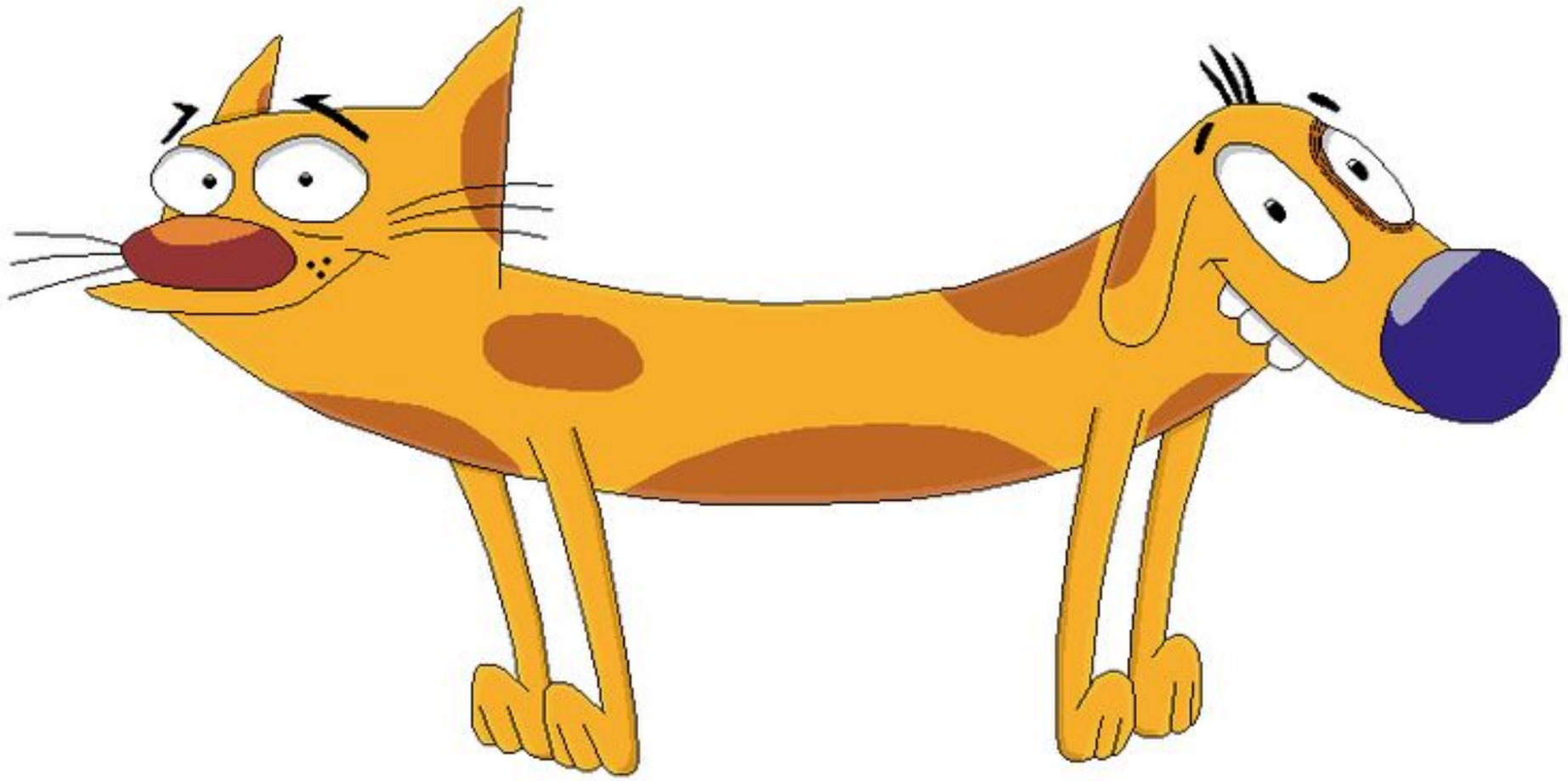
The Dynamic Turn in Strategy Logics: On Individual and Group Abilities

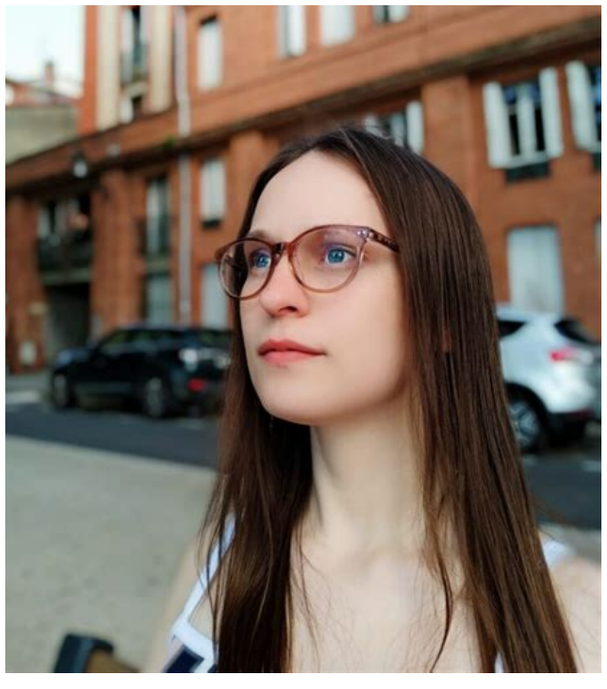
Rustam Galimullin
rustam.galimullin@uib.no
University of Bergen, Norway



Davide Catta
LIPN

Maksim Gladyshev
U. of Utrecht

Hermine Grosinger
Örebro U.



Munyque Mittelmann
LIPN

Nima Motamed
U. of Utrecht

Thomas Ågotnes
U. of Bergen

The Dynamic Turn in Strategy Logics

The Dynamic Turn in Strategy Logics

Blue Sky Ideas Track

Rustam Galimullin
University of Bergen
Bergen, Norway
rustam.galimullin@uib.no

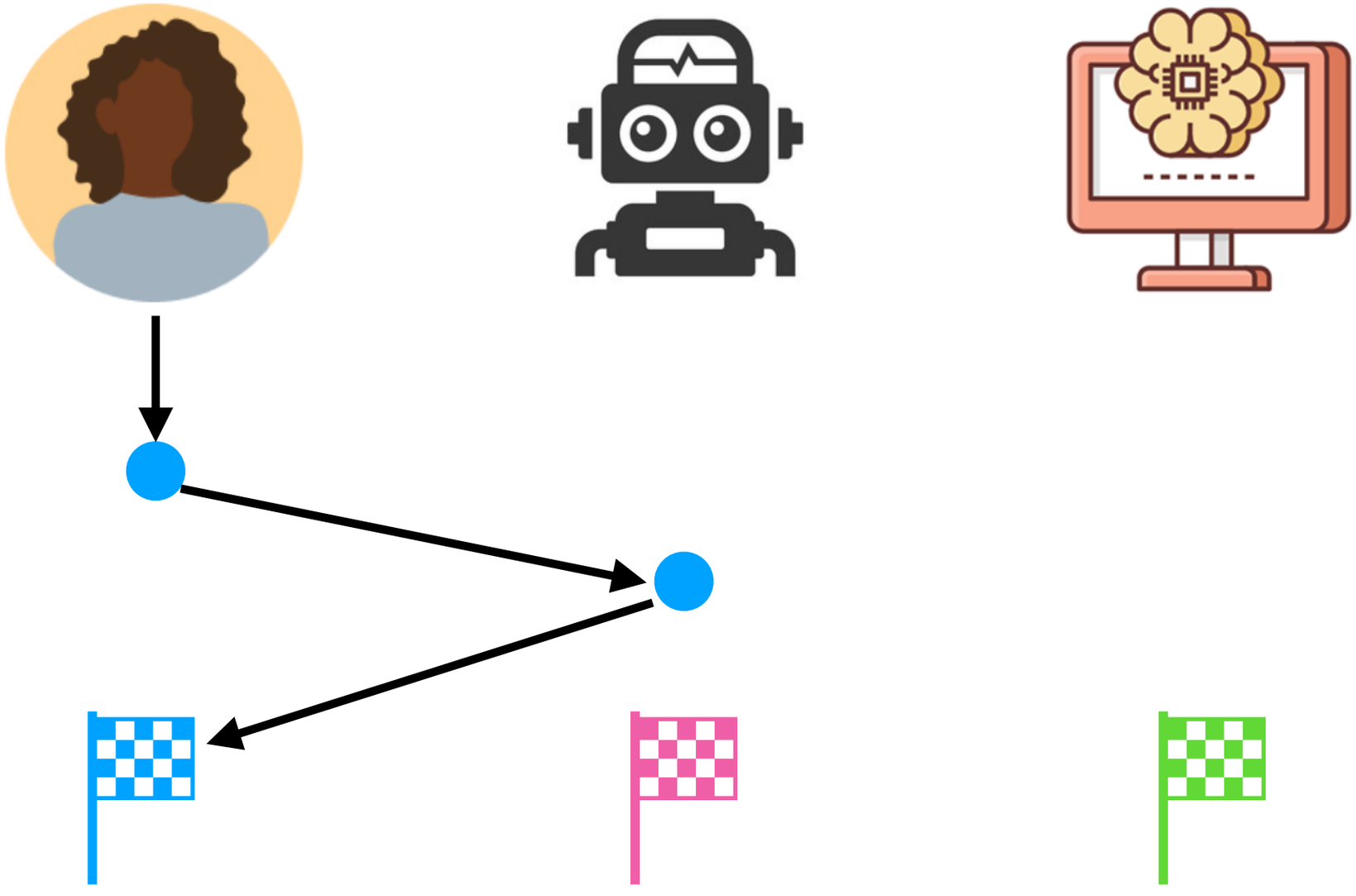
Maksim Gladyshev
Utrecht University
Utrecht, The Netherlands
m.gladyshev@uu.nl

