Dynamic epistemic logic for distributed computing — asynchrony and concurrency

Hans van Ditmarsch, CNRS, LORIA, University of Lorraine

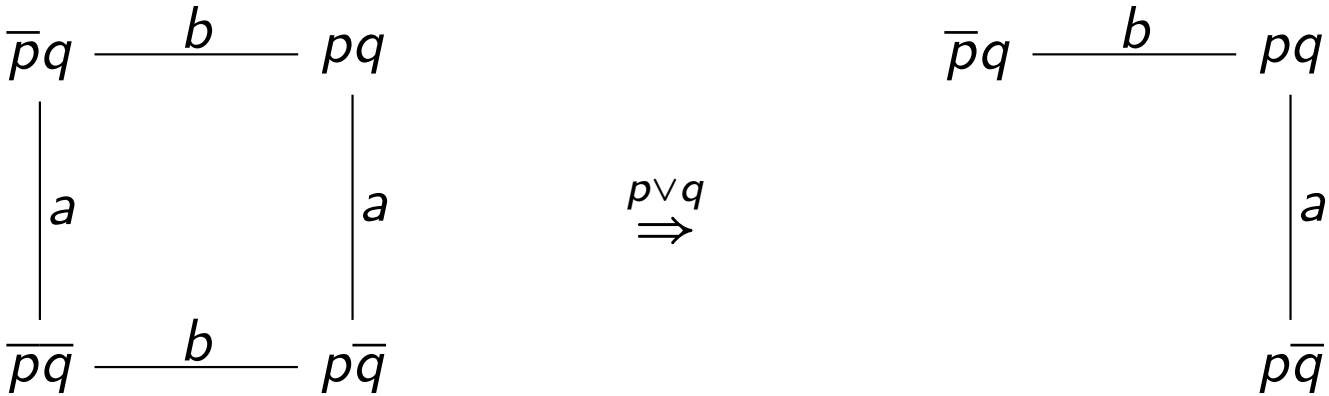
March 6, 2021

ICLA 2021

DEL for DC — asynchrony and concurrency

Dynamic epistemic logic:

Anne knows whether p ('There is a Xmas tree on Place Stanislas').
Bill knows whether q ('St Nicolas has arrived in Nancy').
Anne and Bill are being told that $p \vee q$ is true. What do they learn?
In state $p\overline{q}$ (p is true, q is false), Bill now knows that p is true,
but Anne considers it possible that Bill doesn't know that p is true.



epistemic model $\overset{\text{epistemic action}}{\Rightarrow}$ updated epistemic model

DEL for DC — asynchrony and concurrency

Dynamic epistemic logic:

- $\text{model} \overset{\text{action}}{\Rightarrow} \text{model}_1$

Dynamic epistemic logic for distributed computing:

- **distinguishing histories of epistemic actions**

  $\text{model} \overset{\text{action}}{\Rightarrow} \text{model}_1 \overset{\text{action}_1}{\Rightarrow} \text{model}_2$

  $\text{model} \overset{\text{action}_2}{\Rightarrow} \text{model}_3 \overset{\text{action}_3}{\Rightarrow} \text{model}_4$

  What if the agent cannot distinguish model_1 from model_4?

- **separating sending from receiving actions**

  $\text{model} \overset{\text{send action}}{\Rightarrow} \text{model}_1 \overset{\text{receive that action}}{\Rightarrow} \text{model}_2$

  What if the action (message) is not received by the agent?

- **concurrency in dynamic epistemic logic**

  $$\left.\begin{array}{l} \text{model} \overset{\text{action}}{\Rightarrow} \text{model}_1 \\ \text{model} \overset{\text{action}_1}{\Rightarrow} \text{model}_2 \end{array}\right\} \text{model}_3$$

  What if action and action_1 concurrently result in model_3?

DEL for DC — asynchrony and concurrency

Dynamic epistemic logic for distributed computing:

- **distinguishing histories of epistemic actions**
  - non-public communication with a weak signal (100 prisoners)
  - gossip protocols where agents only know their own calls
  - knowledge for simplicial complexes
- **separating sending from receiving actions**
  - the logic of asynchronous announcements
- **concurrency in dynamic epistemic logic**
  - 'nobody steps forward' *concurrently* in 'muddy children'

Distinguishing histories of epistemic actions

Part I

How to distinguish action histories of different length in DEL?

Distinguishing histories of epistemic actions

Executing an action is like a tick of the clock.
For a public announcement this is what we want.
For private announcements this may not always be what we want.
If this is what we want, dynamic epistemic logic is synchronous.

[van Benthem *et al.*, JPL 2009] propose temporal properties for an agent a to distinguish histories $\alpha, \beta, \ldots$ of actions (events) $e, f, \ldots$.

- *synchronicity*: if $\alpha \sim_a \beta$ then $|\alpha| = |\beta|$
- *(synchronous) perfect recall*: if $\alpha.e \sim_a \beta.f$ then $\alpha \sim_a \beta$
- *no miracles*: if $\alpha.e \sim_a \beta.f$, then $\alpha' \sim_a \beta'$ implies $\alpha'.e \sim_a \beta'.f$

[Dégremont, Löwe, Witzel, TARK 2011] propose properties:

- *(Pnueli) perfect recall*: if $\alpha.e \sim_a \beta$, then $\exists \beta' \sqsubseteq \beta$ with $\alpha \sim_a \beta'$
- *synchronous grounding*:
  $\alpha \sim_a \beta$ and $|\alpha| \leq |\beta|$ imply $\exists \beta' \sqsubseteq \beta$ with $\alpha \sim_a \beta'$ and $|\alpha| = |\beta'|$.

They provide a translation of DEL into asynchronous ETL.
Perfect recall and synchronicity gets synchronous perfect recall.

Distinguishing histories of epistemic actions

This was for TARK 2011. Why is there no journal version?
TARK 2011 co-author Andreas Witzel got a job at Google ...
Maybe there are also other reasons.

