_a\psi)$	*Yes*
D	$\neg\mathbb{K}_a\bot$	*Yes*
T	$\mathbb{K}_a\varphi \to \varphi$	*No*
4	$\mathbb{K}_a\varphi \to \mathbb{K}_a\mathbb{K}_a\varphi$	*Yes*
4$^+$	$\mathbb{K}_a\varphi \leftrightarrow \mathbb{K}_a\mathbb{K}_a\varphi$	*Yes*
5	$\neg\mathbb{K}_a\varphi \to \mathbb{K}_a\neg\mathbb{K}_a\varphi$	*Yes*
5$^+$	$\neg\mathbb{K}_a\varphi \leftrightarrow \mathbb{K}_a\neg\mathbb{K}_a\varphi$	*Yes*
B	$\varphi \to \mathbb{K}_a\neg\mathbb{K}_a\neg\varphi$	*No*

Invalidity of Axiom T



Let M be as above
Now, $M, q \models \mathbb{K}_a \neg p$, but $M, q \not\models \neg p$

In Quest for the Truth Axiom

- $\mathbb{K}_a$ is not S5: axioms **K**, **D**, **4**, **5** hold, but **T** does not
- However, if we slightly restrict the language, then the corresponding **T** axiom becomes strongly valid

- Let CSL$^-$ be the subset of CSL in which, between every occurrence of constructive knowledge ($\mathbb{C}_A, \mathbb{E}_A, \mathbb{D}_A$) and negation, there is always at least one operator other than conjunction
- In particular, the requirement is met when $\mathbb{C}_A, \mathbb{E}_A, \mathbb{D}_A$ are never immediately followed by $\neg$ or $\wedge$

Theorem

Every CSL$^-$ instance of **T** *(i.e., $\mathbb{K}_a\psi \to \psi$) is strongly valid.*

In Quest for the Truth Axiom

- $\mathbb{K}_a$ is not S5: axioms **K**, **D**, **4**, **5** hold, but **T** does not
- However, if we slightly restrict the language, then the corresponding **T** axiom becomes strongly valid

- Let CSL$^-$ be the subset of CSL in which, between every occurrence of constructive knowledge ($\mathbb{C}_A, \mathbb{E}_A, \mathbb{D}_A$) and negation, there is always at least one operator other than conjunction
- In particular, the requirement is met when $\mathbb{C}_A, \mathbb{E}_A, \mathbb{D}_A$ are never immediately followed by $\neg$ or $\wedge$

Theorem

Every CSL$^-$ instance of **T** *(i.e., $\mathbb{K}_a\psi \rightarrow \psi$) is strongly valid.*

In Quest for the Truth Axiom

- $\mathbb{K}_a$ is not S5: axioms **K**, **D**, **4**, **5** hold, but **T** does not
- However, if we slightly restrict the language, then the corresponding **T** axiom becomes strongly valid

- Let CSL$^-$ be the subset of CSL in which, between every occurrence of constructive knowledge ($\mathbb{C}_A, \mathbb{E}_A, \mathbb{D}_A$) and negation, there is always at least one operator other than conjunction
- In particular, the requirement is met when $\mathbb{C}_A, \mathbb{E}_A, \mathbb{D}_A$ are never immediately followed by $\neg$ or $\wedge$

Theorem

Every CSL$^-$ instance of **T** *(i.e., $\mathbb{K}_a\psi \to \psi$) is strongly valid.*

In Quest for the Truth Axiom

Is then the constructive knowledge in CSL$^-$ S5?

Not really

- The extension of schema **T** is different in CSL and CSL$^-$
- More importantly, in CSL$^-$ schemata **K** and **5** are not valid, but they are not invalid either – they are simply *not formulae at all*
- Finally, CSL$^-$ lacks the S5 principle of uniform substitution

In Quest for the Truth Axiom

Is then the constructive knowledge in CSL$^-$ S5?

Not really

- The extension of schema **T** is different in CSL and CSL$^-$
- More importantly, in CSL$^-$ schemata **K** and **5** are not valid, but they are not invalid either – they are simply *not formulae at all*
- Finally, CSL$^-$ lacks the S5 principle of uniform substitution

In Quest for the Truth Axiom

Is then the constructive knowledge in CSL^- S5?
Not really

- The extension of schema **T** is different in CSL and CSL^-
- More importantly, in CSL^- schemata **K** and **5** are not valid, but they are not invalid either – they are simply *not formulae at all*
- Finally, CSL^- lacks the S5 principle of uniform substitution

Properties of Collective Constructive Knowledge

Theorem

Below, we list some of the S5 properties for collective constructive knowledge operators. "Yes" means that the schema is strongly valid; "No" means that it is not even weakly valid.

	$\mathbb{C}_A$	$\mathbb{E}_A$	$\mathbb{D}_A$
K	*Yes*	*Yes*	*Yes*
D	*Yes*	*Yes*	*Yes*
T	*No*	*No*	*No*
4	*Yes*	*No*	*Yes*
$\mathbf{4}^+$	*Yes*	*No*	*Yes*
5	*Yes*	*No*	*Yes*
$\mathbf{5}^+$	*Yes*	*No*	*Yes*
B	*No*	*No*	*No*

Properties of Collective Constructive Knowledge

Theorem

Every CSL$^-$ instance of schema **T** *for collective constructive knowledge operators* $\mathbb{C}_A, \mathbb{E}_A, \mathbb{D}_A$ *is strongly valid.*

Normal Form and State-Based Semantics

Constructive Normal Form

A CSL formula is in *constructive normal form (CSNF)* if every subformula starting with a $\hat{\mathcal{K}}_A$ operator is of the form $\hat{\mathcal{K}}_{A_1} \ldots \hat{\mathcal{K}}_{A_n} \psi$ where ψ starts with a cooperation modality.

Proposition

Every CSL formula is strongly equivalent to a formula in constructive normal form.

Normal Form and State-Based Semantics

Constructive Normal Form

A CSL formula is in *constructive normal form (CSNF)* if every subformula starting with a $\hat{\mathcal{K}}_A$ operator is of the form $\hat{\mathcal{K}}_{A_1} \ldots \hat{\mathcal{K}}_{A_n}\psi$ where ψ starts with a cooperation modality.

Proposition

Every CSL formula is strongly equivalent to a formula in constructive normal form.

Normal Form CSL

Observation

The "normal form CSL" can be given semantics entirely in terms of models and states.

$M, q \models \hat{\mathcal{K}}^1_{A_1} \ldots \hat{\mathcal{K}}^n_{A_n} \langle\langle A \rangle\rangle \gamma$ iff there exists S_A such that, for every $\lambda \in out(\mathrm{img}(q, rel(\hat{\mathcal{K}}^1_{A_1} \ldots \hat{\mathcal{K}}^n_{A_n}), S_A)$, we have that $M, \lambda \models \gamma$,

where $rel(\hat{\mathcal{K}}^1_{A_1} \ldots \hat{\mathcal{K}}^n_{A_n}) = \sim^{\mathcal{K}^1}_{A_1} \circ \cdots \circ \sim^{\mathcal{K}^n}_{A_n}$.

Normal Form CSL

Observation

The “normal form CSL” can be given semantics entirely in terms of models and states.

$M, q \models \hat{\mathcal{K}}^1_{A_1} \dots \hat{\mathcal{K}}^n_{A_n} \langle\!\langle A \rangle\!\rangle \gamma$ iff there exists S_A such that, for every $\lambda \in out(\mathrm{img}(q, rel(\hat{\mathcal{K}}^1_{A_1} \dots \hat{\mathcal{K}}^n_{A_n}), S_A)$, we have that $M, \lambda \models \gamma$,

where $rel(\hat{\mathcal{K}}^1_{A_1} \dots \hat{\mathcal{K}}^n_{A_n}) = \sim^{\mathcal{K}^1}_{A_1} \circ \dots \circ \sim^{\mathcal{K}^n}_{A_n}$.

