The Epistemic μ-calculus [1]

Cătălin Dima (LACL)

LACL, U-PEC

Based on joint (and ongoing) work with
R. Bozianu (LACL) and C. Enea (LIAFA)

1. Work partially supported by the ANR research project "EQINOCS" no. ANR-11-BS02-0004

A classical picture in the background

- LTL = FOL1S $\subsetneq$ S1S.
- CTL $\subsetneq$ CTL* $\subsetneq$ SnS.
- S2S = (binary) tree automata = turn-based 2-player games.
- MSO/bisimulation = μ-calculus (on trees).
- ATL $\subsetneq$ ATL* $\subsetneq$ modal μ-calculus.

What about the temporal epistemic framework ?

The μ-calculus of knowledge

Syntax :

$$\varphi ::= p \mid \varphi \wedge \varphi \mid \neg\varphi \mid AX\,\varphi \mid K_a \phi \mid \mu Z.\varphi$$

where $Z \in \mathcal{Z}, a \in Ag, p \in \Pi = \bigcup_{a \in Ag} \Pi_a$.

Synchronous & perfect recall semantics in terms of trees $t : \mathbb{N}^* \rightharpoonup \Pi$,

$$\| \bullet \| : Form(Z_1, \ldots, Z_n) \to \left[\left(2^{\mathrm{supp}(t)}\right)^n \to 2^{\mathrm{supp}(t)} \right]$$

- $\|AX.\phi\|(S_1, \ldots, S_n) = AX\big(\|\phi\|(S_1, \ldots, S_n)\big)$ where

  $$AX(S) = \{x \in \mathrm{supp}(t) \mid \forall i \in \mathbb{N} \text{ if } xi \in \mathrm{supp}(t) \text{ then } xi \in S\}$$

- $\|K_a.\phi\|(S_1, \ldots, S_n) = K_a\big(\|\phi\|(S_1, \ldots, S_n)\big)$ where

  $$K_a(S) = \{x \in \mathrm{supp}(t) \mid \forall y \in \mathrm{supp}(t), \text{ if } x \sim_a y \text{ then } y \in S\}$$

  where $x \sim_a y$ if $\forall z \prec x, z' \prec y, |z| = |z'|$ implies $t(z) \cap \Pi = t(z') \cap \Pi$.

The μ-calculus of knowledge

Syntax :

$$\varphi ::= p \mid \varphi \wedge \varphi \mid \neg\varphi \mid AX\varphi \mid K_a\phi \mid \mu Z.\varphi$$

where $Z \in \mathcal{Z}, a \in Ag, p \in \Pi = \bigcup_{a \in Ag} \Pi_a$.

Synchronous & perfect recall semantics in terms of trees $t : \mathbb{N}^* \rightharpoonup \Pi$,

$$\| \bullet \| : Form(Z_1, \ldots, Z_n) \to \left[\left(2^{\mathrm{supp}(t)}\right)^n \to 2^{\mathrm{supp}(t)} \right]$$

- $\|AX.\phi\|(S_1, \ldots, S_n) = AX\big(\|\phi\|(S_1, \ldots, S_n)\big)$ where

$$AX(S) = \{x \in \mathrm{supp}(t) \mid \forall i \in \mathbb{N} \text{ if } xi \in \mathrm{supp}(t) \text{ then } xi \in S\}$$

- $\|K_a.\phi\|(S_1, \ldots, S_n) = K_a\big(\|\phi\|(S_1, \ldots, S_n)\big)$ where

$$K_a(S) = \{x \in \mathrm{supp}(t) \mid \forall y \in \mathrm{supp}(t), \text{ if } x \sim_a y \text{ then } y \in S\}$$

where $x \sim_a y$ if $\forall z \prec x, z' \prec y, |z| = |z'|$ implies $t(z) \cap \Pi = t(z') \cap \Pi$.

The μ-calculus of knowledge

Syntax :

$$\varphi ::= p \mid \varphi \wedge \varphi \mid \neg\varphi \mid AX\,\varphi \mid K_a\phi \mid \mu Z.\varphi$$

where $Z \in \mathcal{Z}, a \in Ag, p \in \Pi = \bigcup_{a \in Ag} \Pi_a$.