Munyque Mittelmann
LIPN, CNRS, Université Sorbonne Paris Nord
Villetaneuse, France
mittelmann@lipn.univ-paris13.fr

Nima Motamed
Utrecht University
Utrecht, The Netherlands
n.motamed@uu.nl

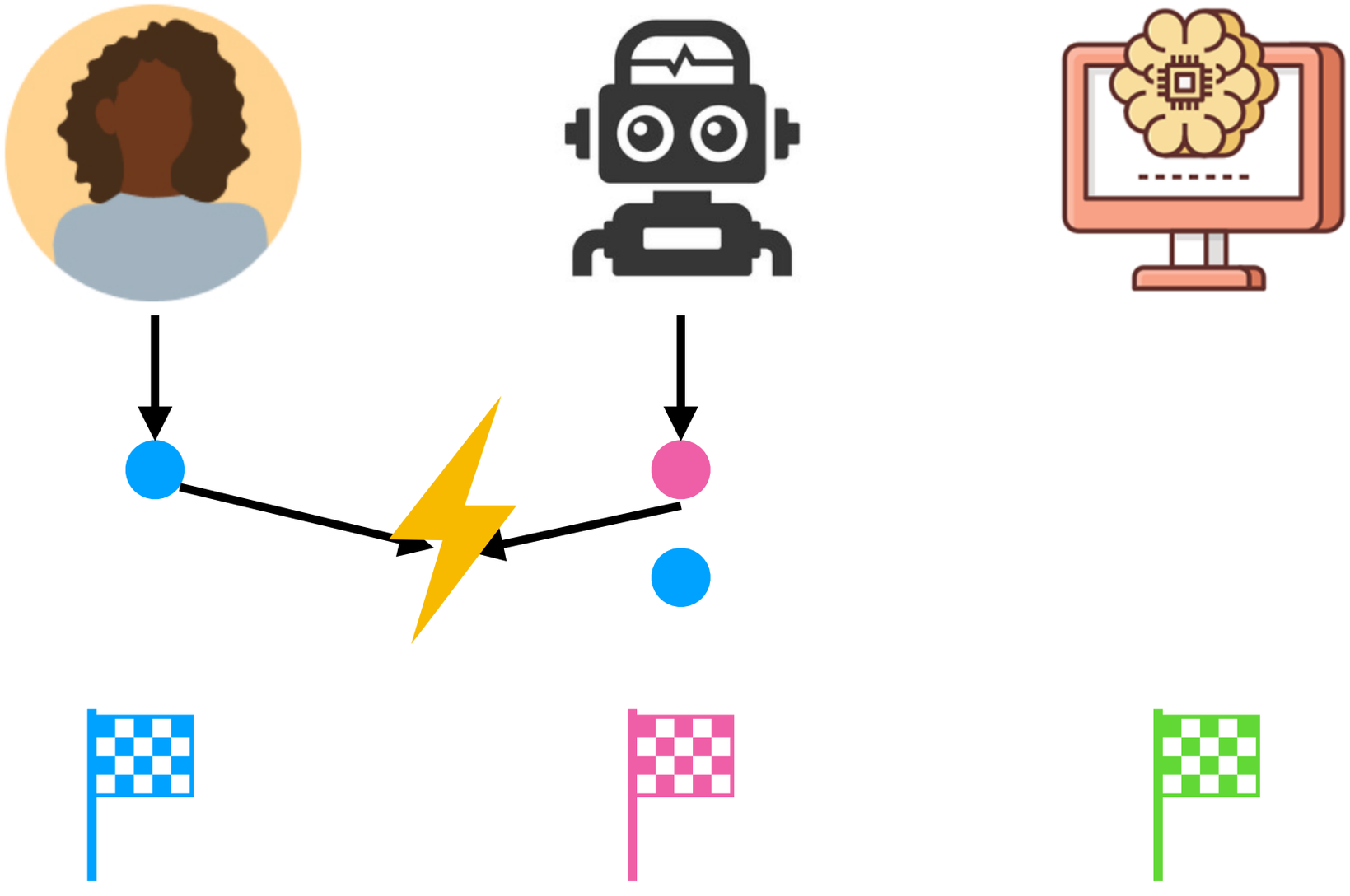
Strategy Logics

Strategy logics (SLs) is a family of formal frameworks devised for specification and verification of multi-agent systems