Normal Form CSL vs. Onion Soup

- $\neg\mathbb{E}_{\mathbb{A}\mathrm{gt}}\langle\langle\mathbb{A}\mathrm{gt}\rangle\rangle\Diamond$open
- $\mathbb{D}_{\mathbb{A}\mathrm{gt}}\langle\langle\mathbb{A}\mathrm{gt}\rangle\rangle\Diamond$open
- $\mathbb{D}_{\{1,\ldots,n-1\}}\langle\langle\mathbb{A}\mathrm{gt}\rangle\rangle\Diamond$open

These are normal form formulae!

Strategies for Different Settings

Four variants of ability: IR, Ir, iR, ir (Schobbens 2004)

- I/i: perfect/imperfect information
- R/r: perfect/imperfect recall

- r: $s_a : St \to Act$ (memoryless strategies)
- R: $s_a : St^+ \to Act$ (perfect recall strategies)

- i: only uniform strategies,
- I: no restrictions

- r: s_a is uniform iff $q \sim_a q' \Rightarrow s_a(q) = s_a(q')$
- R: s_a is uniform iff $\lambda \approx_a \lambda' \Rightarrow s_a(\lambda) = s_a(\lambda')$
- $\lambda \approx_a \lambda'$ iff $\forall_i \lambda[i] \sim_a \lambda'[i]$

Strategies for Different Settings

Four variants of ability: IR, Ir, iR, ir (Schobbens 2004)

- I/i: perfect/imperfect information
- R/r: perfect/imperfect recall

- r: $s_a : St \to Act$ (memoryless strategies)
- R: $s_a : St^+ \to Act$ (perfect recall strategies)

- i: only uniform strategies,
- I: no restrictions

- r: s_a is uniform iff $q \sim_a q' \Rightarrow s_a(q) = s_a(q')$
- R: s_a is uniform iff $\lambda \approx_a \lambda' \Rightarrow s_a(\lambda) = s_a(\lambda')$
- $\lambda \approx_a \lambda'$ iff $\forall_i \lambda[i] \sim_a \lambda'[i]$

Strategies for Different Settings

Four variants of ability: IR, Ir, iR, ir (Schobbens 2004)

- I/i: perfect/imperfect information
- R/r: perfect/imperfect recall

- r: $s_a : St \to Act$ (memoryless strategies)
- R: $s_a : St^+ \to Act$ (perfect recall strategies)

- i: only uniform strategies,
- I: no restrictions

- r: s_a is uniform iff $q \sim_a q' \Rightarrow s_a(q) = s_a(q')$
- R: s_a is uniform iff $\lambda \approx_a \lambda' \Rightarrow s_a(\lambda) = s_a(\lambda')$
- $\lambda \approx_a \lambda'$ iff $\forall_i \lambda[i] \sim_a \lambda'[i]$

Strategies for Different Settings

Four variants of ability: IR, Ir, iR, ir (Schobbens 2004)

- I/i: perfect/imperfect information
- R/r: perfect/imperfect recall

- r: $s_a : St \rightarrow Act$ (memoryless strategies)
- R: $s_a : St^+ \rightarrow Act$ (perfect recall strategies)

- i: only uniform strategies,
- I: no restrictions

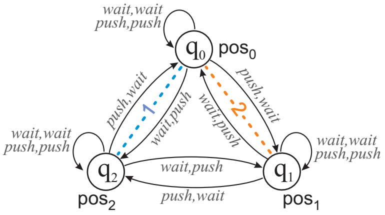
- r: s_a is uniform iff $q \sim_a q' \Rightarrow s_a(q) = s_a(q')$
- R: s_a is uniform iff $\lambda \approx_a \lambda' \Rightarrow s_a(\lambda) = s_a(\lambda')$
- $\lambda \approx_a \lambda'$ iff $\forall_i \lambda[i] \sim_a \lambda'[i]$

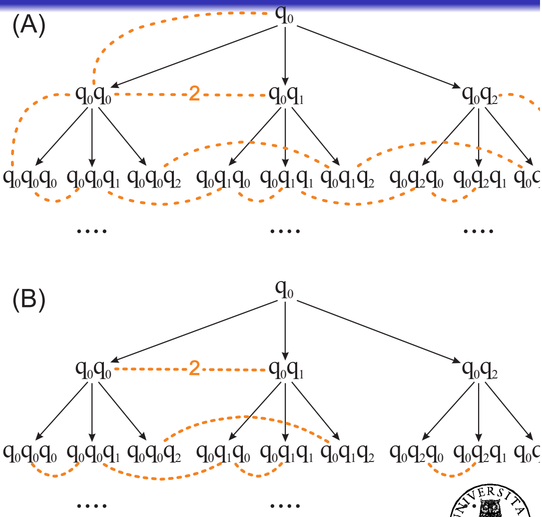
Model Checking Complexity

logic	*ir*	*iR*	*Ir*	*IR*
$\langle\langle\Gamma\rangle\rangle - ATL$	NP	U [11]	nl [2]	nl [2]
ATL	$\Delta_2 P$	U [11]	nl [2]	nl [2]
ATL^+	$\Delta_3 P$	U [11]	$\Delta_3 P$	$\Delta_3 P$
ATL^*	$PSPACE$	U [11]	$PSPACE$	$DEXP$ [9]

NP	complete for nondeterministic polynomial time
$\Delta_2 P = P^{NP}$	complete for polynomial calls to an NP oracle
$\Delta_3 P = P^{NP^{NP}}$	complete for polynomial calls to a $\Sigma_2 P$ oracle
EXP	complete for deterministic exponential time
DEXP	complete for deterministic doubly exponential time
U	undecidable
l	size of the formula
n	size of the model

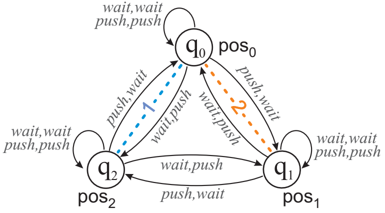
Perfect vs. Imperfect Recall



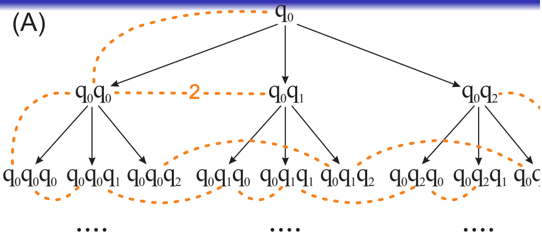


Advice: the restrictions on strategies and the semantics of epistemic operators should match!