Synchronous & perfect recall semantics in terms of trees $t : \mathbb{N}^* \rightharpoonup \Pi$,

$$\| \bullet \| : Form(Z_1, \ldots, Z_n) \to \left[\left(2^{\mathrm{supp}(t)}\right)^n \to 2^{\mathrm{supp}(t)} \right]$$

- $\|AX.\phi\|(S_1, \ldots, S_n) = AX\big(\|\phi\|(S_1, \ldots, S_n)\big)$ where

$$AX(S) = \{x \in \mathrm{supp}(t) \mid \forall i \in \mathbb{N} \text{ if } xi \in \mathrm{supp}(t) \text{ then } xi \in S\}$$

- $\|K_a.\phi\|(S_1, \ldots, S_n) = K_a\big(\|\phi\|(S_1, \ldots, S_n)\big)$ where

$$K_a(S) = \{x \in \mathrm{supp}(t) \mid \forall y \in \mathrm{supp}(t), \text{ if } x \sim_a y \text{ then } y \in S\}$$

where $x \sim_a y$ if $\forall z \prec x, z' \prec y, |z| = |z'|$ implies $t(z) \cap \Pi = t(z') \cap \Pi$.

The μ-calculus of knowledge

Syntax :

$$\varphi ::= p \mid \varphi \wedge \varphi \mid \neg\varphi \mid AX\,\varphi \mid K_a\phi \mid \mu Z.\varphi$$

where $Z \in \mathcal{Z}, a \in Ag, p \in \Pi = \bigcup_{a \in Ag} \Pi_a$.

Synchronous & perfect recall semantics in terms of trees $t : \mathbb{N}^* \rightharpoonup \Pi$,

$$\| \bullet \| : Form(Z_1, \ldots, Z_n) \to \left[\left(2^{\mathrm{supp}(t)}\right)^n \to 2^{\mathrm{supp}(t)} \right]$$

- $\|AX.\phi\|(S_1, \ldots, S_n) = AX\left(\|\phi\|(S_1, \ldots, S_n)\right)$ where
  $$AX(S) = \{x \in \mathrm{supp}(t) \mid \forall i \in \mathbb{N} \text{ if } xi \in \mathrm{supp}(t) \text{ then } xi \in S\}$$
- $\|K_a.\phi\|(S_1, \ldots, S_n) = K_a\left(\|\phi\|(S_1, \ldots, S_n)\right)$ where
  $$K_a(S) = \{x \in \mathrm{supp}(t) \mid \forall y \in \mathrm{supp}(t), \text{ if } x \sim_a y \text{ then } y \in S\}$$
  where $x \sim_a y$ if $\forall z \prec x, z' \prec y, |z| = |z'|$ implies $t(z) \cap \Pi = t(z') \cap \Pi$.

The modal μ-calculus of knowledge

Syntax :

$$\varphi ::= p \mid \varphi \wedge \varphi \mid \neg\varphi \mid \langle \overline{c} \rangle \varphi \mid K_a \phi \mid \mu Z.\varphi$$

where $Z \in \mathcal{Z}, a \in Ag, p \in \Pi = \bigcup_{a \in Ag} \Pi_a$ and $\overline{c} \in Act = \bigtimes_{a \in Ag} Act_a$.

Synchronous & perfect recall semantics in terms of trees $t : \mathbb{N}^* \rightharpoonup \Pi \times Act$,

$$\| \bullet \| : Form \to \left[\left(2^{\mathrm{supp}(t)} \right)^n \to 2^{\mathrm{supp}(t)} \right]$$

- $\|\langle \overline{c} \rangle.\phi\|(S_1, \dots, S_n) = \langle \overline{c} \rangle \big(\|\phi\|(S_1, \dots, S_n) \big)$ where

  $$\langle \overline{c} \rangle (S) = \{ x \in \mathrm{supp}(t) \mid \forall i \in \mathbb{N} \text{ if } xi \in \mathrm{supp}(t) \text{ and } t\big|_{Act}(xi) = \overline{c} \text{ then } xi \in S \}$$