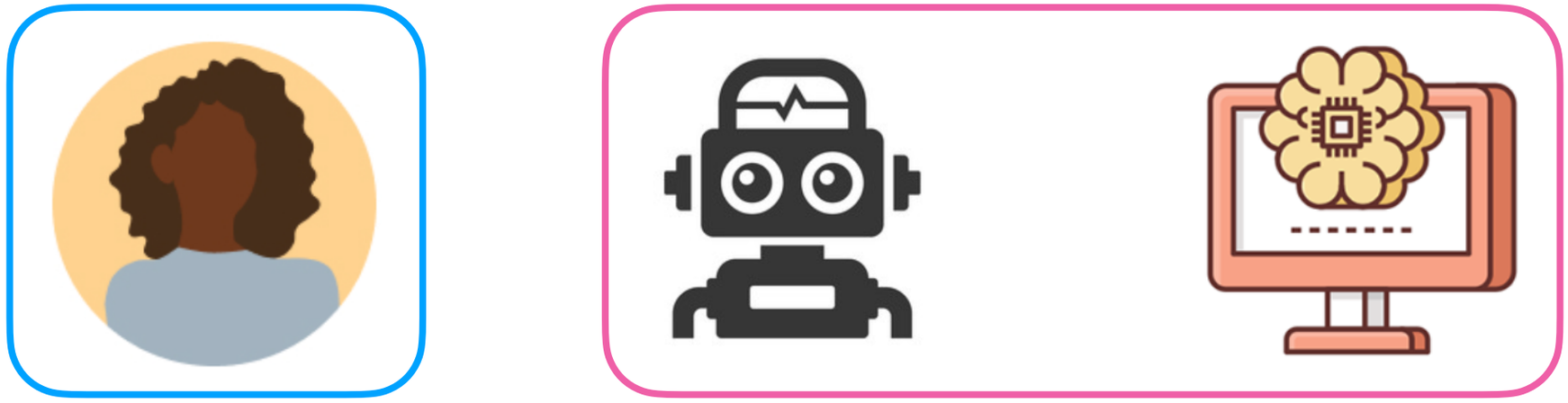
Strategy Logics

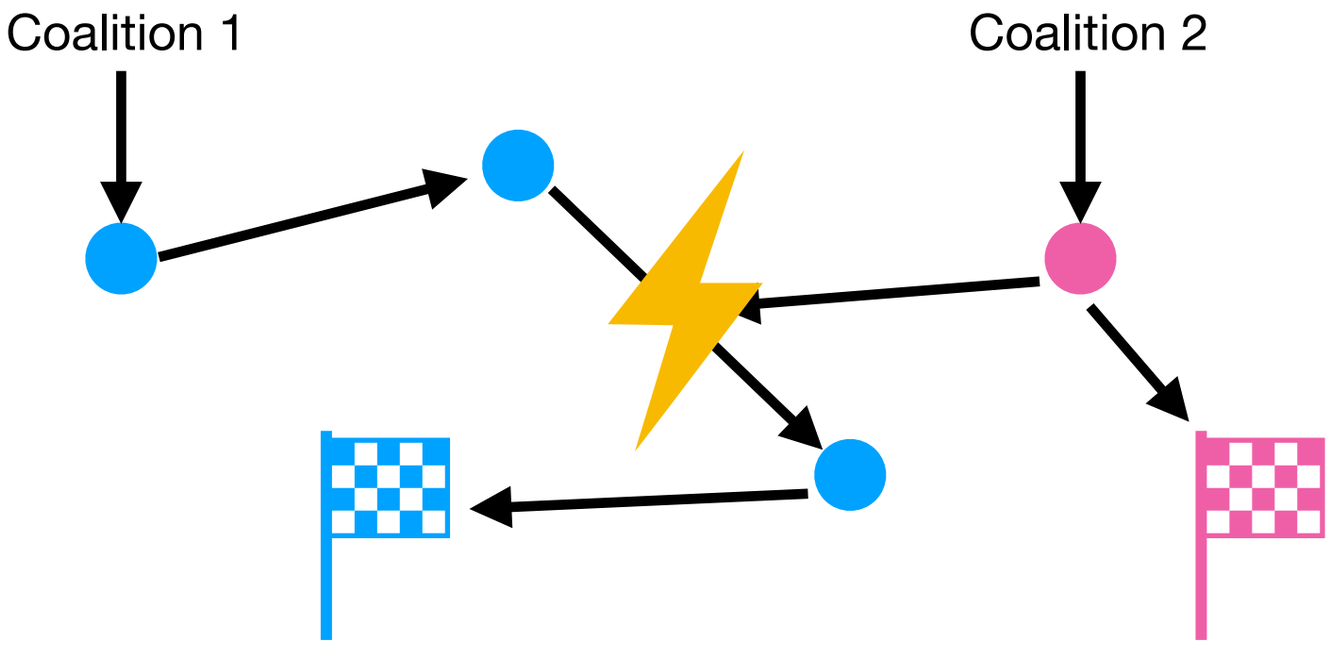
Strategy logics (SLs) is a family of formal frameworks devised for specification and verification of multi-agent systems



Strategy Logics

Strategy logics (SLs) is a family of formal frameworks devised for specification and verification of multi-agent systems





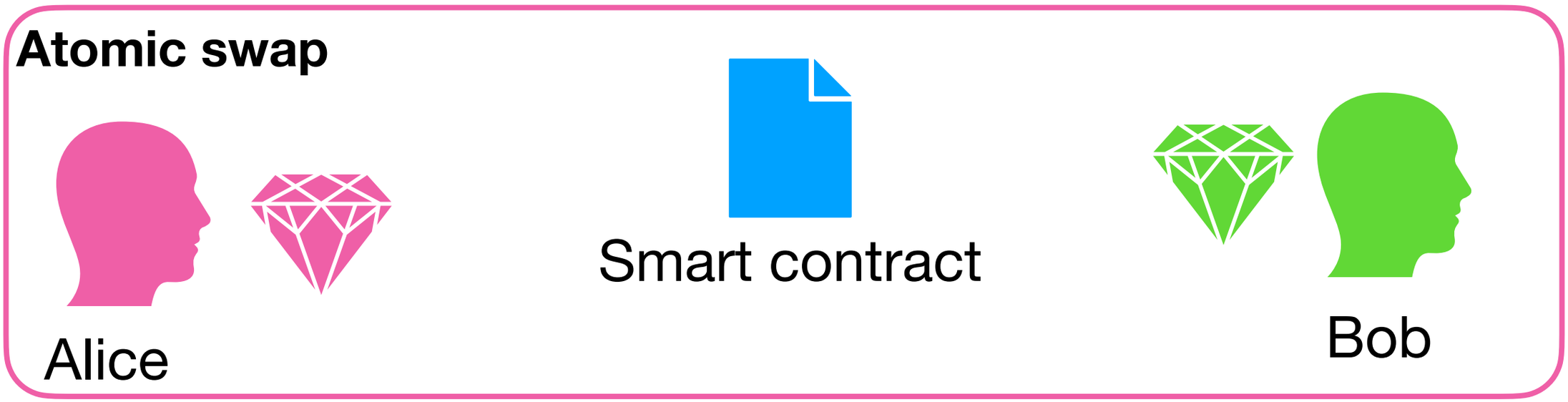
Strategy Logics

We focus on logics interpreted on Concurrent Game Models (CGMs) like Coalition Logic, ATL, etc

Let P be a countable set of propositional variables, and $A = \{1,...,n\}$ be a finite set of agents. A CGM M is (Act, S, act, τ, V), where Act is a finite set of actions, $S \neq \emptyset$ is a set of states, $act : A \times S \to 2^{Act} \backslash \emptyset$ assigns a set of actions to each agent, $\tau : \{s, a_1, \ldots, a_n \mid s \in S, a_i \in act(s, i)\} \to S$ is an outcome function, and $V : S \to 2^P$ is a valuation function.

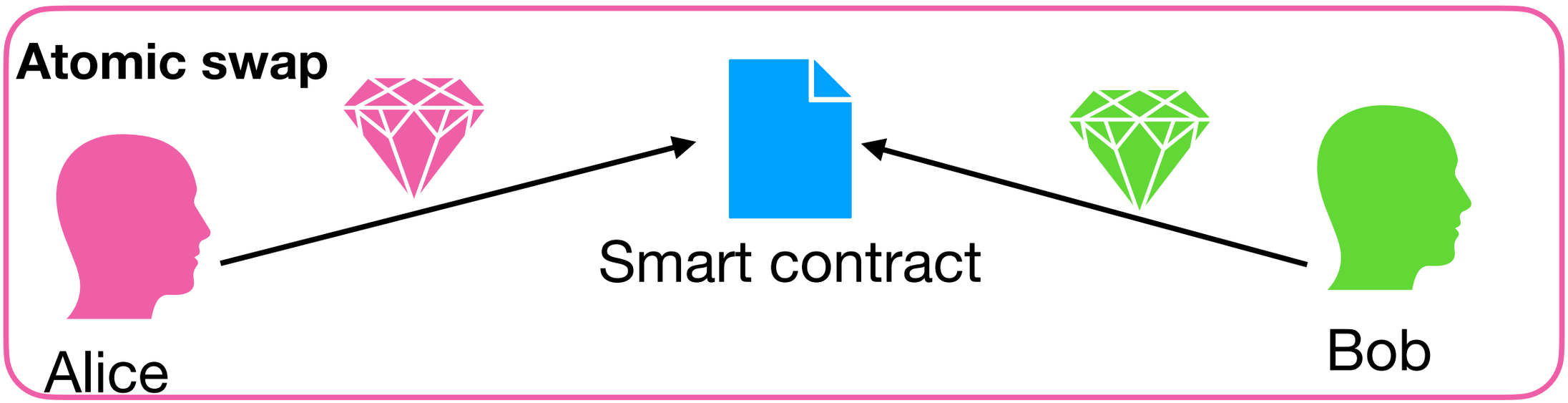
In (standard) CGMs, τ is total and deterministic

Atomic Swap



Van der Meyden. *On the specification and verification of atomic swap smart contracts*. 2019

Atomic Swap

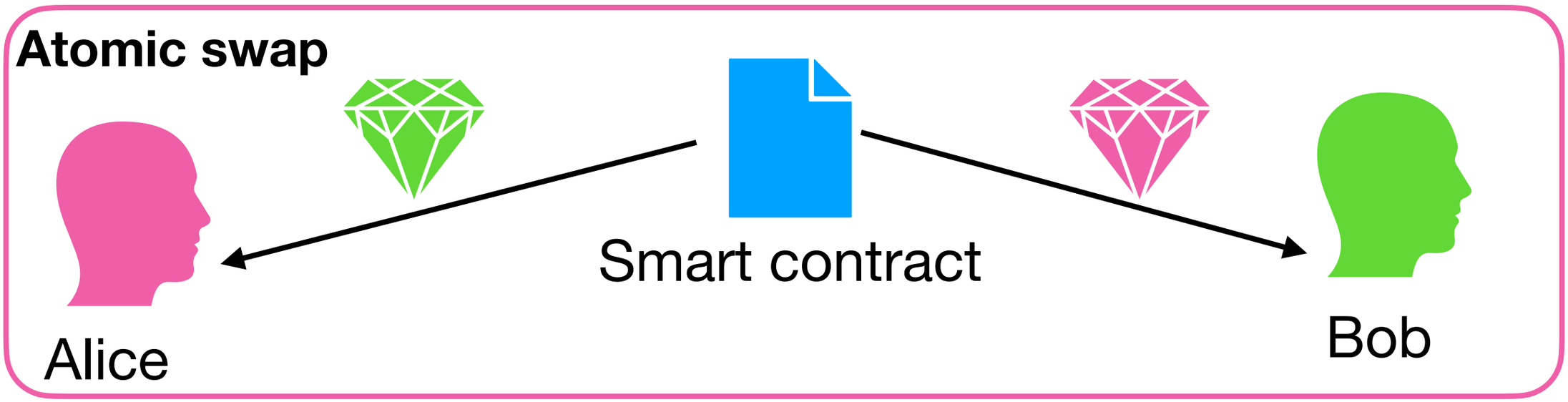


Van der Meyden. *On the specification and verification of atomic swap smart contracts*. 2019