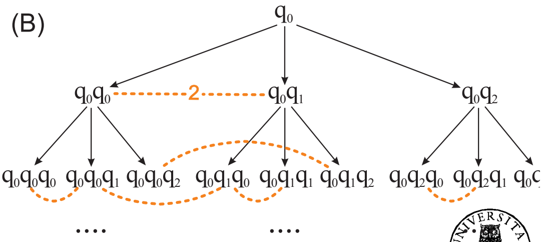
Perfect vs. Imperfect Recall



(A)

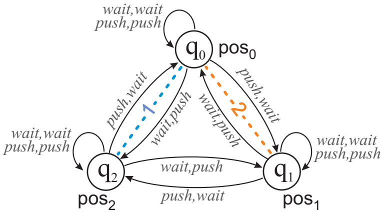


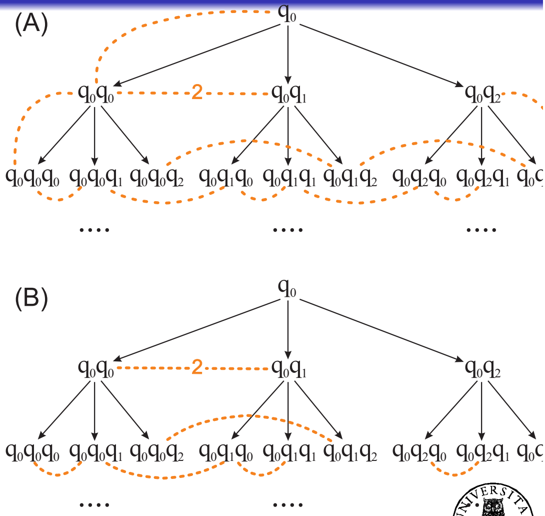
(B)



Advice: the restrictions on strategies and the semantics of epistemic operators should match!

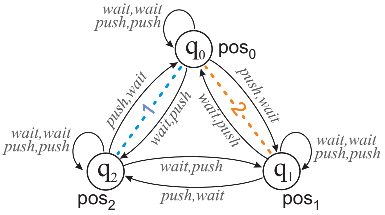
Perfect vs. Imperfect Recall



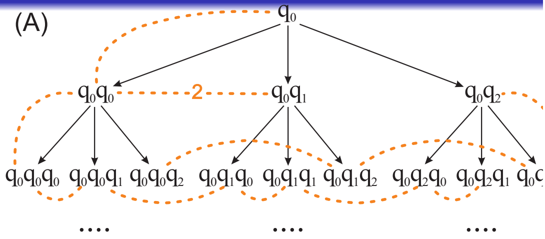


Advice: the restrictions on strategies and the semantics of epistemic operators should match!

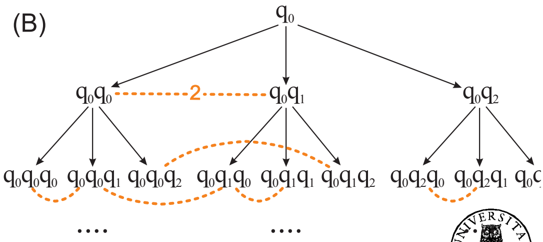
Perfect vs. Imperfect Recall



(A)



(B)



Advice: the restrictions on strategies and the semantics of epistemic operators should match!

Contents

1 ATL

2 Strategic Reasoning under Imperfect Information

3 Group Announcement Logic

Elevator pitch

Group Announcement Logic extends public announcement logic with:

$$\langle G \rangle \phi : \text{"Group } G \text{ can make an announcement after which } \phi \text{ is true"}$$

Adding quantification: APAL

$$M, s \models \langle \phi_1 \rangle \phi_2 \Leftrightarrow M, s \models \phi_1 \text{ and } M|\phi_1, s \models \phi_2$$

Idea (van Benthem, Analysis, 2004): interpret the modal diamond as "there is an announcement such that.."

Arbitrary announcement logic (APAL) (Balbiani et al., TARK 2007):

$$\varphi ::= p \mid K_i\varphi \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \langle \varphi_1 \rangle \varphi_2 \mid \Diamond\phi$$

$$M, s \models \Diamond\phi \Leftrightarrow \exists\psi \; M, s \models \langle \psi \rangle \phi$$

Quantification in APAL

$$M, s \models \Diamond\phi \Leftrightarrow \exists\psi \; M, s \models \langle\psi\rangle\phi$$

Note: the quantification includes announcements that cannot be truthfully made in the system

Quantification: announcements by an agent

$$K_i\psi$$

Quantification: announcements by an agent

$$M, s \models \langle i \rangle \phi \Leftrightarrow \exists \psi \; M, s \models \langle K_i \psi \rangle \phi$$

Quantification: announcements by a group

$$M, s \models \langle G \rangle \phi \quad \Leftrightarrow \quad \exists \{\psi_i : i \in G\}\ M, s \models \langle \bigwedge_{i \in G} K_i \psi_i \rangle \phi$$

Group Announcement Logic (GAL):

$$\varphi ::= p \mid K_i \varphi \mid \neg \varphi \mid \varphi_1 \wedge \varphi_2 \mid \langle \varphi_1 \rangle \varphi_2 \mid \langle G \rangle \phi$$

H. van Ditmarsch, *The Russian Cards Problem*, Studia Logica **75**, 2003

Example: **The Russian Cards Problem**

From a pack of seven known cards 0,1,2,3,4,5,6 Anne and Bill each draw three cards and Cath gets the remaining card. How can Anne and Bill openly inform each other about their cards, without Cath learning who holds any of their cards?

Example: **The Russian Cards Problem**

From a pack of seven known cards 0,1,2,3,4,5,6 Anne and Bill each draw three cards and Cath gets the remaining card. How can Anne and Bill openly inform each other about their cards, without Cath learning who holds any of their cards?

Formalisation: 012_a : "Ann has cards 0,1 and 2"

$$(one)\ \bigwedge_{ijk}(ijk_b \rightarrow K_a ijk_b) \quad (two)\ \bigwedge_{ijk}(ijk_a \rightarrow K_b ijk_a)$$

$$(three)\ \bigwedge_{q=0}^{6}((q_a \rightarrow \neg K_c q_a) \wedge (q_b \rightarrow \neg K_c q_b))$$

Example: **The Russian Cards Problem**

From a pack of seven known cards 0,1,2,3,4,5,6 Anne and Bill each draw three cards and Cath gets the remaining card. How can Anne and Bill openly inform each other about their cards, without Cath learning who holds any of their cards?

Formalisation: 012_a : "Ann has cards 0,1 and 2"

$$(\mathit{one})\ \bigwedge_{ijk}(ijk_b \rightarrow K_a ijk_b) \quad (\mathit{two})\ \bigwedge_{ijk}(ijk_a \rightarrow K_b ijk_a)$$

$$(\mathit{three})\ \bigwedge_{q=0}^{6}((q_a \rightarrow \neg K_c q_a) \wedge (q_b \rightarrow \neg K_c q_b))$$

Known solution:

$$\mathit{anne} \equiv 012_a \vee 034_a \vee 056_a \vee 135_a \vee 246_a$$

$$\mathit{bill} \equiv 345_b \vee 125_b \vee 024_b$$

Example: **The Russian Cards Problem**

From a pack of seven known cards 0,1,2,3,4,5,6 Anne and Bill each draw three cards and Cath gets the remaining card. How can Anne and Bill openly inform each other about their cards, without Cath learning who holds any of their cards?

Formalisation: 012_a : "Ann has cards 0,1 and 2"

$$(one) \ \bigwedge_{ijk}(ijk_b \rightarrow K_a ijk_b) \quad (two) \ \bigwedge_{ijk}(ijk_a \rightarrow K_b ijk_a)$$

$$(three) \ \bigwedge_{q=0}^{6}((q_a \rightarrow \neg K_c q_a) \wedge (q_b \rightarrow \neg K_c q_b))$$

Known solution:

$$anne \equiv 012_a \vee 034_a \vee 056_a \vee 135_a \vee 246_a$$

$$bill \equiv 345_b \vee 125_b \vee 024_b$$

PAL: $\langle K_a anne \rangle \langle K_b bill \rangle (one \wedge two \wedge three)$

H. van Ditmarsch, *The Russian Cards Problem*, Studia Logica **75**, 2003

Example: **The Russian Cards Problem**

From a pack of seven known cards 0,1,2,3,4,5,6 Anne and Bill each draw three cards and Cath gets the remaining card. How can Anne and Bill openly inform each other about their cards, without Cath learning who holds any of their cards?