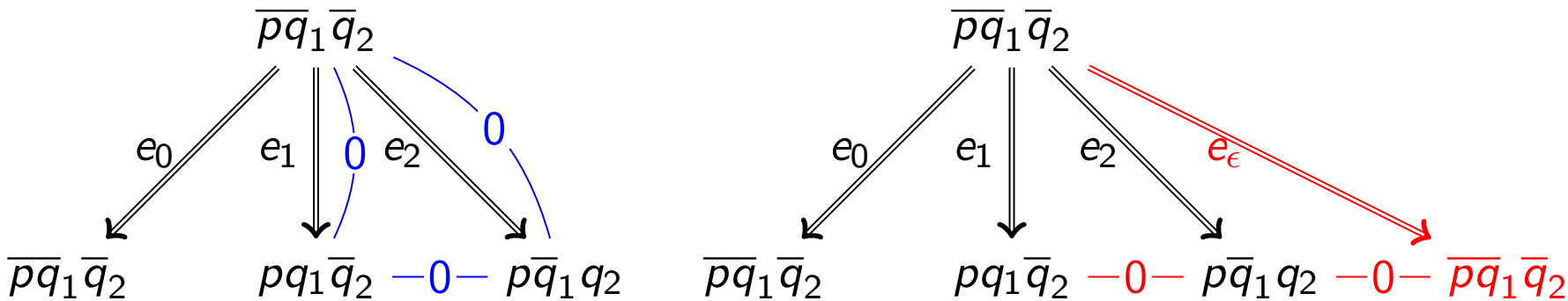
For some applications asynchrony is simulated in synchronous DEL:

- One Hundred Prisoners and a Light Bulb
- Gossip Protocols
- Immediate Snapshot (Distributed Computing)

One Hundred Prisoners and a Light Bulb

- There are three prisoners $0, 1, 2$. Prisoner 0 is the counter.
- e_0: 0 turns off the light when it is on.
- e_1/e_2: 1 and 2 turn on the light the first time they see it's off.
- p: 'the light is on', q_1/q_2: '1/2 has (ever) turned on the light'.
- virtual action e_ϵ: 'nothing happens'

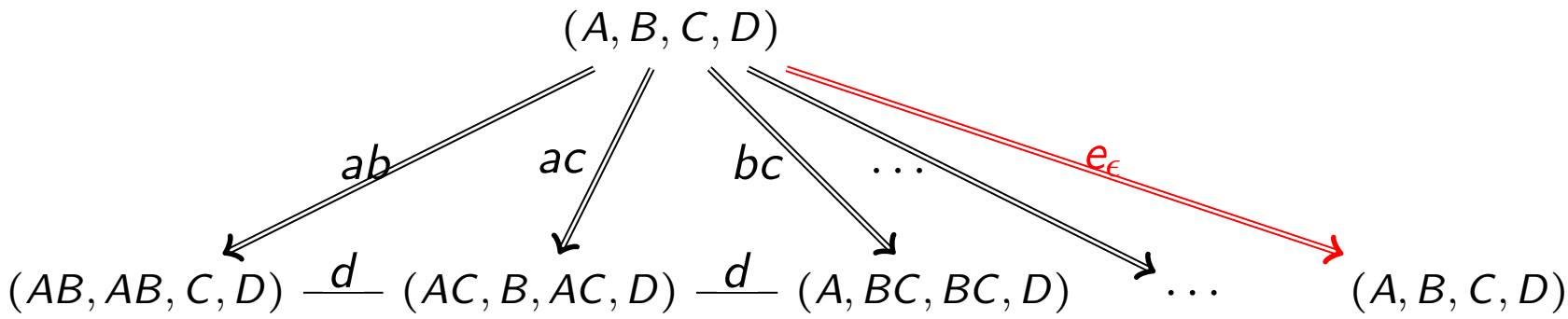
Asynchronous: $e_1 \sim_0 e_2 \sim_0 \epsilon$ Synchronous: $e_1 \sim_0 e_2 \sim_0 e_\epsilon$



[vD, van Eijck, Wu, *One Hundred Prisoners . . . .* KR 2010]
And: $(e_1 \cup e_2) \sim_0 (e_1 \cup e_2)^+$. A poor man's modelling of asynchrony in DEL? Not really. Only the knowledge of 0 and value of p count.

Gossip

In gossip protocols n agents $a, b, c, \ldots$ each holding a secret A, B, C,... call each other until all agents know all secrets. In a call the agents exchange all secrets they know. Agents are only aware of calls they are involved in. If $n = 4$, six different calls may be made.



Trying to model asynchrony, we again added the trivial call e_ϵ. After first call ab, agent d knows that no agent knows more than two secrets. But after second call bc, b and c know three secrets:

$$(A, B, C, D) \stackrel{ab}{\Rightarrow} (AB, AB, C, D) \stackrel{bc}{\Rightarrow} (AB, ABC, ABC, D) \stackrel{ac}{\Rightarrow} \ldots$$

Although $ab \sim_d ac \sim_d bc \ (\sim_d e_\epsilon)$, agent d also cannot distinguish single calls from call sequences: $ab \sim_d ab.bc \sim_d ab.bc.ac \sim_d \ldots$

Gossip

One action: By adding 'nothing happens' action e_ϵ, indistinguish. for all agents from other (indistinguishable) actions, we cannot simulate asynchronous DEL in synchronous DEL.
Arbitrarily many actions: Still, in asynchronous DEL, adding e_ϵ in that way does not change truth: $[\alpha]\varphi \leftrightarrow [\alpha.e_\epsilon]\varphi$.
Boundedly many actions: In PDL for asynchronous gossip, histories may be bounded by removing 'redundant' calls. This makes the execution tree finite, and allows for 'DEL-style' reduction axioms: $[ab]K_c\varphi \leftrightarrow (\dots) \bigwedge_{\alpha \sim_c ab} K_c[\alpha]\varphi$. (Conjunction $\bigwedge$ over finite set.)

Gossip as an action model:
[Attamah, vD, Grossi, vdHoek, *Knowl. and Gossip*, ECAI 2014]
[Gattinger, *New Directions in Model Checking DEL*, PhD, 2018]
Finitary execution trees for gossip protocols:
[Apt, Wojtczak, *Verification (...) Gossip Protocols*, JAIR 2018]
[vD, vdHoek, Kuijer, *The Logic of Gossiping*, AI Journal, 2020]

Immediate snapshot

In distributed computing, in the **immediate snapshot**, processes (agents) asynchronously communicate their local state value to each other by a shared memory. Each process *writes* its value and simultaneously *reads* all values already written. It may do that concurrently with other processes: this determines a *concurrency class*. A *schedule* is an ordered partition of the processes in concurrency classes. For three processes a, b, c, the schedules are $a.b.c, a.bc, abc, b.a.c, \ldots$ Then, $a.c.b \sim_a a.b.c \sim_a a.bc$: a only reads its own value. Also, $a.b.c \not\sim_b a.bc$: in $a.b.c$, b reads the values of a, b, but in $a.bc$, those of a, b, c. Etc.

An action model representing these indistinghuishabilities, has a domain consisting of all different schedules (multiplied by the number of different valuations). Is this asynchronous DEL?