Atomic Swap

Van der Meyden. *On the specification and verification of atomic swap smart contracts*. 2019

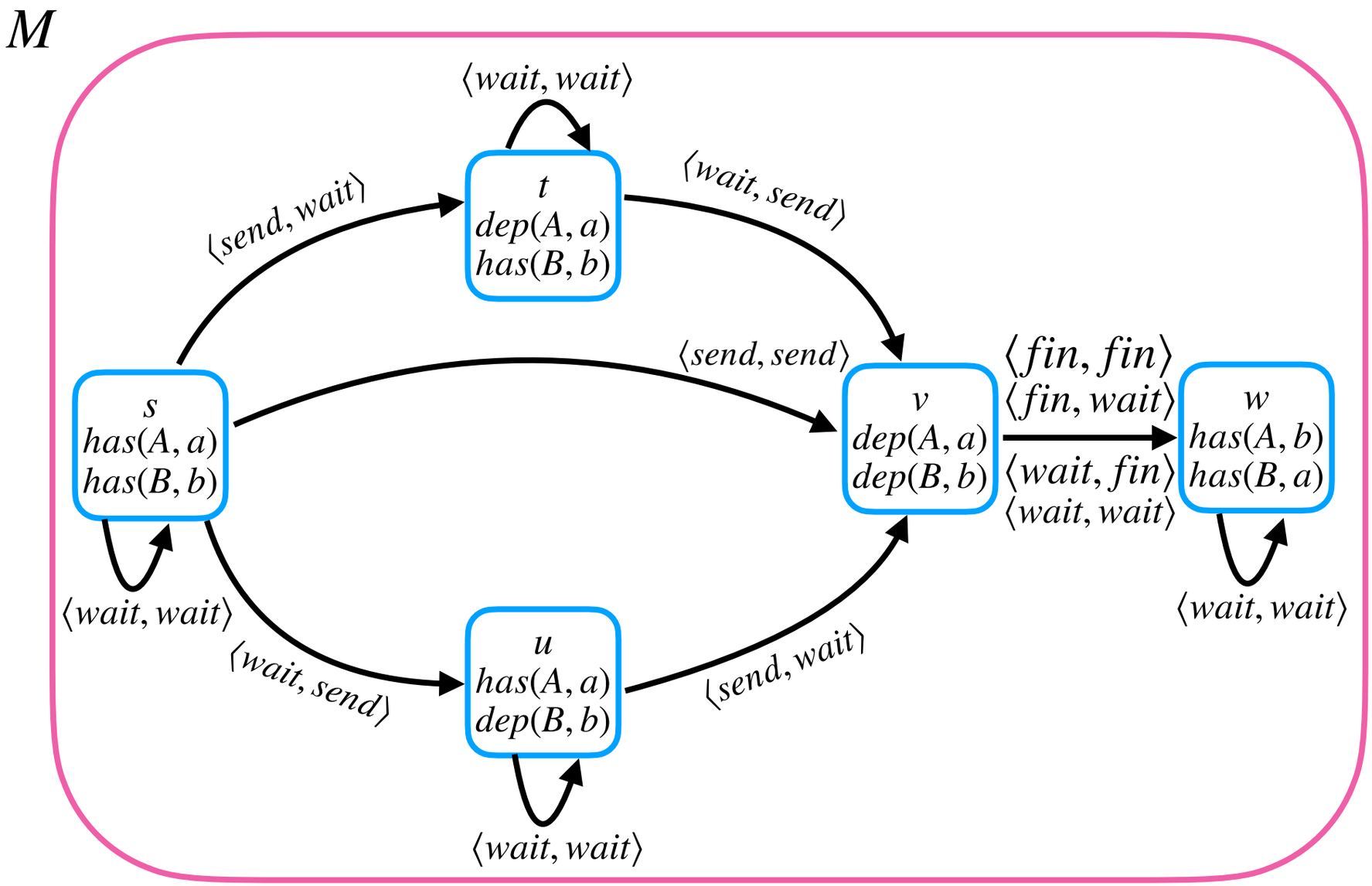
Atomic Swap



Van der Meyden. *On the specification and verification of atomic swap smart contracts*. 2019

Strategy Logics

We focus on logics interpreted on Concurrent Game Models (CGMs) like Coalition Logic, ATL, etc



Strategy Logics

$\langle\langle C \rangle\rangle \varphi$: coalition C **has** a strategy to ensure φ **no matter what** agents outside of the coalition do

$[\![C]\!] \varphi$: **whatever** coalition C does, agents outside of the coalition **have** a strategy to ensure φ

Functionality. Authorised or honest agents can achieve their goals

Security. Unauthorised or malicious agents can not achieve undesirable results

Strategy Logics

ATL $\ni \varphi, \psi := p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \langle\langle C \rangle\rangle \mathsf{X}\varphi \mid \langle\langle C \rangle\rangle \varphi \mathsf{U} \psi \mid \langle\langle C \rangle\rangle \varphi \mathsf{R} \psi$

CL $\ni \varphi := p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \langle\langle C \rangle\rangle \mathsf{X}\varphi$

A (memoryless) **strategy** is a function $\sigma_C : S \to Act^C$ such that $\sigma_i \in act(s, i)$ for all $i \in C$ and $s \in S$. Given a CGM M, state s_0, and a strategy σ_A, the **outcome** $out(s_0, \sigma_A)$ is a sequence $s_0, s_1, s_2, \ldots$, where for each $t \in \mathbb{N}$,

$$s_{t+1} = \tau(s_t, \sigma_1(s_t), \ldots, \sigma_n(s_t)).$$

Strategy Logics

ATL $\ni \varphi := p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \langle\langle C \rangle\rangle \mathsf{X}\varphi \mid \langle\langle C \rangle\rangle \varphi \mathsf{U} \psi \mid \langle\langle C \rangle\rangle \varphi \mathsf{R} \psi$

CL $\ni \varphi := p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \langle\langle C \rangle\rangle \mathsf{X}\varphi$

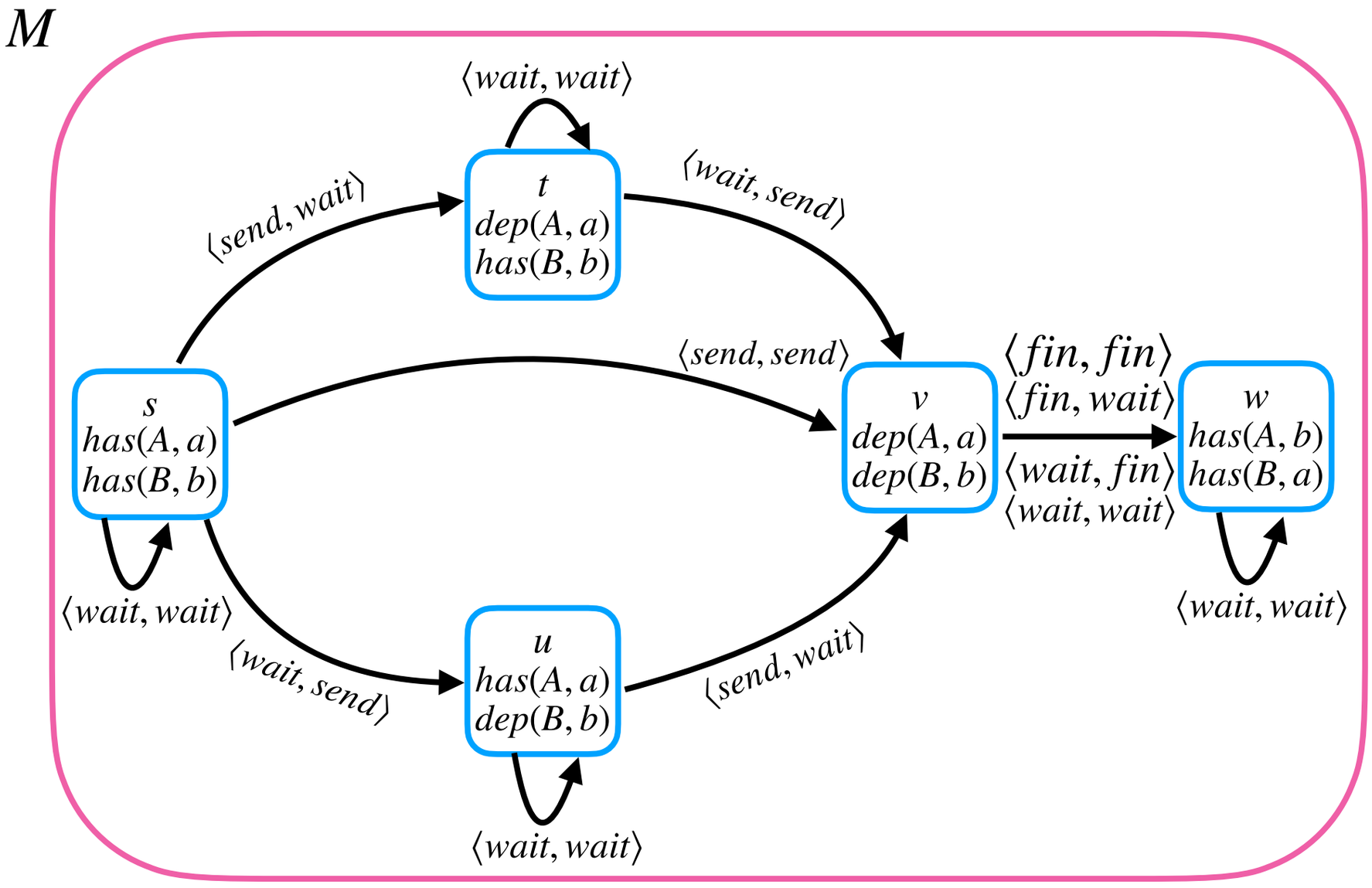
$M, s \vDash \langle\langle C \rangle\rangle \mathsf{X}\varphi$ iff $\exists \sigma_C, \forall \sigma_{\overline{C}} : M, out_1(s, \sigma_C \cup \sigma_{\overline{C}}) \vDash \varphi$