Formalisation: 012_a : "Ann has cards 0,1 and 2"

$$(\mathit{one})\ \bigwedge_{ijk}(ijk_b \rightarrow K_a ijk_b) \quad (\mathit{two})\ \bigwedge_{ijk}(ijk_a \rightarrow K_b ijk_a)$$

$$(\mathit{three})\ \bigwedge_{q=0}^{6}((q_a \rightarrow \neg K_c q_a) \wedge (q_b \rightarrow \neg K_c q_b))$$

Known solution: $\mathit{anne} \equiv 012_a \vee 034_a \vee 056_a \vee 135_a \vee 246_a$

$\mathit{bill} \equiv 345_b \vee 125_b \vee 024_b$

PAL: $\langle K_a \mathit{anne}\rangle\langle K_b \mathit{bill}\rangle(\mathit{one} \wedge \mathit{two} \wedge \mathit{three})$

GAL: $\langle a\rangle\langle b\rangle(\mathit{one} \wedge \mathit{two} \wedge \mathit{three})$

Quantification: sequences of announcements

$$\langle G \rangle \langle G \rangle \phi \rightarrow \langle G \rangle \phi?$$

Quantification: sequences of announcements

$$\langle G \rangle \langle G \rangle \phi \to \langle G \rangle \phi ?$$

Answer: Yes.

Quantification: sequences of announcements

$$\langle G \rangle \langle G \rangle \phi \rightarrow \langle G \rangle \phi$$

$$M, s \models \langle G \rangle \phi \Leftrightarrow \text{ there is an announcement by } G\text{, after which } \phi$$

Quantification: sequences of announcements

$$\langle G\rangle\langle G\rangle\phi \rightarrow \langle G\rangle\phi$$

$M, s \models \langle G\rangle\phi \Leftrightarrow$ there is an announcement by G, after which ϕ

$\Leftrightarrow$ there is a sequence of announcements by G, after which ϕ

Example: **Russian Cards** (ctnd.)

$$\langle K_a anne \rangle \langle K_b bill \rangle (one \wedge two \wedge three)$$

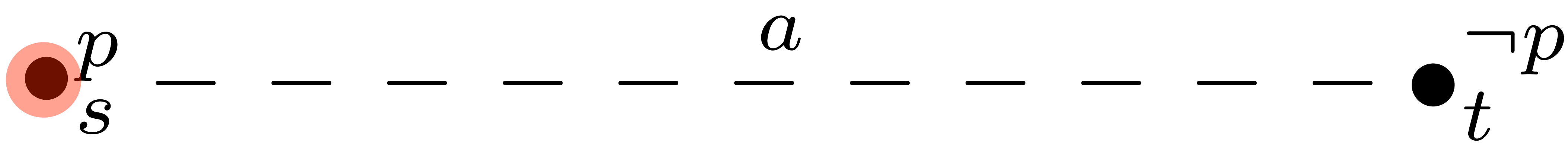
$$\langle a \rangle \langle b \rangle (one \wedge two \wedge three)$$

$$\langle ab \rangle (one \wedge two \wedge three)$$

Knowledge and ability in GAL

- Recall:
  - the de dicto/de re distinction
  - knowledge of ability de re cannot be expressed in general
- In GAL, knowledge and action are intimately connected
- How do the previous observations apply to GAL?

Being able to without knowing it



$$s \models \langle a \rangle p \wedge \neg K_a \langle a \rangle p$$

Example: **Russian Cards** (ctnd.)

$$\langle K_a anne\rangle\langle K_b bill\rangle(one \land two \land three)$$

$$\langle a\rangle\langle b\rangle(one \land two \land three)$$

$$\langle ab\rangle(one \land two \land three)$$

$$K_a\langle ab\rangle(one \land two \land three)$$

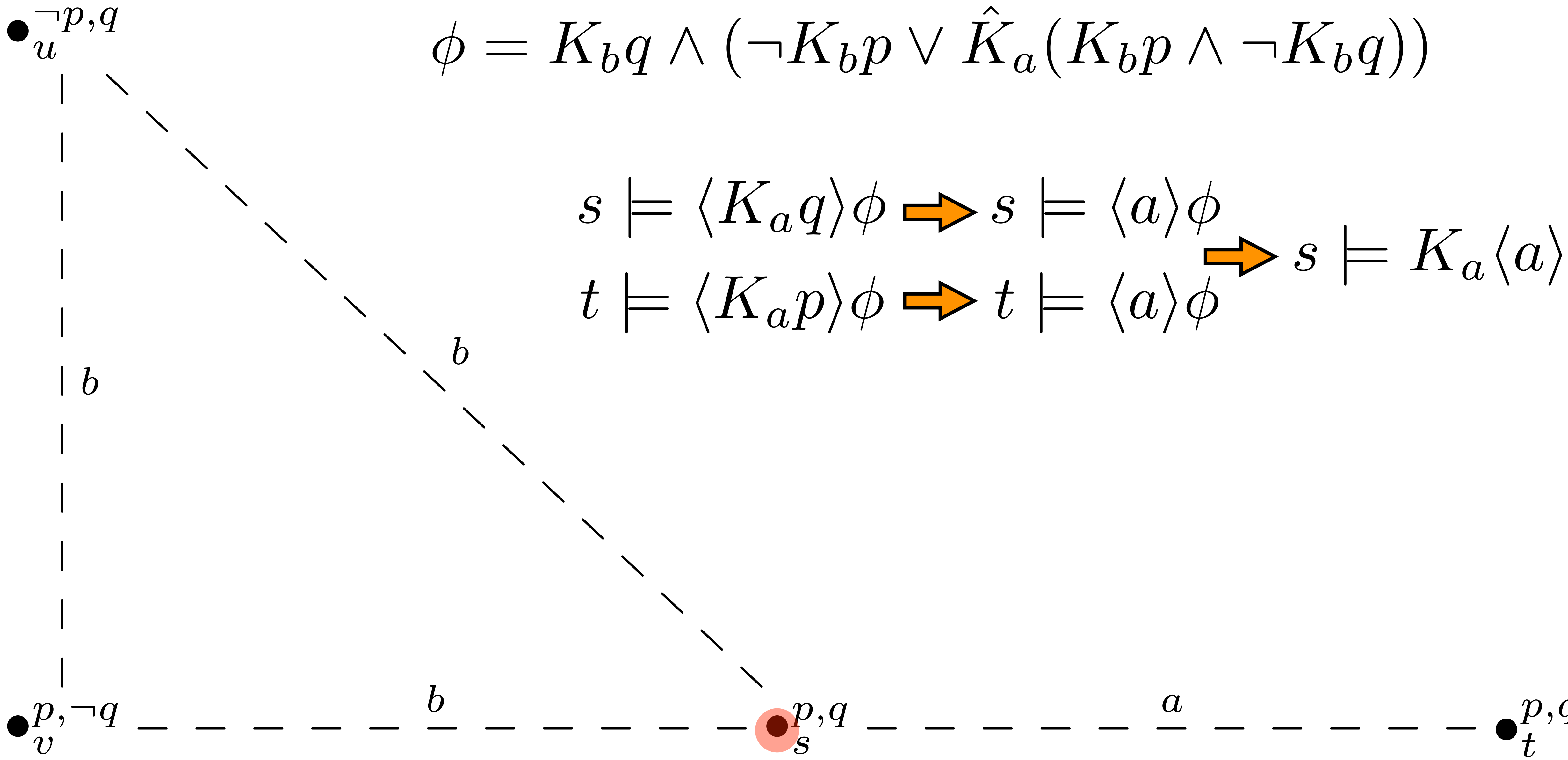
Being able to, knowing that, but not knowing how