[Ledent, Goubault, Rajsbaum, *A Simplicial Complex Model for DEL to study Distributed Task Computability*. GandALF 2018]
[Ledent, *Geometric semantics for asynch. comput.*, PhD, 2019]

Immediate snapshot

In the action model for the immediate snapshot, domain elements are schedules, i.e., sequences of concurrency classes. The schedule of a snapshot is a way to approach synchronization in an asynchronous system. We could call it a *round* of actions. A schedule is a perfectly acceptable domain element for an action model in synchronous DEL.

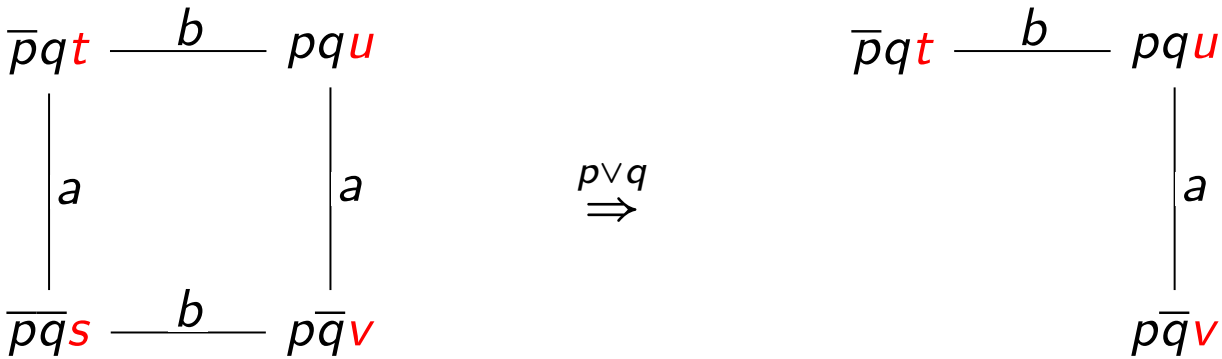
Similarly, for an asynchronous (terminating) gossip protocol we can consider an action model consisting of all maximal call sequences.

Either way we 'flatten' time into the straightjacket of synchronous DEL.

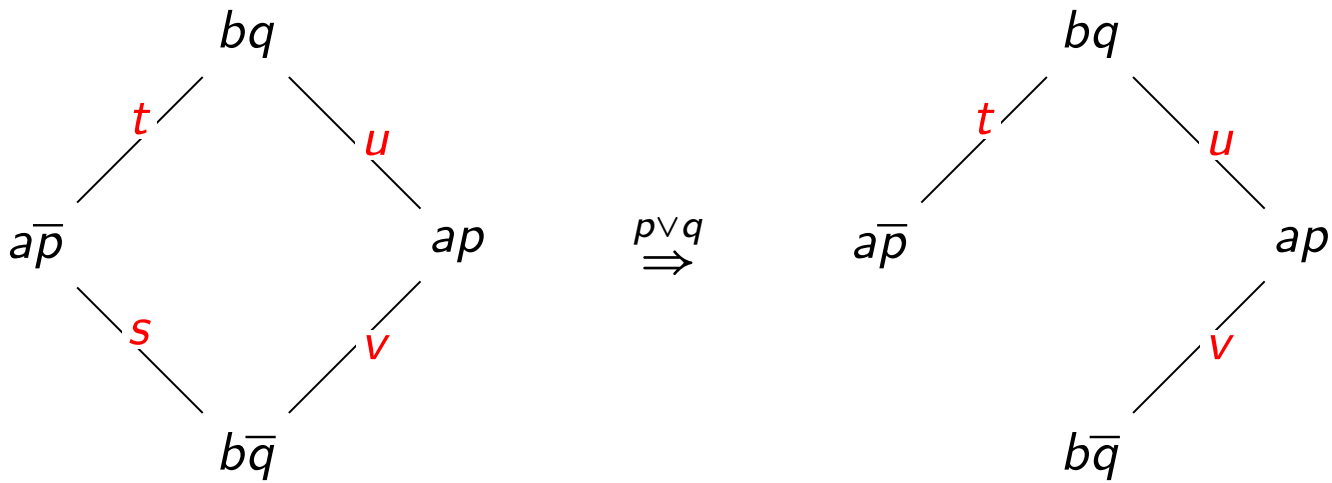
Goubault et al. do more: they translate dynamic epistemic logic into simplicial geometry. This is a gold mine for DEL applications. Let's see some gold.

Epistemic models and simplicial complexes

Anne knows whether p. Bill knows whether q. The environment informs Anne and Bill that $p \vee q$. What do they learn?

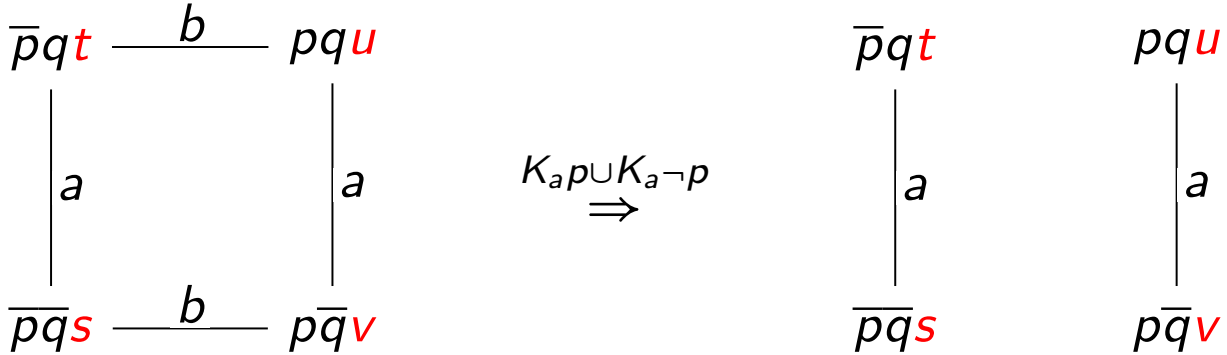


Representation as a 1-dimensional simplicial complex:

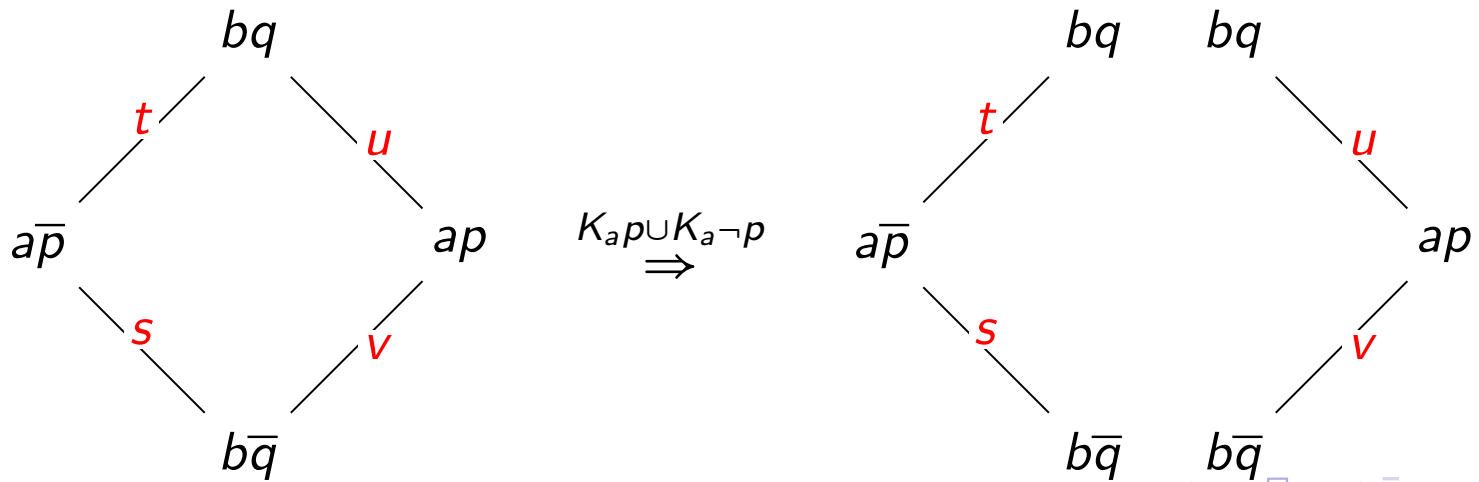


Epistemic models and simplicial complexes

Anne knows whether p. Bill knows whether q. Anne informs Bill of the value of p. What do they learn? (A more suitable example, as one agent informs the other.)



Representation as a 1-dimensional simplicial complex:



Epistemic models and simplicial complexes

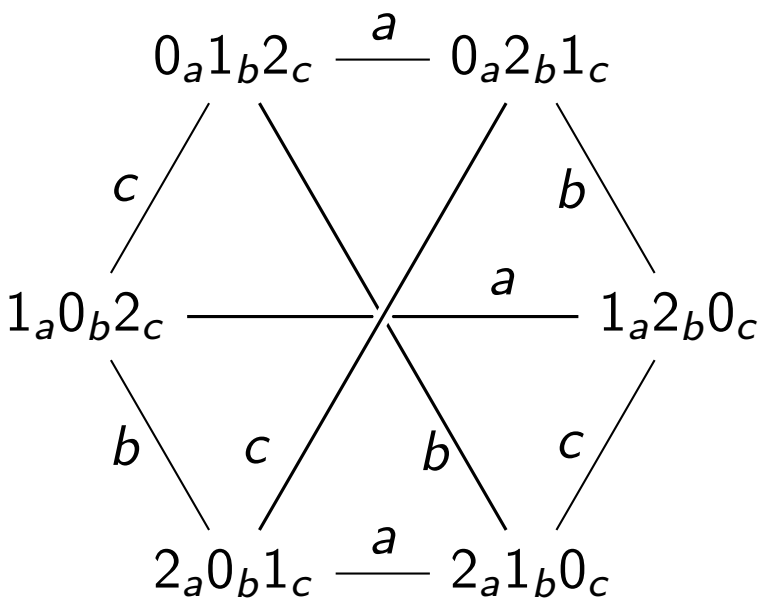
Given a set of *vertices* V, a *simplicial complex* C is a set of non-empty finite subsets of V, called *simplexes*, closed under subsets, and containing all singletons. A simplicial complex is *pure* if all maximal simplexes X have the same *dimension* $|X| - 1$.

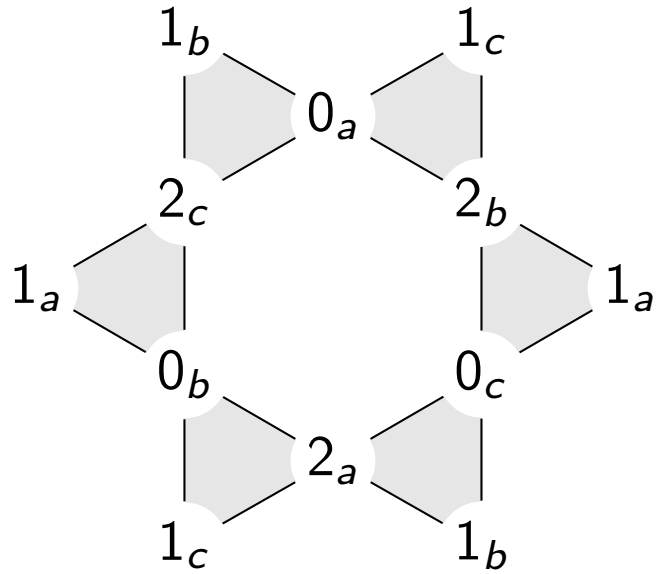
With a *chromatic function* we decorate the vertices of simplicial complexes with *colours* (agents) $a \in A$. Vertices of a simplex must have different colours. A *valuation* assigns *local variables* to vertices. A simplicial complex with a chromatic function and a valuation is a *chromatic simplicial model*.

The simplicial model just represented was 1-dimensional, for 2 agents a, b. Let us see a 2-dimensional simplicial model.

Epistemic models and simplicial complexes

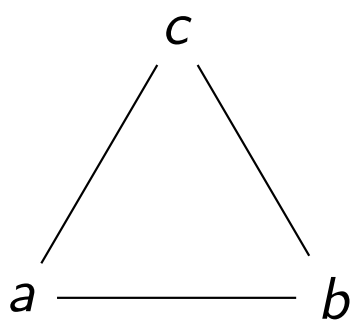
A epistemic model representing three players each holding a card (variable 0_a stands for "player a holds card 0," and so on) and the corresponding 2-dimensional simplicial complex. The outer vertices of for the same agent must be identified.