- $\|K_a.\phi\|(S_1, \dots, S_n) = K_a \big(\|\phi\|(S_1, \dots, S_n) \big)$ where

  $$K_a(S) = \{ x \in \mathrm{supp}(t) \mid \forall y \in \mathrm{supp}(t), \text{ if } x \sim_a y \text{ then } y \in S \}$$

  where $x \sim_a y$ if $\forall z \prec x, z' \prec y$, $|z| = |z'|$ implies $t\big|_{\Pi}(z) \cap \Pi = t\big|_{\Pi}(z') \cap \Pi$ and $t\big|_{Act_a}(z) = t\big|_{Act_a}(z')$.

Issues on the expressivity of $K\mu$

- Common knowledge :

$$C_{a,b}\phi = \nu Z.(\phi \wedge K_a Z \wedge K_b Z)$$

- KB_n through the usual fixpoint definition :

$$Ap\,\mathcal{U}\,q = \mu Z.q \vee (p \wedge A \bigcirc Z)$$

- ATL with perfect information :

$$\langle\langle A\rangle\rangle \diamond p = \mu Z.\Big(p \vee \bigvee_{c_A \in Act_A} \bigwedge_{c_{\overline{A}} \in Act_{\overline{A}}} [c_A, c_{\overline{A}}]Z\Big)$$

Issues on the expressivity of $K\mu$

- Common knowledge :

$$C_{a,b}\phi = \nu Z.(\phi \wedge K_a Z \wedge K_b Z)$$

- KB_n through the usual fixpoint definition :

$$Ap\,\mathcal{U}\,q = \mu Z.q \vee (p \wedge A \bigcirc Z)$$

- ATL with perfect information :

$$\langle\!\langle A \rangle\!\rangle \Diamond p = \mu Z.\Big(p \vee \bigvee_{c_A \in Act_A} \bigwedge_{c_{\overline{A}} \in Act_{\overline{A}}} [c_A, c_{\overline{A}}]Z\Big)$$

Expressivity of $K\mu$

- Expressing winning strategies in 2-player games with the proponent having imperfect observability :

$$\nu Z_n \mu Z_{n-1} \ldots \mu Z_1 . \bigvee_{\alpha \in Act_0} K_a \bigvee_{k \le n} \Big(p_k \wedge \bigwedge_{\beta \in Act_1} [\alpha, \beta] Z_k \Big)$$

Expressivity of $K\mu$?

- ATL with perfect information :

$$\langle\langle A\rangle\rangle \diamond p = \mu Z.\Big(p \vee \bigvee_{c_A \in Act_A} \bigwedge_{c_{\overline{A}} \in Act_{\overline{A}}} [c_A, c_{\overline{A}}]Z\Big)$$

- ATL with imperfect information ?

Expressivity of $K\mu$?

- ATL with perfect information :

$$\langle\!\langle A \rangle\!\rangle \Diamond p = \mu Z.\Big(p \vee \bigvee_{c_A \in Act_A} \bigwedge_{c_{\overline{A}} \in Act_{\overline{A}}} [c_A, c_{\overline{A}}] Z\Big)$$

- ATL with imperfect information ?

Expressivity of $K\mu$?

- ATL with perfect information :

$$\langle\langle A\rangle\rangle \diamond p = \mu Z.\Big(p \vee \bigvee_{c_A \in Act_A} \bigwedge_{c_{\overline{A}} \in Act_{\overline{A}}} [c_A, c_{\overline{A}}]Z\Big)$$

- ATL with imperfect information ?
  - Let's try :

$$\langle\langle A\rangle\rangle \diamond p = \mu Z.K_A\Big(p \vee \bigvee_{c_A \in Act_A} \bigwedge_{c_{\overline{A}} \in Act_{\overline{A}}} [c_A, c_{\overline{A}}]Z\Big)$$

Expressivity of $K\mu$?