$$\phi = K_b q \wedge (\neg K_b p \vee \hat{K}_a (K_b p \wedge \neg K_b q))$$

$$s \models \langle K_a q \rangle \phi \Rightarrow s \models \langle a \rangle \phi$$

$$t \models \langle K_a p \rangle \phi \Rightarrow t \models \langle a \rangle \phi$$

$$\Rightarrow s \models K_a \langle a \rangle \phi$$



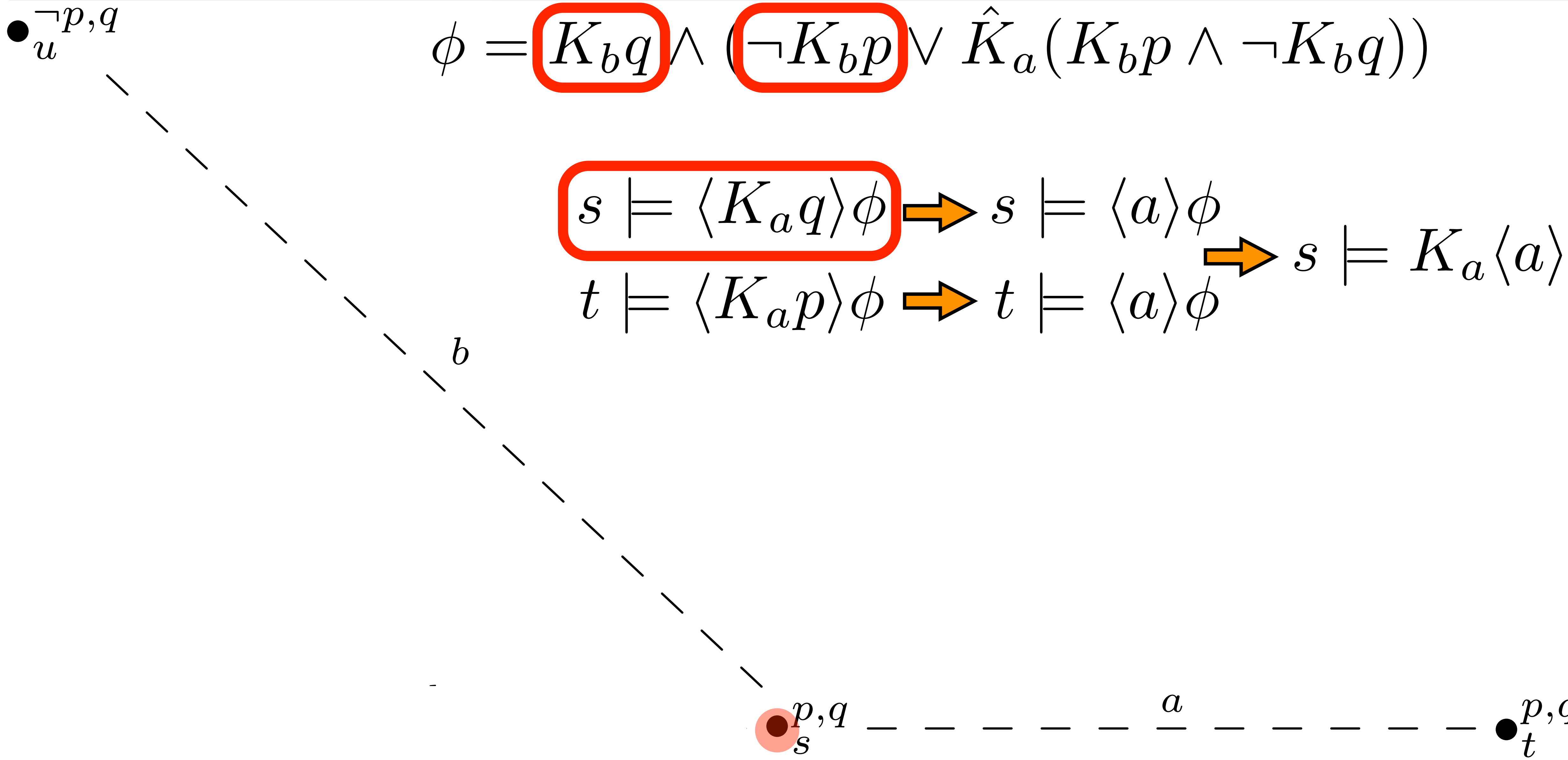
Being able to, knowing that, but not knowing how

$$\phi = \boxed{K_b q} \wedge (\boxed{\neg K_b p} \vee \hat{K}_a(K_b p \wedge \neg K_b q))$$

$$\boxed{s \models \langle K_a q\rangle\phi} \Rightarrow s \models \langle a\rangle\phi$$

$$t \models \langle K_a p\rangle\phi \Rightarrow t \models \langle a\rangle\phi$$

$$\Rightarrow s \models K_a\langle a\rangle\phi$$



$$s \models \phi$$

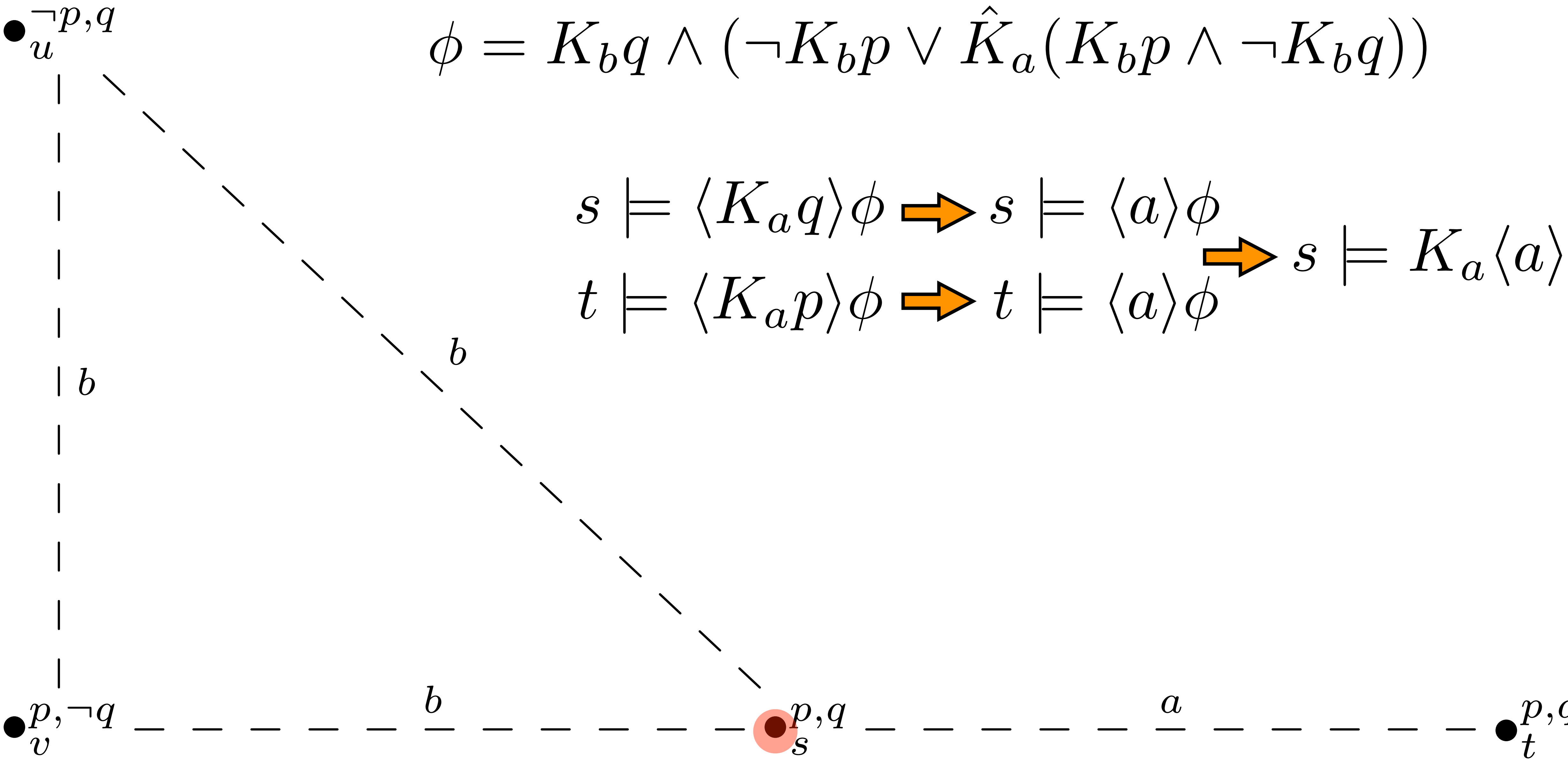
Being able to, knowing that, but not knowing how