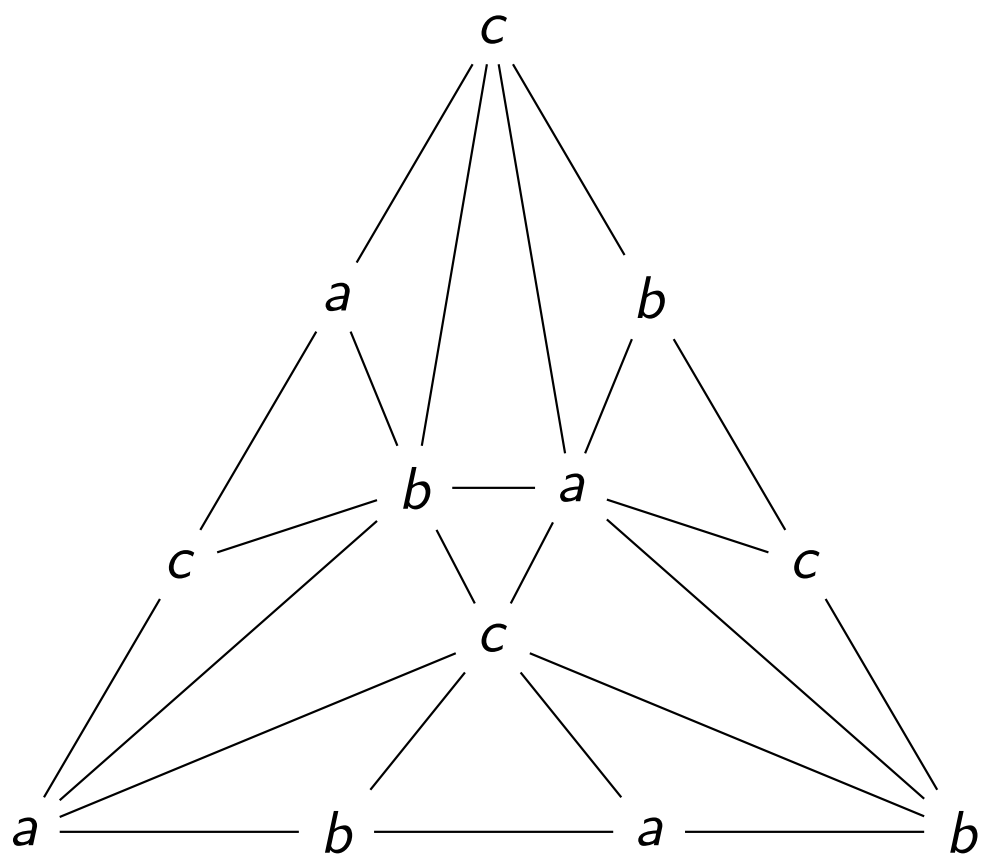




Epistemic models and simplicial complexes

The simplicial action model for the immediate snapshot is what is known as the *standard subdivision*. Without any explanation, on the left a 2-dimensional simplex, on the right its subdivision.





Epistemic models and simplicial complexes

Local epistemic model: all variables are known by some agent a;

- local epistemic models correspond to simplicial complexes
- action models correspond to simplicial complexes
- the update of an epistemic model with an action model corresponds to the product of two simplicial models

Further issues of interest:

- bisimulation for simplicial complexes
- binary consensus, equality negation, ...: as action models
- finite epistemic models can be made local? (open)
- 'common distributed knowledge' on manifolds
- epistemic logic for impure complexes (with crash/failure) *Wanted Dead or Alive*, `arxiv.org/abs/2103.03032`

[vD, Ledent, Goubault, Rajsbaum, *Knowledge and simplicial complexes*, `arxiv.org/abs/2002.08863`, IACAP-2019 book]

Asynchronous histories in dynamic epistemic logic — challenges

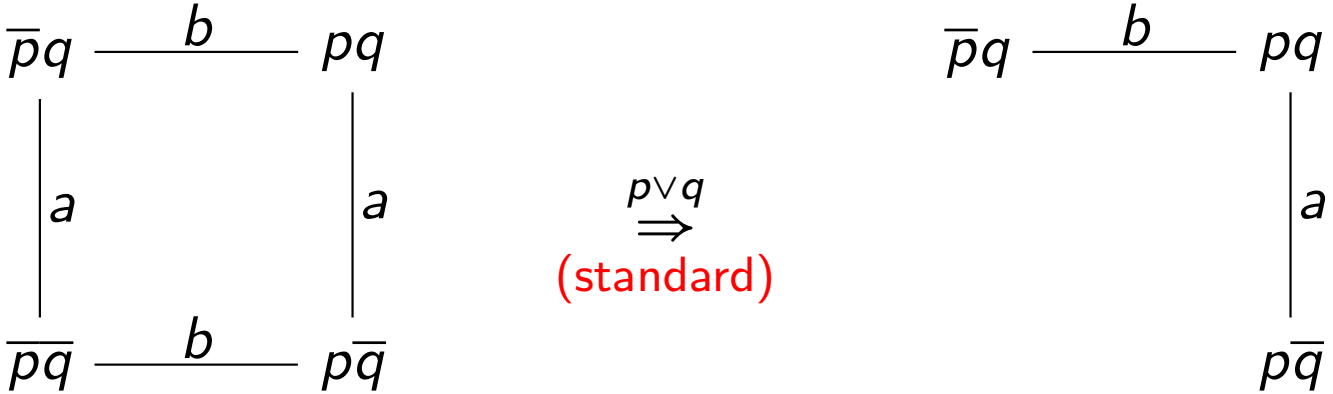
- Dynamic epistemic logic with synchronous grounding for rounds of asynchronous epistemic actions
- An asynchronous dynamic epistemic logic with a history-based semantics of knowledge (assuming some notion of agency)
- Novel group epistemic notions on simplicial complexes
- Modelling in DEL standard tasks in distributed computing

Part II

Separating sending actions from receiving actions in DEL

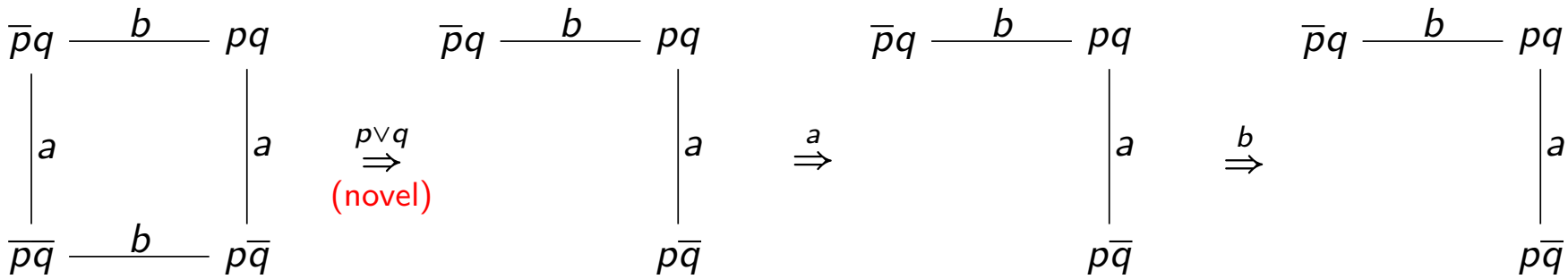
Public announcement (Plaza 1989)

Anne knows whether p. Bill knows whether q.
Anne and Bill are being told that $p \vee q$ is true.
This is the public announcement of $p \vee q$.
Anne and Bill receive the announcement synchronously.