- ATL with perfect information :

$$\langle\langle A\rangle\rangle \Diamond p = \mu Z.\Big(p \vee \bigvee_{c_A \in Act_A} \bigwedge_{c_{\overline{A}} \in Act_{\overline{A}}} [c_A, c_{\overline{A}}]Z\Big)$$

- ATL with imperfect information ?
  - Let's try :

$$\langle\langle A\rangle\rangle \Diamond p = \mu Z.K_A\Big(p \vee \bigvee_{c_A \in Act_A} \bigwedge_{c_{\overline{A}} \in Act_{\overline{A}}} [c_A, c_{\overline{A}}]Z\Big)$$

  - Non-feasible strategies !

Expressivity of $K\mu$?

- ATL with perfect information :

$$\langle\!\langle A \rangle\!\rangle \Diamond p = \mu Z.\Big(p \vee \bigvee_{c_A \in Act_A} \bigwedge_{c_{\overline{A}} \in Act_{\overline{A}}} [c_A, c_{\overline{A}}]Z\Big)$$

- ATL with imperfect information ?
  - Let's try :

$$\langle\!\langle A \rangle\!\rangle \Diamond p = K_A \mu Z.\Big(p \vee \bigvee_{c_A \in Act_A} K_A \bigwedge_{c_{\overline{A}} \in Act_{\overline{A}}} [c_A, c_{\overline{A}}]Z\Big)$$

Expressivity of $K\mu$?

- ATL with perfect information :

$$\langle\!\langle A\rangle\!\rangle \Diamond p = \mu Z.\Big(p \vee \bigvee_{c_A \in Act_A} \bigwedge_{c_{\overline{A}} \in Act_{\overline{A}}} [c_A, c_{\overline{A}}]Z\Big)$$

- ATL with imperfect information ?
  - Let's try :

$$\langle\!\langle A\rangle\!\rangle \Diamond p = K_A \mu Z.\Big(p \vee \bigvee_{c_A \in Act_A} K_A \bigwedge_{c_{\overline{A}} \in Act_{\overline{A}}} [c_A, c_{\overline{A}}]Z\Big)$$

  - With distributed knowledge !

Expressivity of $K\mu$?

- ATL with perfect information :

$$\langle\!\langle A \rangle\!\rangle \diamond p = \mu Z.\Big(p \vee \bigvee_{c_A \in Act_A} \bigwedge_{c_{\overline{A}} \in Act_{\overline{A}}} [c_A, c_{\overline{A}}]Z\Big)$$

- ATL with imperfect information ?
  - Let's try :

$$\langle\!\langle A \rangle\!\rangle \diamond p = K_A \mu Z.\Big(p \vee \bigvee_{c_A \in Act_A} K_A \bigwedge_{c_{\overline{A}} \in Act_{\overline{A}}} [c_A, c_{\overline{A}}]Z\Big)$$

$$\langle\!\langle A \rangle\!\rangle \diamond p = \bigwedge_{a \in A} K_a \mu Z.\Big(p \vee \bigvee_{c_A \in Act_A} \bigwedge_{a \in A} K_a \bigwedge_{c_{\overline{A}} \in Act_{\overline{A}}} [c_A, c_{\overline{A}}]Z\Big)$$

Expressivity of $K\mu$?

- ATL with perfect information :

$$\langle\!\langle A \rangle\!\rangle \Diamond p = \mu Z.\Big(p \vee \bigvee_{c_A \in Act_A} \bigwedge_{c_{\overline{A}} \in Act_{\overline{A}}} [c_A, c_{\overline{A}}] Z\Big)$$

- ATL with imperfect information ?
  - Let's try :

$$\langle\!\langle A \rangle\!\rangle \Diamond p = K_A \mu Z.\Big(p \vee \bigvee_{c_A \in Act_A} K_A \bigwedge_{c_{\overline{A}} \in Act_{\overline{A}}} [c_A, c_{\overline{A}}] Z\Big)$$

$$\langle\!\langle A \rangle\!\rangle \Diamond p = \bigwedge_{a \in A} K_a \mu Z.\Big(p \vee \bigvee_{c_A \in Act_A} \bigwedge_{a \in A} K_a \bigwedge_{c_{\overline{A}} \in Act_{\overline{A}}} [c_A, c_{\overline{A}}] Z\Big)$$

  - Yeah, but both are too strong !
  - They require that the objective p be attained at the same moment in each identically observable run !