$$\phi = K_b q \wedge (\neg K_b p \vee \hat{K}_a(K_b p \wedge \neg K_b q))$$

$$s \models \langle K_a q\rangle\phi \Rightarrow s \models \langle a\rangle\phi$$

$$t \models \langle K_a p\rangle\phi \Rightarrow t \models \langle a\rangle\phi$$

$$\Rightarrow s \models K_a\langle a\rangle\phi$$



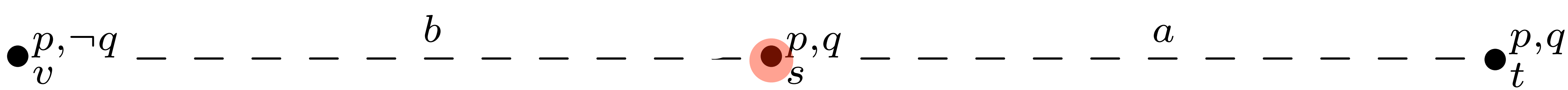
Being able to, knowing that, but not knowing how

$$\phi = \boxed{K_b q} \wedge (\neg K_b p \vee \boxed{\hat{K}_a(K_b p \wedge \neg K_b q)})$$

$$s \models \langle K_a q \rangle \phi \Rightarrow s \models \langle a \rangle \phi$$

$$\boxed{t \models \langle K_a p \rangle \phi} \Rightarrow t \models \langle a \rangle \phi$$

$$\Rightarrow s \models K_a \langle a \rangle \phi$$



$$t \models \phi$$

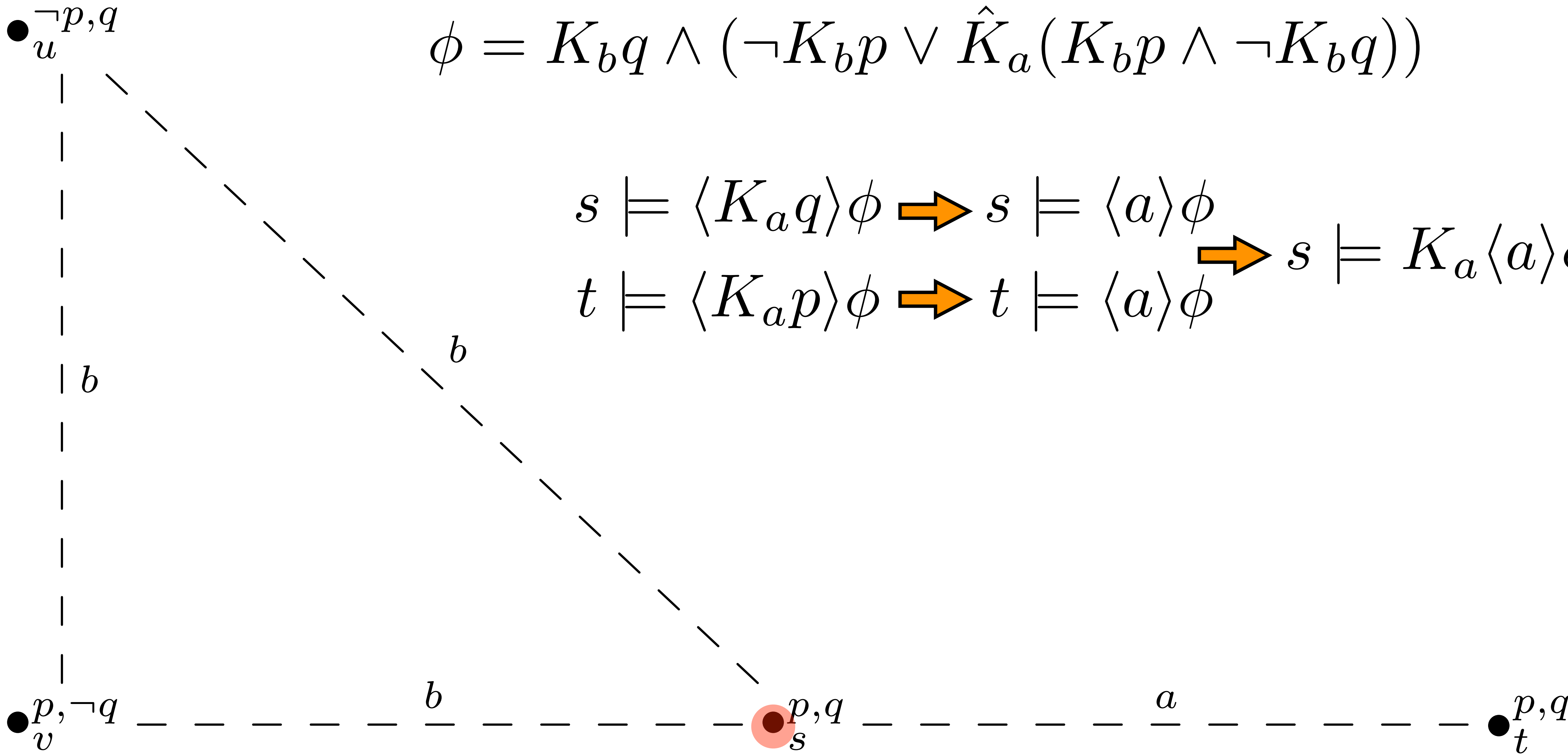
Being able to, knowing that, but not knowing how

$$\phi = K_b q \wedge (\neg K_b p \vee \hat{K}_a(K_b p \wedge \neg K_b q))$$

$$s \models \langle K_a q\rangle\phi \Rightarrow s \models \langle a\rangle\phi$$

$$t \models \langle K_a p\rangle\phi \Rightarrow t \models \langle a\rangle\phi$$