$M, s \vDash \langle\langle C \rangle\rangle \psi \mathsf{U} \varphi$ iff

$\exists \sigma_C, \forall \sigma_{\overline{C}}, \exists k \in \mathbb{N} : M, out_k(s, \sigma_C \cup \sigma_{\overline{C}}) \vDash \varphi$ and

$M, out_m(s, \sigma_C \cup \sigma_{\overline{C}}) \vDash \psi$ for all $m < k$

Strategy Logics



$$M, s \models \neg\langle\langle A \rangle\rangle \mathsf{X}(dep(A, a) \wedge dep(B, b))$$

$$M, s \models \langle\langle A, B \rangle\rangle \mathsf{X}(dep(A, a) \wedge dep(B, b))$$

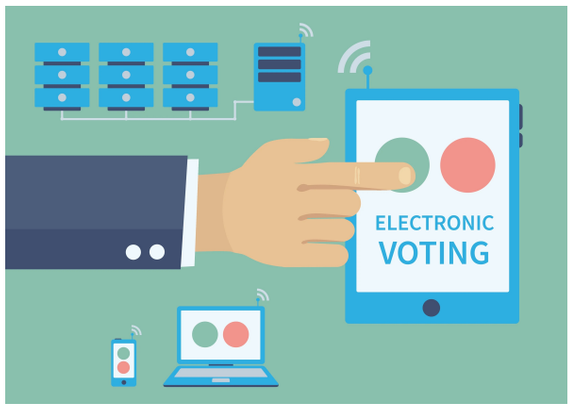
$$M, s \models [\![A]\!]\Big(\neg has(A, b) \mathsf{U} (has(A, b) \wedge has(B, a))\Big)$$

Strategy Logics

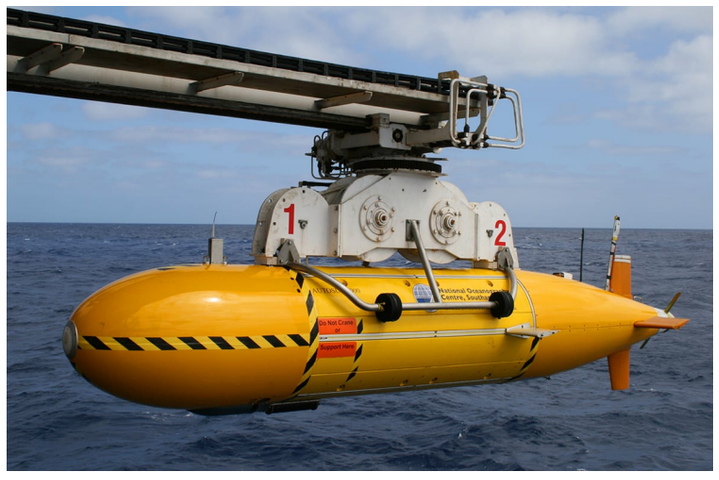
We focus on logics interpreted on Concurrent Game Models (CGMs) like Coalition Logic, ATL, etc

Functionality. Authorised or honest agents can achieve their goals

Security. Unauthorised or malicious agents can not achieve undesirable results



SELENE e-voting



National Oceanography Centre

Ezeikiel et al. *Verifying fault tolerance and self-diagnosability of an autonomous underwater vehicle*. 2011

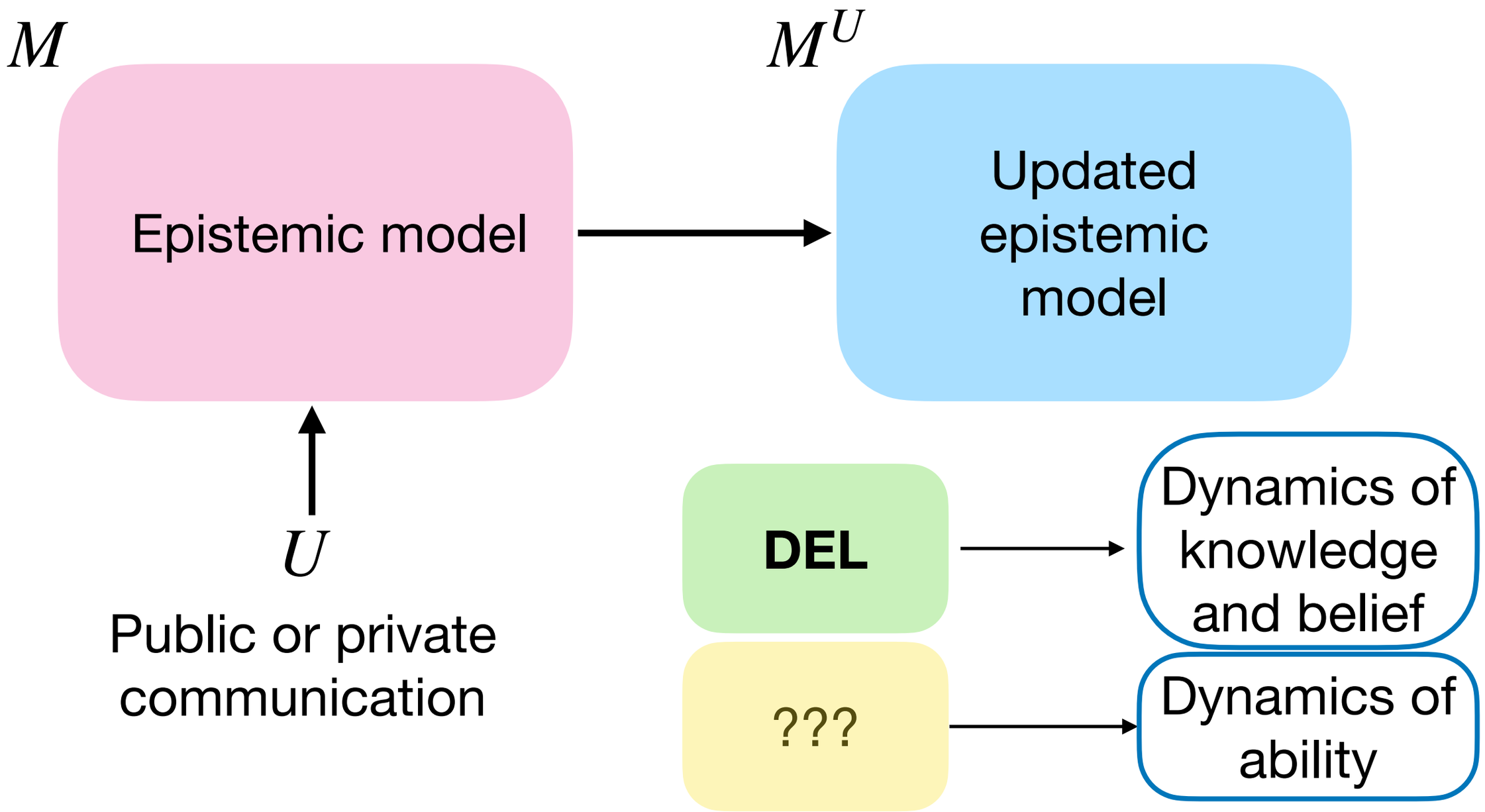
Jamroga et al. *Model Checking the SELENE E-Voting Protocol in Multi-agent Logics*. 2018

The Dynamic Turn

(dynamics as in model transformations)

Dynamics?