Asynchronous announcement

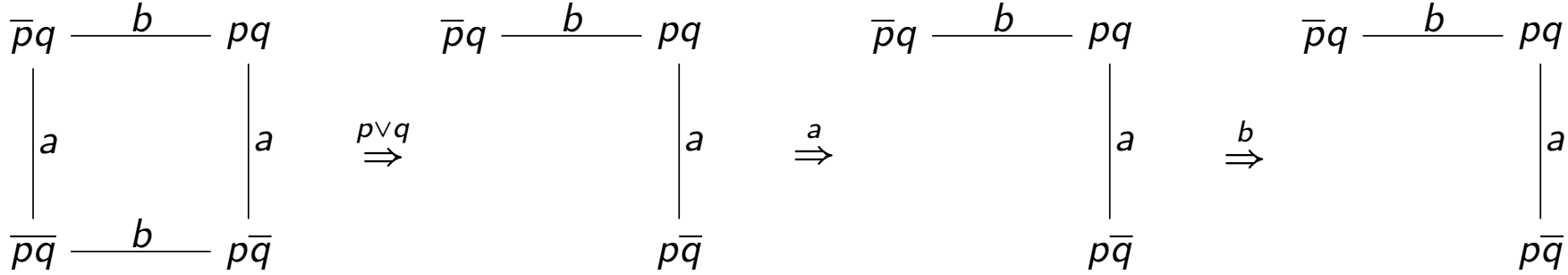
Anne knows whether p. Bill knows whether q.
After the asynchronous announcement of $p \vee q$,
first Anne and then Bill receives this.



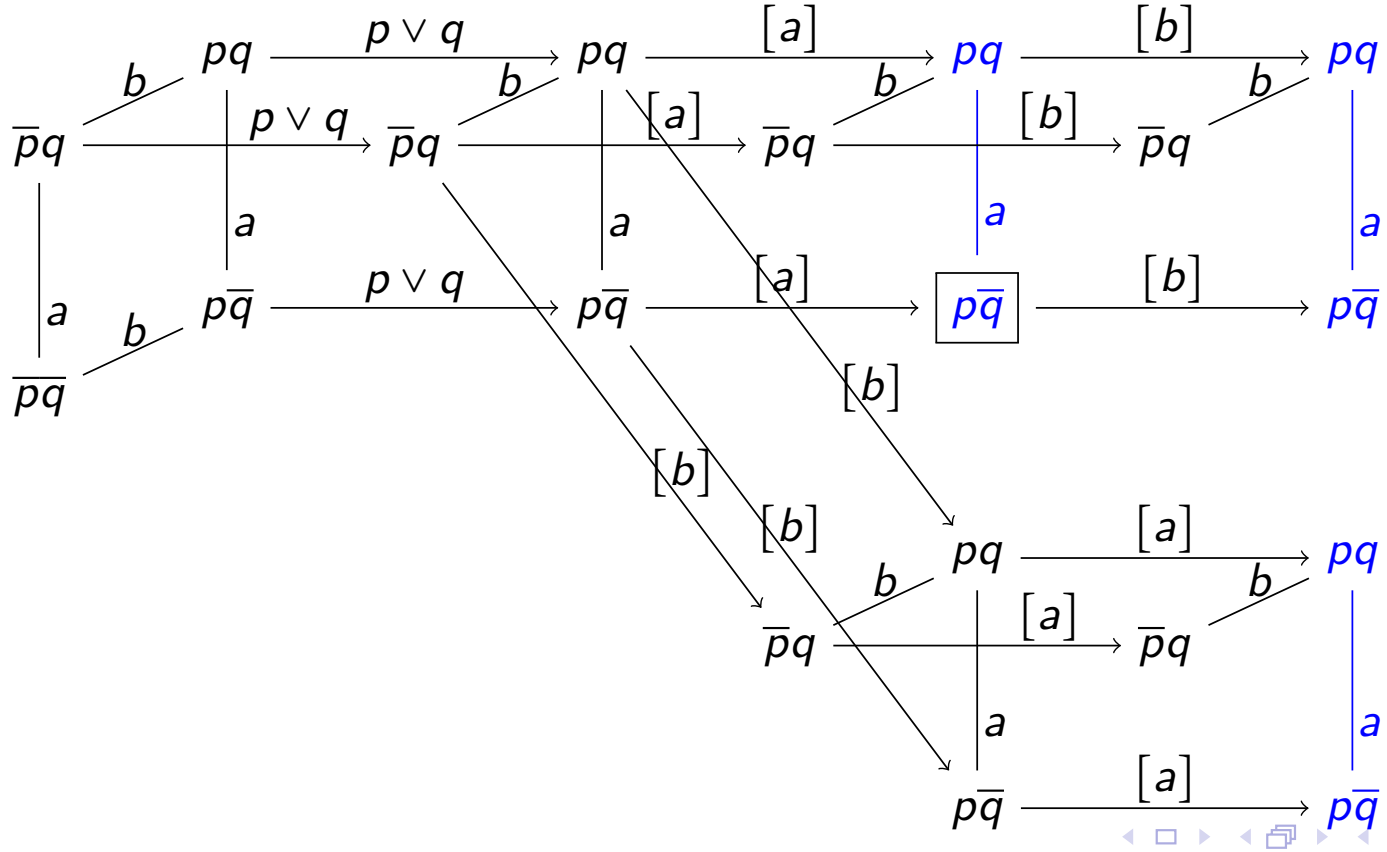
Standard:
$[\varphi]B_a\psi \leftrightarrow (\varphi \to B_a[\varphi]\psi)$
New:
$[\varphi][a]B_a\psi \leftrightarrow (\varphi \to B_a[\varphi][a]\psi \wedge B_a[\varphi][a][b]\psi \wedge B_a[\varphi][b][a]\psi)$

Knowledge or belief? What's going on?