$$\Rightarrow s \models K_a\langle a\rangle\phi$$



Being able to, knowing that, but not knowing how

$$\phi = K_b q \wedge (\neg K_b p \vee \hat{K}_a (K_b p \wedge \neg K_b q))$$

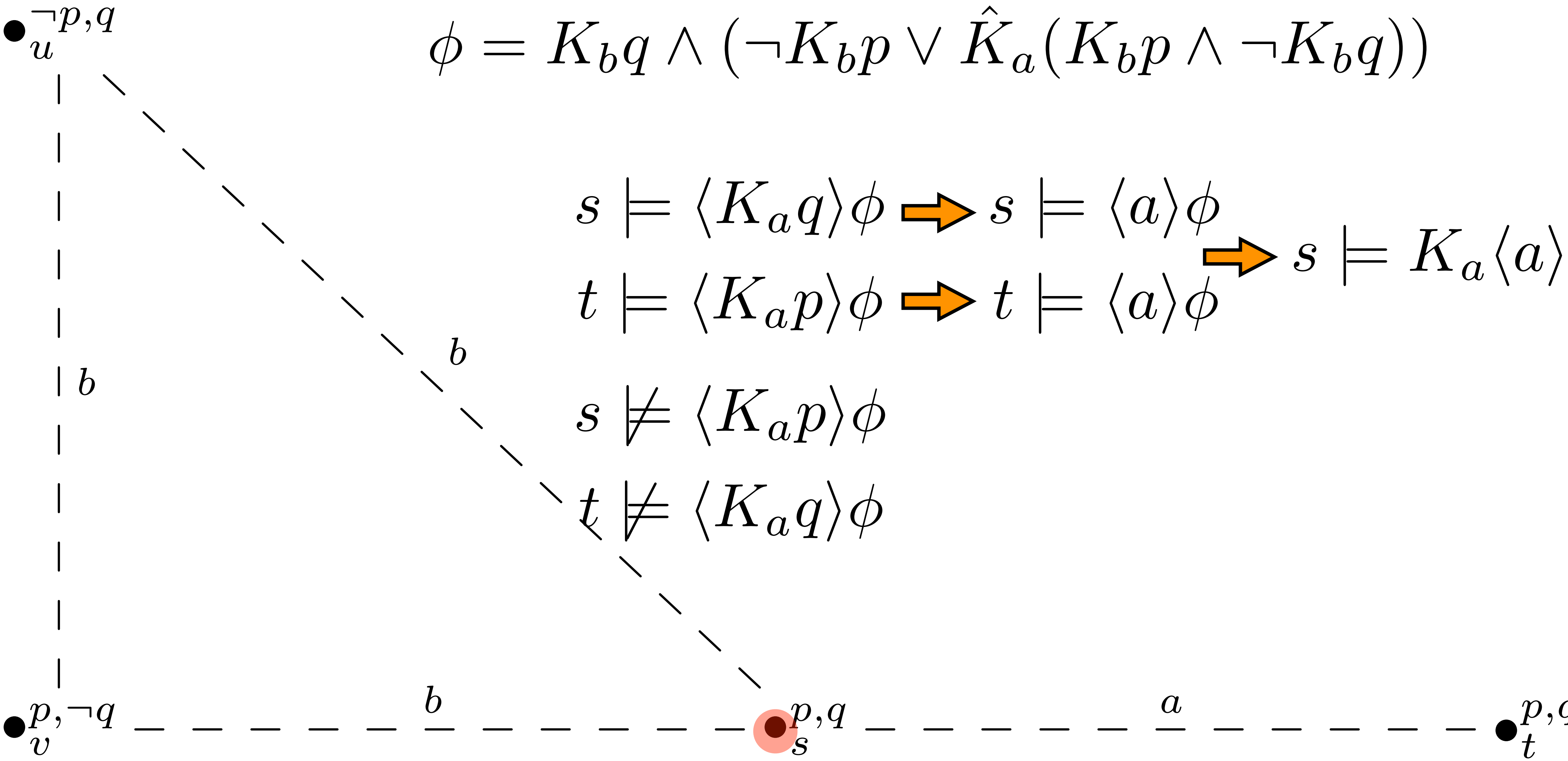
$$s \models \langle K_a q \rangle \phi \Rightarrow s \models \langle a \rangle \phi$$

$$t \models \langle K_a p \rangle \phi \Rightarrow t \models \langle a \rangle \phi$$

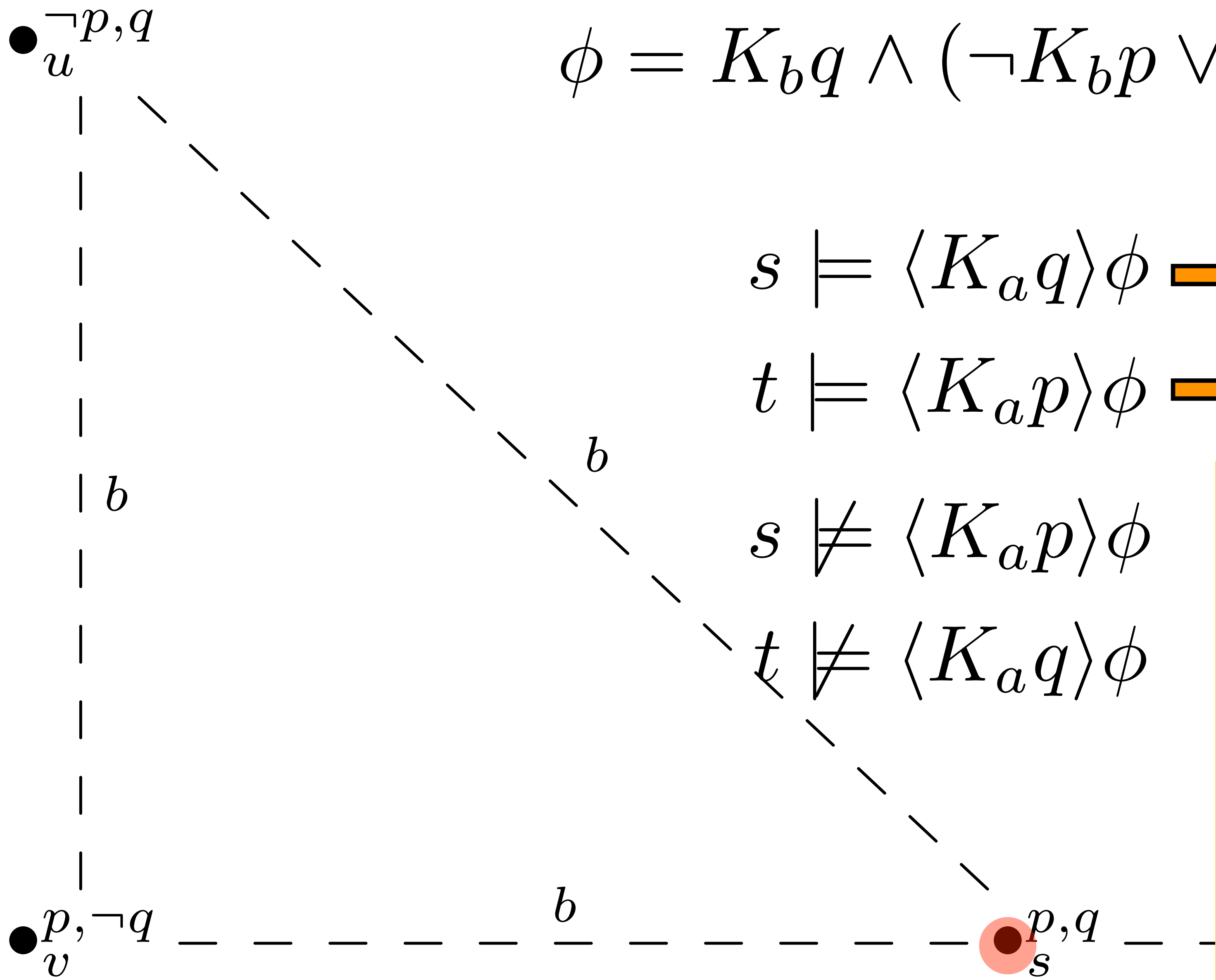
$$\Rightarrow s \models K_a \langle a \rangle \phi$$

$$s \not\models \langle K_a p \rangle \phi$$

$$t \not\models \langle K_a q \rangle \phi$$



Being able to, knowing that, but not knowing how



$$\phi = K_b q \wedge (\neg K_b p \vee \hat{K}_a (K_b p \wedge \neg K_b q))$$

$s \models \langle K_a q \rangle \phi \Rightarrow s \models \langle a \rangle \phi$