Dynamics as in Dynamic Epistemic Logic (DEL), i.e. model transformations



Dynamics in MAS

Normative reasoning: the set of actions is divided into desirable\undesirable

Obstruction logics: a special agent (Demon) can switch off some transitions to prevent an attacker reaching crucial points of a system

Our goal: unifying general approach to reasoning about dynamic phenomena in MAS

Two levels of model dynamics: updates coming ‘from outside’ and updates done by agents in the system

Dynamics in MAS

Normative reasoning: the set of actions is divided into desirable\undesirable

Obstruction logics: a special agent (Demon) can switch off some transitions to prevent an attacker reaching crucial points of a system

Our goal: unifying general approach to reasoning about dynamic phenomena in MAS

$$[U]\varphi$$

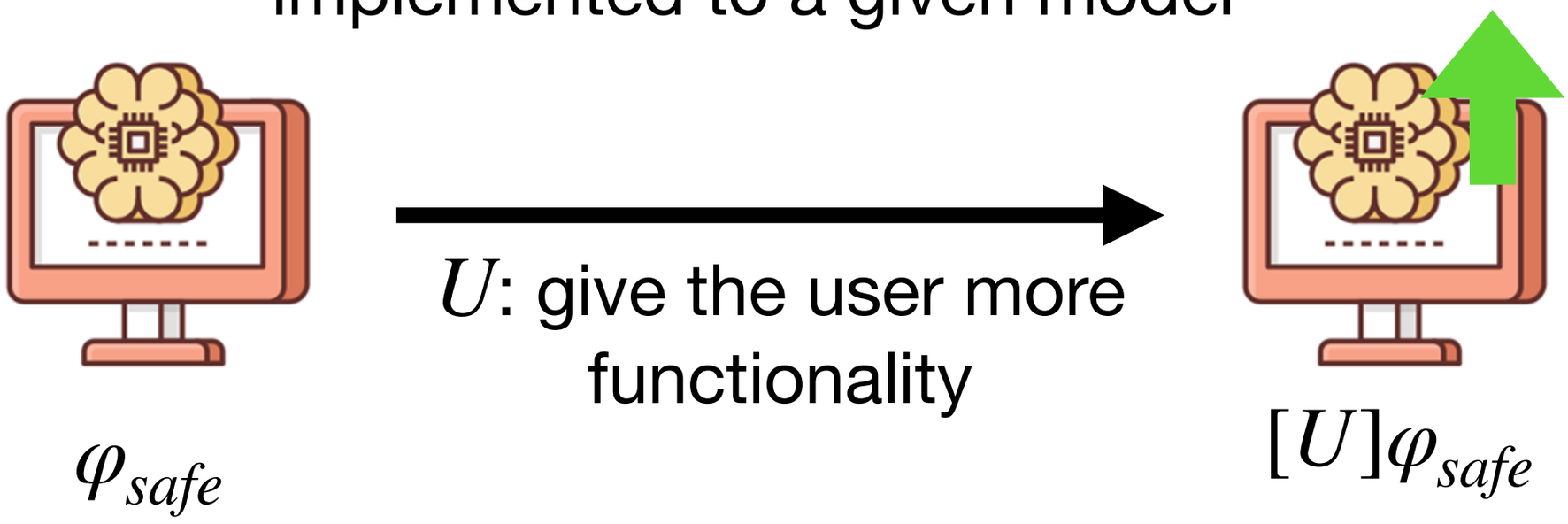
‘After implementing update\upgrade\change U, φ will hold in the resulting updated\upgraded\changed model’

Dynamics in MAS

$$[U]\varphi$$

‘After implementing update\upgrade\change U, φ will hold in the resulting updated\upgraded\changed model’

U is a part of a language, and encapsulates changes to be implemented to a given model

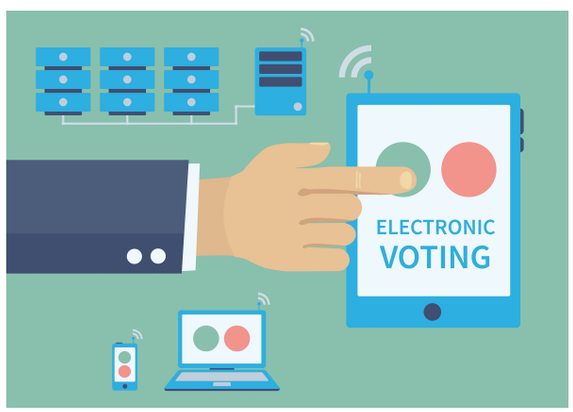


Dynamics in MAS

$$[U]\varphi$$

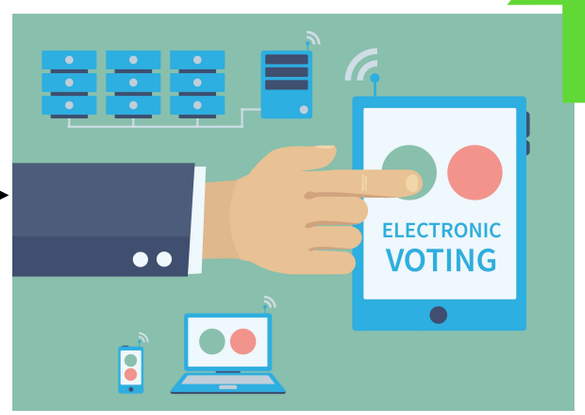
‘After implementing update\upgrade\change U, φ will hold in the resulting updated\upgraded\changed model’

U is a part of a language, and encapsulates changes to be implemented to a given model



$\varphi_{no\_coercion}$

U: switch from single-winner to multiple-winner



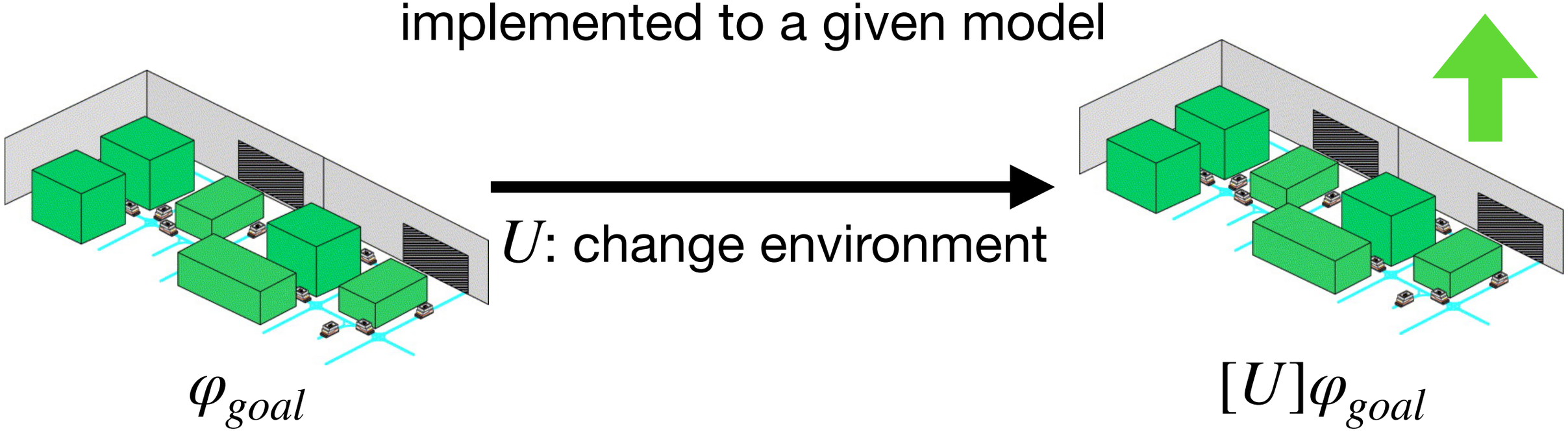
$[U]\varphi_{no\_coercion}$

Dynamics in MAS

$$[U]\varphi$$

‘After implementing update\upgrade\change U, φ will hold in the resulting updated\upgraded\changed model’

U is a part of a language, and encapsulates changes to be implemented to a given model



$$[U]\varphi$$

High-level description of changes: as updates U are part of a language, they represent a succinct high-level representation of an update, as opposed to a low-level description of particular changes in a model

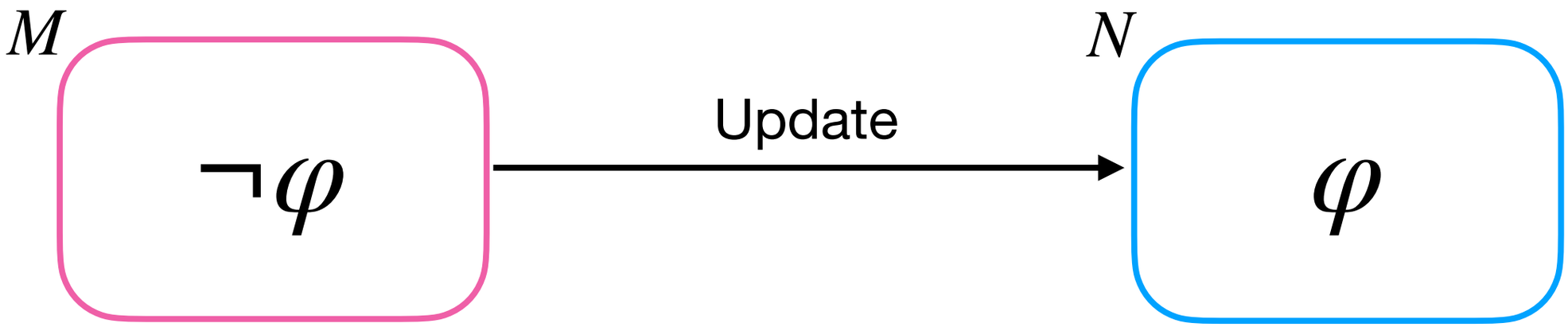
Universality: updates, as logical formulas, are model agnostic (for the same class of models)