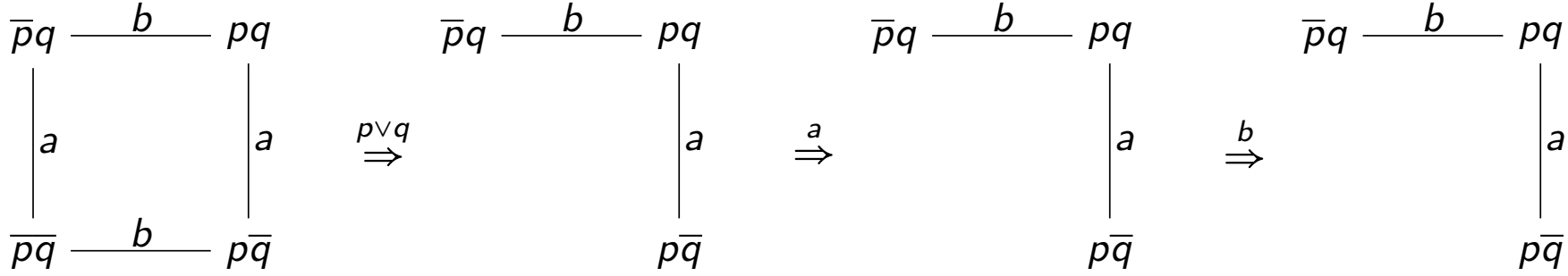
Asynchronous announcement



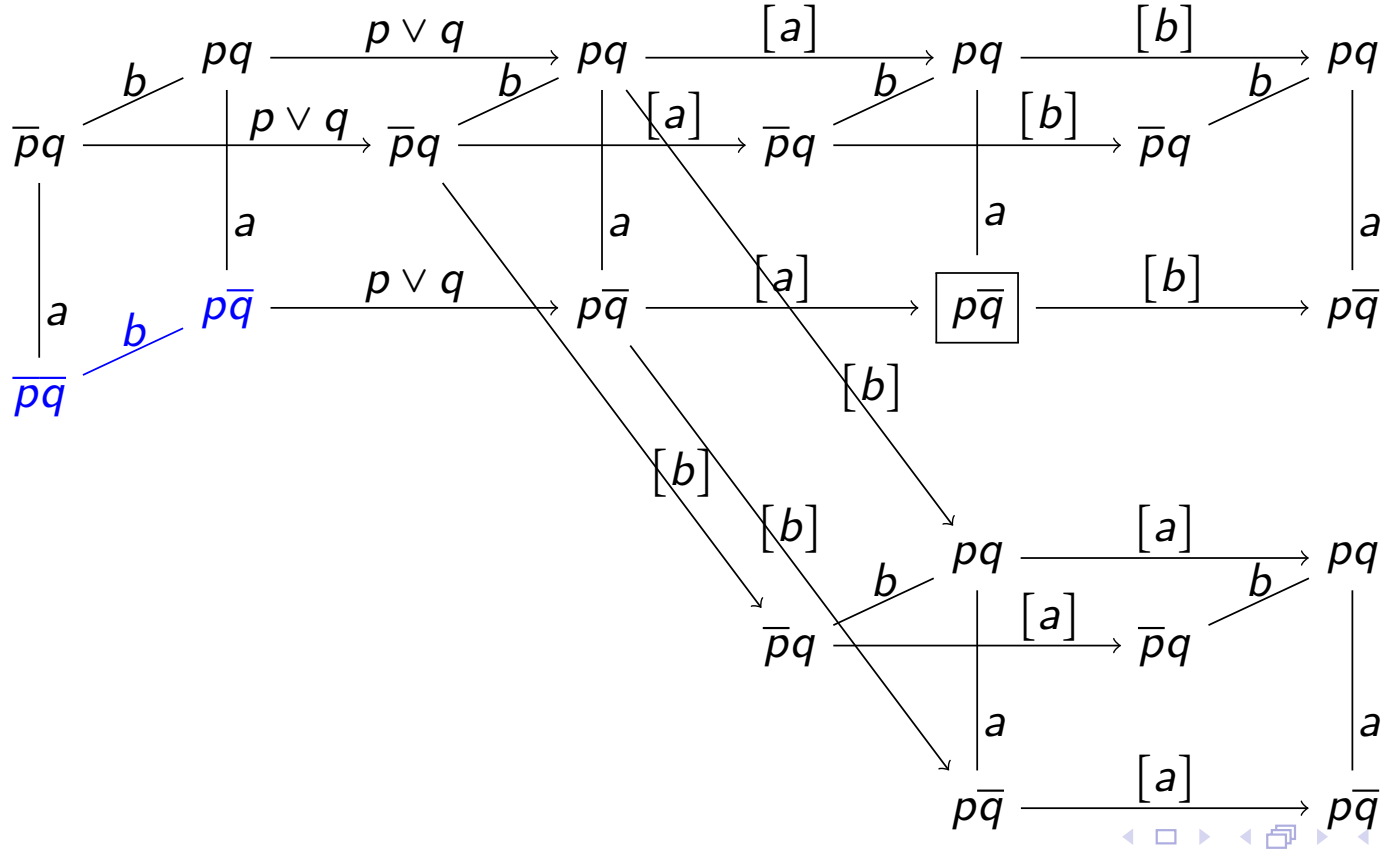
Representation as a history-based model what a considers here



Asynchronous announcement



Representation as a history-based model what b considers here



Announcements, receptions, and histories

- Announcements (messages) are sent by the environment.
- Announcements are individually received by agents.
- Announcements are true when they are sent.
- Announcements are received in the order they were sent.
- Agents assume others receive same/fewer announcements.

— *history* $p \vee q.a.b$: already done.

— *history* $p.b.B_b p.a.a$: B_b, not K_b, see below ...

Anne received both announcements, but Bill only the first one.

After Bill received p, $B_b p$ is true and can be announced.

— *history* $p.B_b p.b.a.a$:

This history is not executable: before $B_b p$ is true, b must receive p.

In distributed computing this would be an inconsistent cut.

— *history* $p.b.(B_b p \wedge B_a \neg B_b p).a.a.b$:

b receives p before a. So $B_b p \wedge B_a \neg B_b p$ is true. (a, b learn this.)

The epistemic operator models belief, not knowledge.

Language of asynchronous announcement logic

Given set P of atoms and set A of agents.

$$\varphi := p \mid \bot \mid \neg\varphi \mid (\varphi \wedge \psi) \mid B_a\varphi \mid [\varphi]\psi \mid [a]\varphi$$

A sequence α of formulas and agents is a *history* if for any prefix, for each a, (number of a's) $\leq$ (number of formulas). (Dyck words)

Given histories α, β, view relation $\rhd_a$ formalizes what histories agent a considers possible.

$\alpha \rhd_a \beta$ iff: α and β contain the same number m of a's,
β contains m announcements,
and those are the first m announcements in α.

Example:
$p.a \rhd_a p.a, p.a.b., p.b.a$
$p.a \rhd_b \epsilon$

Semantics of asynchronous announcement logic

A *model* is a triple (S, R, V). Simultaneously define *satisfaction* relation $\vDash$ and *executability* relation $\bowtie$. (They are well-founded.)