$t \models \langle K_a p \rangle \phi \Rightarrow t \models \langle a \rangle \phi$

$\Rightarrow s \models K_a \langle a \rangle \phi$

$s \not\models \langle K_a p \rangle \phi$

$t \not\models \langle K_a q \rangle \phi$

The same announcement will not achieve the goal in both s and t - *a* does not know *how* to achieve it

Expressing knowledge *de dicto*/*de re*

Ability	Knowledge of ability, *de dicto*	Knowledge of ability, *de re*
$\exists\psi\ s \models \langle K_a\psi\rangle\phi$	$\forall s \sim_a t\ \exists\psi\ t \models \langle K_a\psi\rangle\phi$	$\exists\psi\ \forall s \sim_a t\ t \models \langle K_a\psi\rangle\phi$
$\Updownarrow$	$\Updownarrow$	$\Updownarrow$
$s \models \langle a\rangle\phi$	$s \models K_a\langle a\rangle\phi$	??

Expressing knowledge *de dicto*/*de re*

Ability

$$\exists\psi\ s \models \langle K_a\psi\rangle\phi$$

$$\Updownarrow$$

$$s \models \langle a\rangle\phi$$

Knowledge of ability, *de dicto*

$$\forall s \sim_a t\ \exists\psi\ t \models \langle K_a\psi\rangle\phi$$

$$\Updownarrow$$

$$s \models K_a\langle a\rangle\phi$$

Knowledge of ability, *de re*

$$\exists\psi\ \forall s \sim_a t\ t \models \langle K_a\psi\rangle\phi$$

$$\Updownarrow$$

??

$$s \models \langle a\rangle K_a\phi$$

$$\Updownarrow$$

$$\exists\psi\ s \models \langle K_a\psi\rangle K_a\phi$$

Expressing knowledge *de dicto*/*de re*

Ability

$$\exists\psi\ s \models \langle K_a\psi\rangle\phi$$

$$\Updownarrow$$

$$s \models \langle a\rangle\phi$$

Knowledge of ability, *de dicto*

$$\forall s \sim_a t\ \exists\psi\ t \models \langle K_a\psi\rangle\phi$$

$$\Updownarrow$$

$$s \models K_a\langle a\rangle\phi$$

Knowledge of ability, *de re*

$$\exists\psi\ \forall s \sim_a t\ t \models \langle K_a\psi\rangle\phi$$

$$\Updownarrow$$

$$s \models \langle a\rangle K_a\phi$$

$$\Updownarrow$$

$$\exists\psi\ s \models \langle K_a\psi\rangle K_a\phi$$

Expressing knowledge *de dicto*/*de re*

Ability	Knowledge of ability, *de dicto*	Knowledge of ability, *de re*
$\exists\psi\ s \models \langle K_a\psi\rangle\phi$	$\forall s \sim_a t\ \exists\psi\ t \models \langle K_a\psi\rangle\phi$	$\exists\psi\ \forall s \sim_a t\ t \models \langle K_a\psi\rangle\phi$
$s \models \langle a\rangle\phi$	$s \models K_a\langle a\rangle\phi$	$s \models \langle a\rangle K_a\phi$
		$\exists\psi\ s \models \langle K_a\psi\rangle K_a\phi$

Depends on
(1) the fact that actions are *announcements*
(2) the S5 properties

Example: **Russian Cards** (ctnd.)

Ann and Bill *knows how* to
exectute a successful protocol:

$$\langle a \rangle K_a(two \wedge three \wedge \langle b \rangle K_b(one \wedge two \wedge three))$$

Some logical properties

$$[G \cup H]\phi \to [G][H]\phi$$

$$\langle G\rangle[G]\phi \to [G]\langle G\rangle\phi$$ (Church-Rosser)

$$\langle G\rangle[H]\phi \to [H]\langle G\rangle\phi$$ (..generalised)

Axiomatisation

$S5_n$ axioms and rules

PAL axioms and rules

$[G]\phi \rightarrow [\bigwedge_{i \in G} K_i \psi_i]\phi \qquad \text{where } \psi_i \in \mathcal{L}_{el}$

From ϕ, infer $[G]\phi$

From $\phi \rightarrow [\theta][\bigwedge_{i \in G} K_i p_i]\psi$, infer $\phi \rightarrow [\theta][G]\psi$
where $p_i \notin \Theta_\phi \cup \Theta_\theta \cup \Theta_\psi$

Theorem:
Sound & complete.

Model Checking

The model checking problem:

$$\text{Given } M, s \text{ and } \phi, \text{ does } M, s \models \phi \text{ hold?}$$

Theorem:
The model checking problem is PSPACE-complete

(also extends to APAL)

Coalition Announcement Logic

- The coalition operator in GAL does not have the exists-forall semantics of coalition logic

- Coalition Announcement Logic (CAL) is a variant which has that semantics:

$$\langle [G] \rangle \phi$$

  - means that G can make some joint announcement such that no matter what the other agents announce, phi will become true

CAL

$$\varphi ::= p \mid K_i\varphi \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \langle\!\lbrack G \rbrack\!\rangle\varphi \mid [\varphi_1]\varphi_2$$

$$M, s \models \langle\!\lbrack G \rbrack\!\rangle\varphi$$

iff for every agent $i \in G$ there exists a formula ψ_i such that for every formula ψ_j for each of the agents $j \notin G$ we have that $M, s \models \bigwedge_{i \in G} K_i\psi_i \wedge [K_1\psi_1 \wedge \cdots \wedge K_n\psi_n]\varphi$

CAL: some properties

$$
\begin{array}{ll}
(PAN) & \langle K_1\psi_1 \wedge \cdots \wedge K_n\psi_n\rangle\varphi \rightarrow \langle\!\![N]\!\!\rangle\varphi \\
(PA\emptyset) & \langle\!\![\emptyset]\!\!\rangle\varphi \rightarrow [K_1\psi_1 \wedge \cdots \wedge K_n\psi_n]\varphi \\
 & \langle\!\![G]\!\!\rangle\hat{K}_i\phi \rightarrow \hat{K}_i\langle\!\![G]\!\!\rangle\phi \\
(P) & \langle\!\![G]\!\!\rangle p \leftrightarrow p \\
 & \text{.. and all the axioms of coalition logic}
\end{array}
$$

CAL: many open problems

- Complete axiomatisation, ...

Thank you! For more details:

T. Ågotnes and W. Jamroga, *Constructive Knowledge: What Agents can Achieve under Imperfect Information*, Journal of Applied Non-Classical Logic **17**(4), 2007

T. Ågotnes and H. van Ditmarsch, *Coalitions and Announcements*, Proc. AAMAS 2008

T. Ågotnes, P. Balbiani, H. van Ditmarsch and P. Seban, *Group Announcement Logic*, Journal of Applied Logic **8**(1), 2010

T. Ågotnes, T. French and H. van Ditmarsch, *The Undecidability of Group Announcements*, to appear in Proc. of AAMAS 2014.