Automated checking: checking whether φ holds after U can be done automatically on a computer, instead of implementing all changes in a model by hand

Challenges

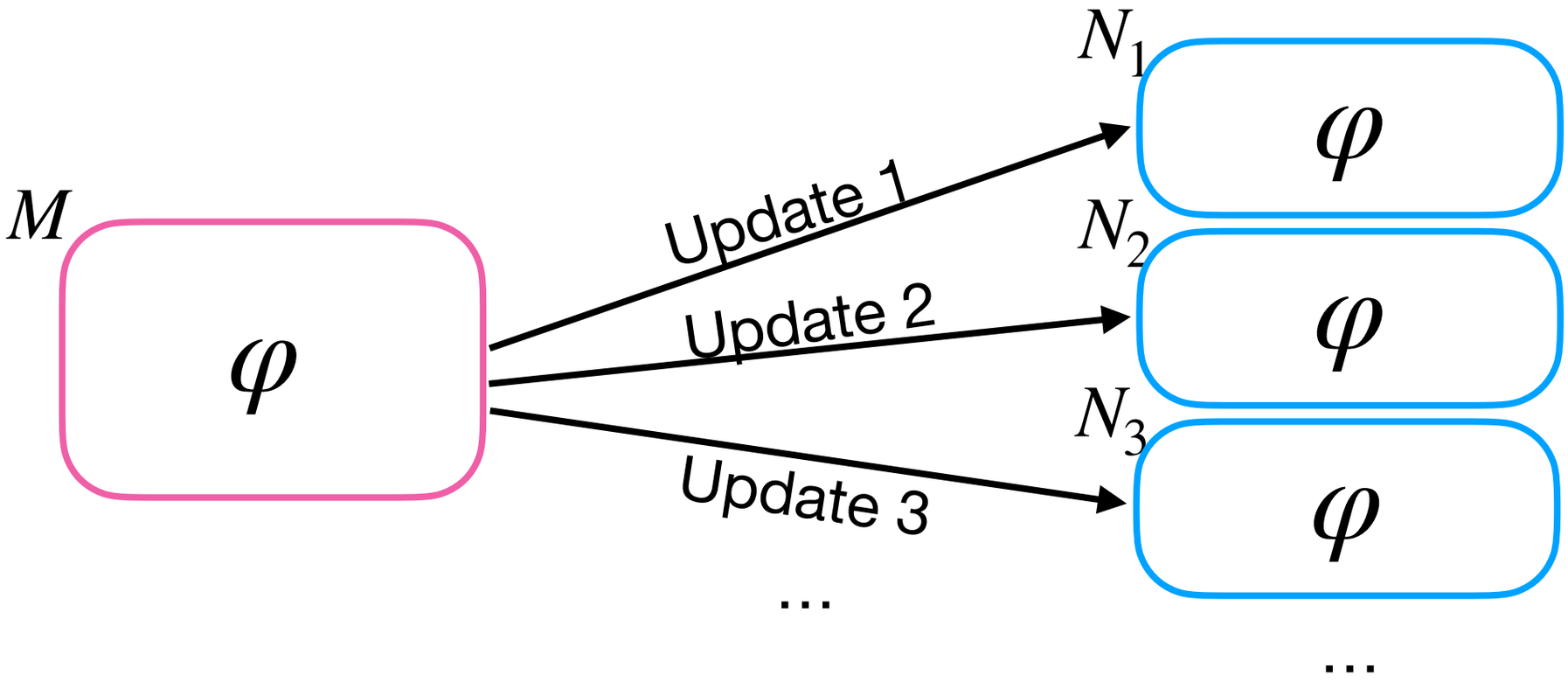
Reasoning about the dynamics of ability: propose meaningful and succinct ways of updating models (changing strategies, environment, abilities, etc); study expressivity and computational properties of new formalisms; add quantification over updates

Reasoning about the dynamics of ability: propose meaningful and succinct ways of updating models (changing strategies, environment, abilities, etc); study expressivity and computational properties of new formalisms; add quantification over updates



Functionality: is there an allowed\possible\enable update of the system to make it satisfy the goal formula?

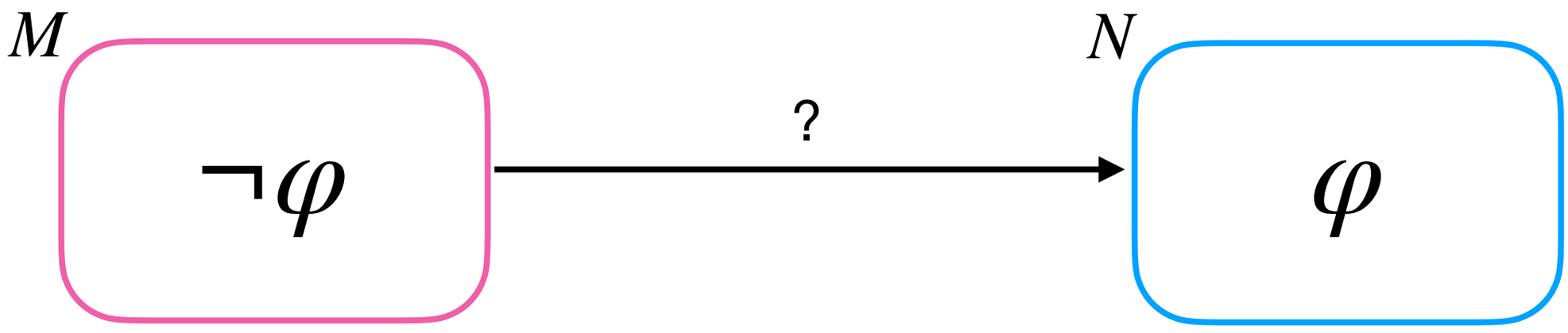
Reasoning about the dynamics of ability: propose meaningful and succinct ways of updating models (changing strategies, environment, abilities, etc); study expressivity and computational properties of new formalisms; add quantification over updates



Safety\resilience: none of the allowed\possible\enabled updates will lead to an unsafe model

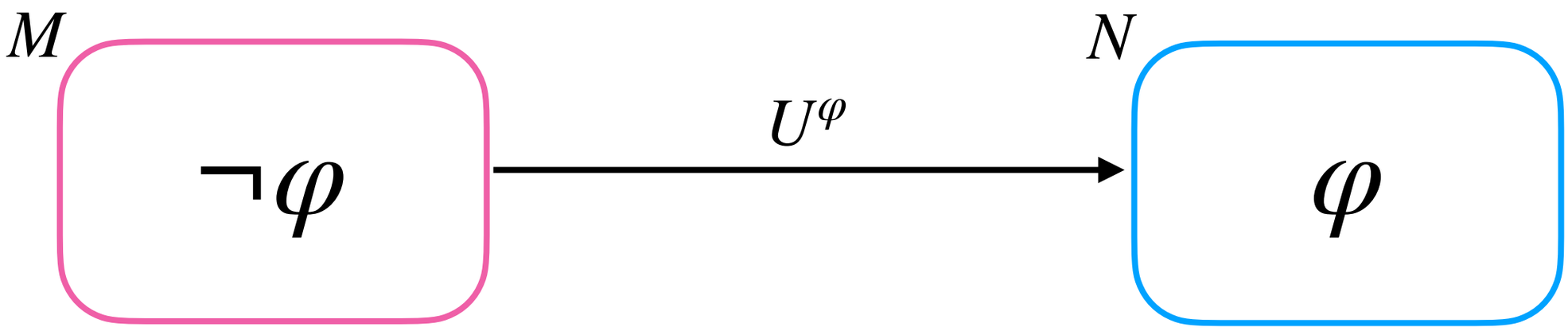
Reasoning about the dynamics of ability: propose meaningful and succinct ways of updating models (changing strategies, environment, abilities, etc); study expressivity and computational properties of new formalisms; add quantification over updates

Automated synthesis\Model repair: given a starting model and a target property φ, construct an update that transforms the model so that the target formula is satisfied; provide minimal synthesis disrupting a model as little as possible