Expressing single-agent coalition ATL in $K\mu$

- Given a tree model t, modify it by guessing the points z where p happened in the past of z.
- The guessing is encoded in the actions of the agent a, which may choose to force the system remember that p has happened.
- Then $\langle\langle A \rangle\rangle \diamond p$ is equivalent with :

$$\tilde{\phi} = \mu Z. \bigvee_{\alpha \in Act_a} K_a\Big(p \vee past_p \vee \Big(\bigwedge_{\beta \in Act_{Ag \setminus \{a\}}} [\alpha, \beta] Z \Big)$$

- Can be applied by structural induction on the formula.
- If the given tree has a *finite presentation* (regular tree), then the resulting tree also has a *finite presentation*.

Expressing single-agent coalition ATL in $K\mu$

- Given a tree model t, modify it by guessing the points z where p happened in the past of z.
- The guessing is encoded in the actions of the agent a, which may choose to force the system remember that p has happened.
- Then $\langle\langle A \rangle\rangle \diamond p$ is equivalent with :

$$\tilde{\phi} = \mu Z. \bigvee_{\alpha \in Act_a} K_a \Big(p \vee past_p \vee \Big(\bigwedge_{\beta \in Act_{Ag \setminus \{a\}}} [\alpha, \beta] Z \Big) \Big)$$

- Can be applied by structural induction on the formula.
- If the given tree has a *finite presentation* (regular tree), then the resulting tree also has a *finite presentation*.

MSO with binary predicates

- Syntax of MSO_{idobs} :

$$\phi ::= x \mid X \mid p(x) \mid x \in X \mid \phi \wedge \phi \mid \neg\phi \mid \forall x \phi \mid \forall X \phi \mid x \leq y \mid \mathrm{idobs}_\Sigma(x,y)$$

  where x, y are individual variables, X are monadic 2nd order predicates, $p \in \Pi$ and $\Sigma \subseteq \Pi$.
- Usual tree semantics with *bounded* tree width.
- $t, [x \mapsto x_0, y \mapsto y_0] \models idobs_\Sigma(x,y)$ if for all $x' \leq x, \forall y' \leq y, \forall p \in \Sigma$, if $|x'| = |y'|$ then $p(x')$ iff $p(y')$.
  - Same Σ-history on the paths $\epsilon \mapsto x$ and $\epsilon \mapsto y$.

Expressing ATL formulas into MSO_{idobs}

- Uninterpreted atoms = $\Pi \cup \bigcup_{a \in Ag} Act_a$.
- Atoms in each Act_a are exclusive.
- Strategy for player a = 2nd order variable Y.
  - At each position, all Y-successors are labeled with the same atom in Act_a.
  - At each position, if an Y-successors is labeled with $\alpha \in Act_a$, then all successors which bear an α belong to Y.
  - Uniform strategy = the same next action in Act_a is chosen at positions having identically a-observable histories.
- LTL subformulas in the scope of an ATL (ATL*) strategy operator translated as usual.
- Strategies based on common knowledge can be expressed too.
  - Reflexive-transitive closure of $idobs_a \cup idobs_b$ can be expressed.
- Fully-uniform and strictly-uniform strategies can be expressed too.

A gap between $K\mu$ and MSO_{idobs} ?

- Conjecture : ATL and $K\mu$ are incomparable.
- Conjecture : $MSO_{idobs} \supsetneq K\mu$.

- Single-agent $K\mu$ has a decidable satisfiability problem.
  - Reducible to a decidable subproblem of the model-checking problem for $K\mu$ (see below).
- $MSO_{eqlevel}$ has an undecidable satisfiability problem.

Automata techniques ?

A gap between $K\mu$ and MSO_{idobs} ?

- Conjecture : ATL and $K\mu$ are incomparable.
- Conjecture : $MSO_{idobs} \supsetneq K\mu$.

- Single-agent $K\mu$ has a decidable satisfiability problem.
  - Reducible to a decidable subproblem of the model-checking problem for $K\mu$ (see below).
- $MSO_{eqlevel}$ has an undecidable satisfiability problem.