$s, \alpha \vDash p$ iff $s \in V(p)$

$s, \alpha \nvDash \bot$

$s, \alpha \vDash \neg\varphi$ iff $s, \alpha \nvDash \varphi$

$s, \alpha \vDash \varphi \wedge \psi$ iff $s, \alpha \vDash \varphi$ and $s, \alpha \vDash \psi$

$s, \alpha \vDash B_a\varphi$ iff $t, \beta \vDash \varphi$ for all t, β s.t. $R_a st, \alpha \rhd_a \beta,$ and $t \bowtie \beta$

$s, \alpha \vDash [\varphi]\psi$ iff $s, \alpha \vDash \varphi$ implies $s, \alpha\varphi \vDash \psi$

$s, \alpha \vDash [a]\varphi$ iff $|\alpha|_a < |\alpha|_!$ implies $s, \alpha a \vDash \varphi$

$s \bowtie \epsilon$

$s \bowtie \alpha a$ iff $s \bowtie \alpha$ and $|\alpha|_a < |\alpha|_!$

$s \bowtie \alpha\psi$ iff $s \bowtie \alpha$ and $s, \alpha \vDash \psi$

$\vDash^\epsilon \varphi$ (φ is *ϵ-valid*) iff for all (S, R, V) and for all $s \in S$, $s, \epsilon \vDash \varphi$.

$\vDash^* \varphi$ (φ is *$*$-valid*) iff for all (S, R, V) and for all s, α, $s, \alpha \vDash \varphi$.

Set of ϵ-validities: AA. Derivable: $s, \epsilon \vDash \langle\alpha\rangle\varphi$ iff $s, \alpha \vDash \varphi$.

Complete Axiomatization of AA — and relation to PAL

AA $[\alpha]\bot \vee$ (i.e., $\langle\alpha\rangle\rightarrow$) relativizes		*PAL*
propositional tautologies		✓
$B_a(\varphi \rightarrow \psi) \rightarrow B_a\varphi \rightarrow B_a\psi$		✓
$[\alpha]p \leftrightarrow [\alpha]\bot \vee p$		$[\varphi]p \leftrightarrow \varphi \rightarrow p$
$[\alpha a]\bot \leftrightarrow [\alpha]\bot$	if $\lvert\alpha\rvert_a < \lvert\alpha\rvert_!$	...
$[\alpha a]\bot$	if $\lvert\alpha\rvert_a \geq \lvert\alpha\rvert_!$	...
$[\alpha\varphi]\bot \leftrightarrow [\alpha]\bot \vee \neg[\alpha]\varphi$		...
$[\alpha]\neg\varphi \leftrightarrow [\alpha]\bot \vee \neg[\alpha]\varphi$		$[\psi]\neg\varphi \leftrightarrow \psi \rightarrow \neg[\psi]\varphi$
$[\alpha](\varphi \vee \psi) \leftrightarrow [\alpha]\varphi \vee [\alpha]\psi$		$[\chi](\varphi \vee \psi) \leftrightarrow [\chi]\varphi \vee [\chi]\psi$
$[\alpha]B_a\varphi \leftrightarrow [\alpha]\bot \vee \bigwedge\{B_a[\beta]\varphi \mid \alpha \rhd_a \beta\}$		$[\psi]B_a\varphi \leftrightarrow \psi \rightarrow B_a[\psi]\varphi$
from φ and $\varphi \rightarrow \psi$ infer ψ		✓
from φ infer $B_a\varphi$		✓

A special case of belief after history is:

$$[\psi][a]B_a\varphi \leftrightarrow (\psi \rightarrow B_a[\psi][a]\varphi \wedge B_a[\psi][a][b]\varphi \wedge B_a[\psi][b][a]\varphi)$$

Asynchronous announcements — references

- Knight, Maubert, Schwarzentruber, *Reasoning about knowledge and messages in asynchronous multi-agent systems*. MSCS, 2019. (Similar language, different semantics)
- Balbiani, vD, Fernández G., *Asynchronous Announcements*, 2019. `https://arxiv.org/abs/1705.03392`
  Continuation of a 2017 *workshop Strategic Reasoning* version
- Balbiani, vD, Fernández G. *From Public Announcements to Asynchronous Announcements*. Proc. of ECAI Santiago, 2020.
- Balbiani, vD, Fernández G. *Quantifying over Asynchronous Information Change.* Proceedings of AiML Helsinki, 2020.

- ...

Asynchronous reception in dynamic epistemic logic — progress and challenges

- partial synchronization and common belief ($a \in A$, $B \subseteq A$)
  $\varphi := p \mid \bot \mid \neg\varphi \mid (\varphi \wedge \psi) \mid B_a\varphi \mid C_B\varphi \mid [\varphi]\psi \mid [B]\varphi$
  asynchronous announcement is the case : $[\varphi]\psi \mid [a]\psi$
  public announcement is (almost) the case : $[\varphi][A]\psi$
- axiomatization of the partial synch. common belief
- well-founded knowledge semantics (belief: 'knowledge so far')
- axiomatization of asynchronous private announcements
- agents sending each other 'announcements' (messages)

- belief is acknowledgement. This solves Muddy Children.
  $B_a(\neg B_b m_b \vee \neg B_b \neg m_b)$ and $B_b m_b$ may both be true.
  We cannot solve this with knowledge; see 1705.03392.

Part III

Concurrent actions in DEL

Muddy Children are a joy forever

At least two out of three children are muddy. Father announces that at least one child is muddy. He asks the children who know whether they are muddy to step forward. Nobody steps forward.

Nobody steps forward is the public announcement of conjunction:

$$\neg(K_a m_a \vee K_a \neg m_a) \wedge \neg(K_b m_b \vee K_b \neg m_b) \wedge \neg(K_c m_c \vee K_c \neg m_c)$$

This is a short-cut! The action is composed of the individual actions of each child not stepping forward. We can do this too:

$$L_{abc}(L_{abc}?\neg(K_a m_a \vee K_a \neg m_a) \cap L_{abc}?\neg(K_b m_b \vee K_b \neg m_b) \cap L_{abc}?\neg(K_c m_c \vee K_c \neg m_c)$$

where $\cap$ is concurrent execution (or observation). Concurrent DEL is like cPDL [Peleg, 1987]. Composing models from **sets** of models.

Concurrency in dynamic epistemic logic — challenges

- true concurrency (à la Peleg)
- intersection concurrency (à la Harel)
- order independence in histories
- agency?

[vD, Kooi, vd Hoek, *Concurrent Dynamic Epistemic Logic*, 2003]
[van Eijck, Wang, *Composing Models*, 2011]
[Maubert, Pinchinat, Schwarzentr., *Reachability Games in DEL* 2019]

DEL for DC — asynchrony and concurrency

Dynamic epistemic logic for distributed computing — asynchrony and concurrency

Lots of work to do for many years and for many researchers

Thanks!