Automata techniques ?

A gap between $K\mu$ and MSO_{idobs} ?

- Conjecture : ATL and $K\mu$ are incomparable.
- Conjecture : $MSO_{idobs} \supsetneq K\mu$.

- Single-agent $K\mu$ has a decidable satisfiability problem.
  - Reducible to a decidable subproblem of the model-checking problem for $K\mu$ (see below).
- $MSO_{eqlevel}$ has an undecidable satisfiability problem.

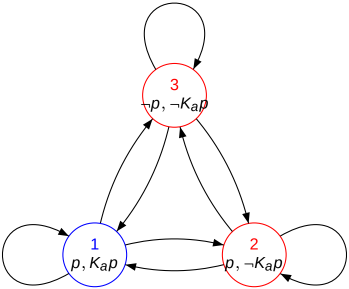
Automata techniques ?

Automata for LTLK

- $\mathcal{A} = (Q, \Pi, \Pi_a, \delta, \pi, \theta, Q_0, \mathcal{R})$.
  - $\delta \subseteq Q \times Q$.
  - $\theta \subseteq 2^Q$: the identical observability constraint.
  - Subsets of initial states : $Q_0 \subseteq 2^Q$.
  - Büchi/Muller/etc. acceptance conditions.
- Runs = Q-trees $t : \mathbb{N}^* \to Q$
  - $(t(x), t(xi)) \in \delta$ for all $xi \in \mathrm{supp}(t)$.
  - $\{t(x) \mid x \sim_a x_0\} \in \theta$, for all $x_0 \in \mathrm{supp}(t)$.
  - Each infinite path in t satisfies $\mathcal{R}$.
- Language = set of trees which are homomorphic images of runs under $\pi : Q \to \Pi$.
  - If t is accepted by $\mathcal{A}$ then any t' with $runs(t') = runs(t)$ is accepted too.
- Notion generalizable to n agents : $(\theta_a)_{a \in Ag}$.

Automata for LTLK (2)

Example for $K_a p$:



$\Pi_a = \emptyset$ $\theta = \{\{1\}, \{2,3\}, \{3\}\}$

- $\forall t : \mathbb{N}^* \to Q$ run in $\mathcal{A}$, for any position $x \in \mathrm{supp}(t)$, $(\pi(t), x) \models K_a p$ iff $t(x) = 1$.
- Similarly, $(\pi(t), x) \models \neg K_a p$ iff $t(x) \in \{2, 3\}$.
- Can be refined for larger Π_a.

Automata for LTLK (3)

- Closed under union.
- Synchronous product, modeling intersection.
- For any LTLK formula ϕ there exists $\mathcal{A}_\phi$ accepting the same set of trees
  - Π-trees, with $\sim_a$ defined by Π_a for each $a \in Ag$.

Proposition (almost not a conjecture)

Single-agent automata have a decidable emptiness problem.

Probable techniques :

- Solving a (synchronous) 2-player game with the proponent (player 0) having incomplete information.
- Constructing a single-agent $K\mu$ formula and testing its satisfiability.

Can be generalized to CTLK.

Model-checking $K\mu$

- Finite models = multi-agent systems $M = \big(Q, Ag, \delta, q_0, \Pi, (\Pi_a)_{a\in Ag}, \pi\big)$.
- $M \models \phi$ if the tree unfolding t_M satisfies ϕ, $\epsilon \in \|\phi\|(S_1, \ldots, S_n)$ for all $S_1, \ldots, S_n \subseteq \mathrm{supp}(t_M)$.
- Model-checking is undecidable for the μ-calculus of knowledge.
  - Subsumes *CTLC* (aka. CL_n from Halpern & Vardi '86), multi-agent *CTLK* with common knowledge.

Model-checking $K\mu$ (2)

- Decidable subproblem generalizing the need of a hierarchy of observations (Kupferman & Vardi, v.d. Meyden & Wilke & Engelhardt & Su, Finkbeiner & Schewe) :

  ϕ **mixes observations of** a **and** b if $\exists$ subformula $\phi' = K_a\psi$ or $\phi' = P_a\psi$ with ψ containing a free variable Z and s.t. in ψ an epistemic operator for b is applied to a subformula in which the same Z is free.

The non-mixing model checking problem :

Decide whether $t_M \models \phi$ for all instances in which any two agents a, b which have mixed observations in ϕ have compatible observability in M.

- I.e. $\Pi_a \subseteq \Pi_b$ or $\Pi_b \subseteq \Pi_a$.

- An instance (M, ϕ) with $\phi = C_{a,b}p = \nu Z.(p \wedge K_a Z \wedge K_b Z)$ is non-mixing iff a and b have compatible observability in M.
- $K_a K_b \Box p$ is non-mixing for any Π_a and Π_b.
- Subsumes known cases of decidable model-checking problems for LTLK/CTLK/ATL.

Technical approach for proving decidability of the non-mixing model-checking problem

Show that a finitary semantics suffices :

- State-based semantics : $[\bullet] : Form \to \left[(2^Q)^n \to 2^Q\right]$.
- Decidability of the non-epistemic μ-calculus (with tree semantics) :

$$\begin{tikzcd}
(2^Q)^n \arrow[r, "{[\phi]}"] \arrow[d, "(t_M^{-1})^n"'] & 2^Q \arrow[d, "t_M^{-1}"] \\
(2^{\mathrm{supp}(t_M)})^n \arrow[r, "\|\phi\|"] & 2^{\mathrm{supp}(t_M)}
\end{tikzcd}$$

- Generalizable to the μ-calculus of knowledge by including subset-refinements of M.
- Subset construction w.r.t. a commutes with subset construction for agent b only if Π_a and Π_b are compatible ($\subseteq$ or $\supseteq$).

Technical approach for proving decidability of the non-mixing model-checking problem

Show that a finitary semantics suffices :

- State-based semantics : $\lceil \bullet \rceil : \mathit{Form} \to \left[(2^Q)^n \to 2^Q \right]$.
- Decidability of the non-epistemic μ-calculus (with tree semantics) :

$$
\begin{tikzcd}
(2^Q)^n \arrow[r, "\lceil \phi \rceil"] \arrow[d, "(t_M^{-1})^n"'] & 2^Q \arrow[d, "t_M^{-1}"] \\
(2^{\mathrm{supp}(t_M)})^n \arrow[r, "\|\phi\|"] & 2^{\mathrm{supp}(t_M)}
\end{tikzcd}
$$

- Generalizable to the μ-calculus of knowledge by including subset-refinements of M.
- Subset construction w.r.t. a commutes with subset construction for agent b only if Π_a and Π_b are compatible ($\subseteq$ or $\supseteq$).

Technical approach for proving decidability of the non-mixing model-checking problem

Show that a finitary semantics suffices :

- State-based semantics : $\lceil \bullet \rceil : Form \to \left[(2^Q)^n \to 2^Q \right]$.
- Decidability of the non-epistemic μ-calculus (with tree semantics) :

$$\begin{tikzcd} (2^Q)^n \arrow[r, "\lceil \phi \rceil"] \arrow[d, "(t_M^{-1})^n"'] & 2^Q \arrow[d, "t_M^{-1}"] \\ (2^{\mathrm{supp}(t_M)})^n \arrow[r, "\|\phi\|"] & 2^{\mathrm{supp}(t_M)} \end{tikzcd}$$

- Generalizable to the μ-calculus of knowledge by including subset-refinements of M.
- Subset construction w.r.t. a commutes with subset construction for agent b only if Π_a and Π_b are compatible ($\subseteq$ or $\supseteq$).

Remarks and further work

- Maybe fixpoint variants of the ATL operators are better ?
- Tree automata for the μ-calculus of knowledge (work under progress).
- What if we replace *idobs* predicates with 3rd order predicates ?...
  - This would allow *comparing* sets of runs in a system.
- Automata for $K\mu$ and MSO :
  - “Strict” tree versions, alternating generalizations.
  - Difference between $K\mu$ and MSO_{idobs} lies in the presence/absence of an extra constraint on labeling of nodes in a run with sets of states.