Reasoning about the dynamics of ability: propose meaningful and succinct ways of updating models (changing strategies, environment, abilities, etc); study expressivity and computational properties of new formalisms; add quantification over updates

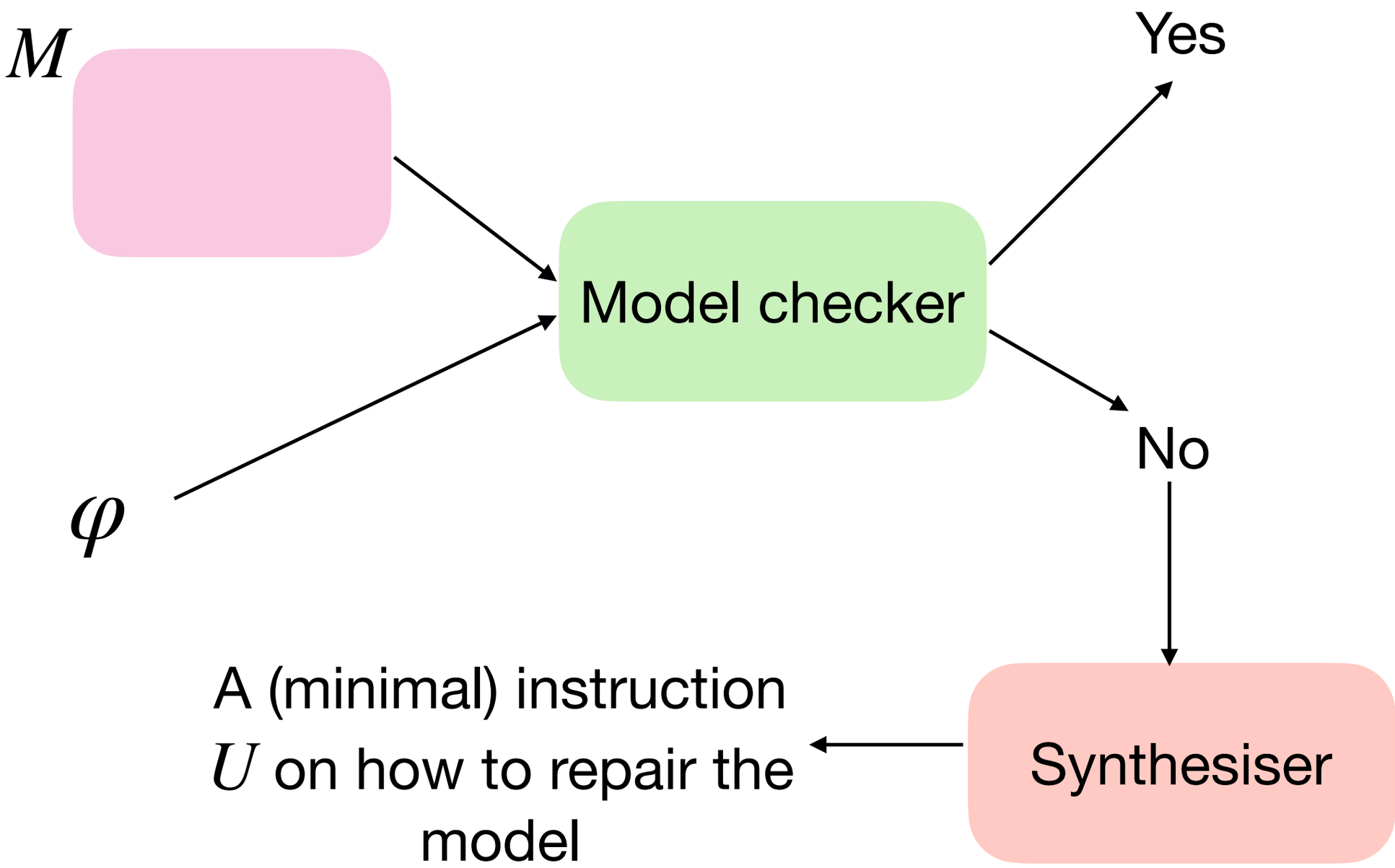
Automated synthesis\Model repair: given a starting model and a target property φ, construct an update that transforms the model so that the target formula is satisfied; provide minimal synthesis disrupting a model as little as possible



Reasoning about the dynamics of ability: propose meaningful and succinct ways of updating models (changing strategies, environment, abilities, etc); study expressivity and computational properties of new formalisms; add quantification over updates

Automated synthesis\Model repair: given a starting model and a target property φ, construct an update that transforms the model so that the target formula is satisfied; provide minimal synthesis disrupting a model as little as possible

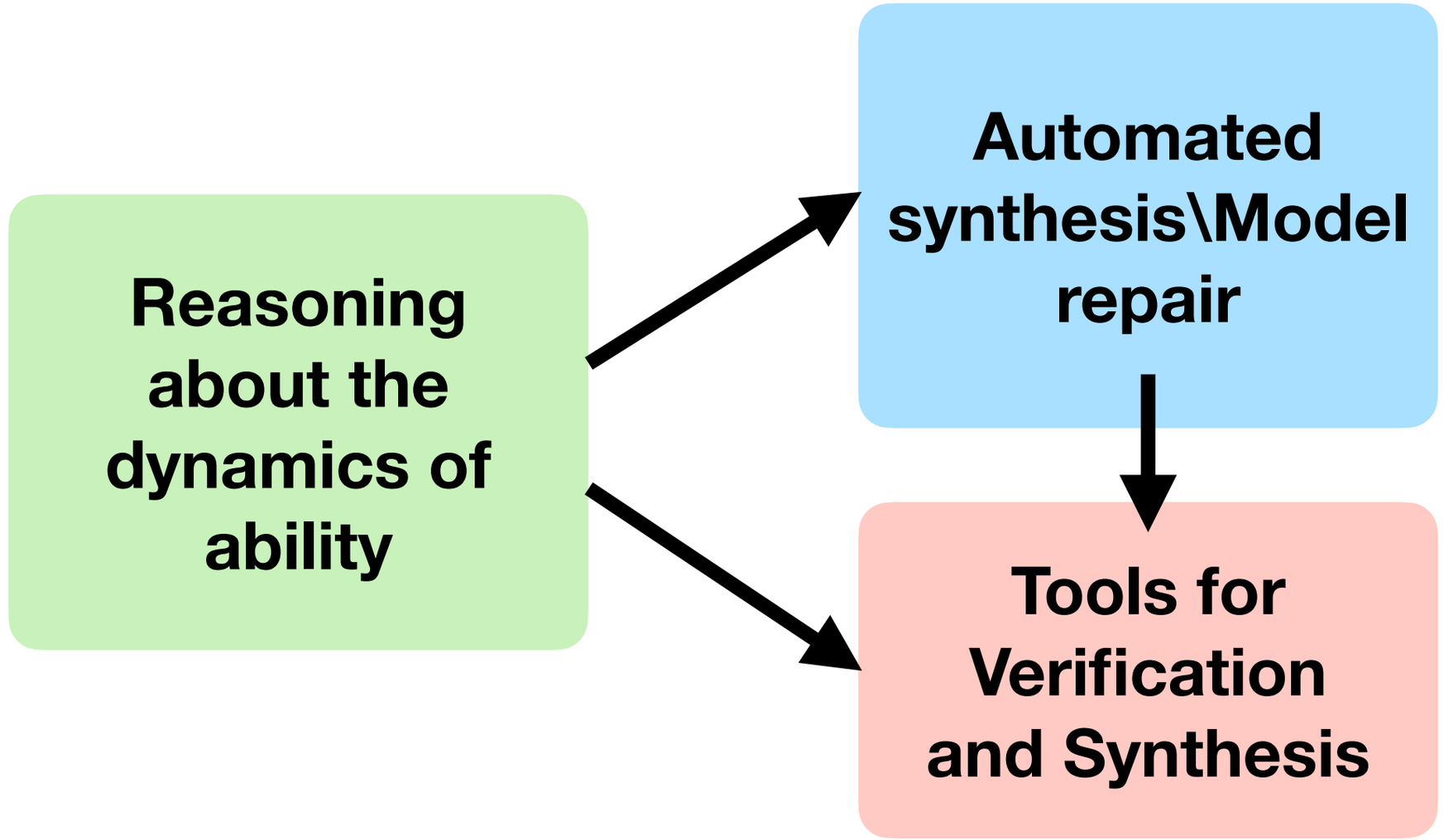
Tools for Verification and Synthesis: create extensions of established model checkers (MCMAS, STV, VITAMIN) for new logics; incorporate automated synthesis



Tools for Verification and Synthesis: create extensions of established model checkers (MCMAS, STV, VITAMIN) for new logics; incorporate automated synthesis

The Dynamic Turn in Strategy Logics

Our goal: unifying general approach to reasoning about dynamic phenomena in MAS



Dynamics of Individual and Group Ability

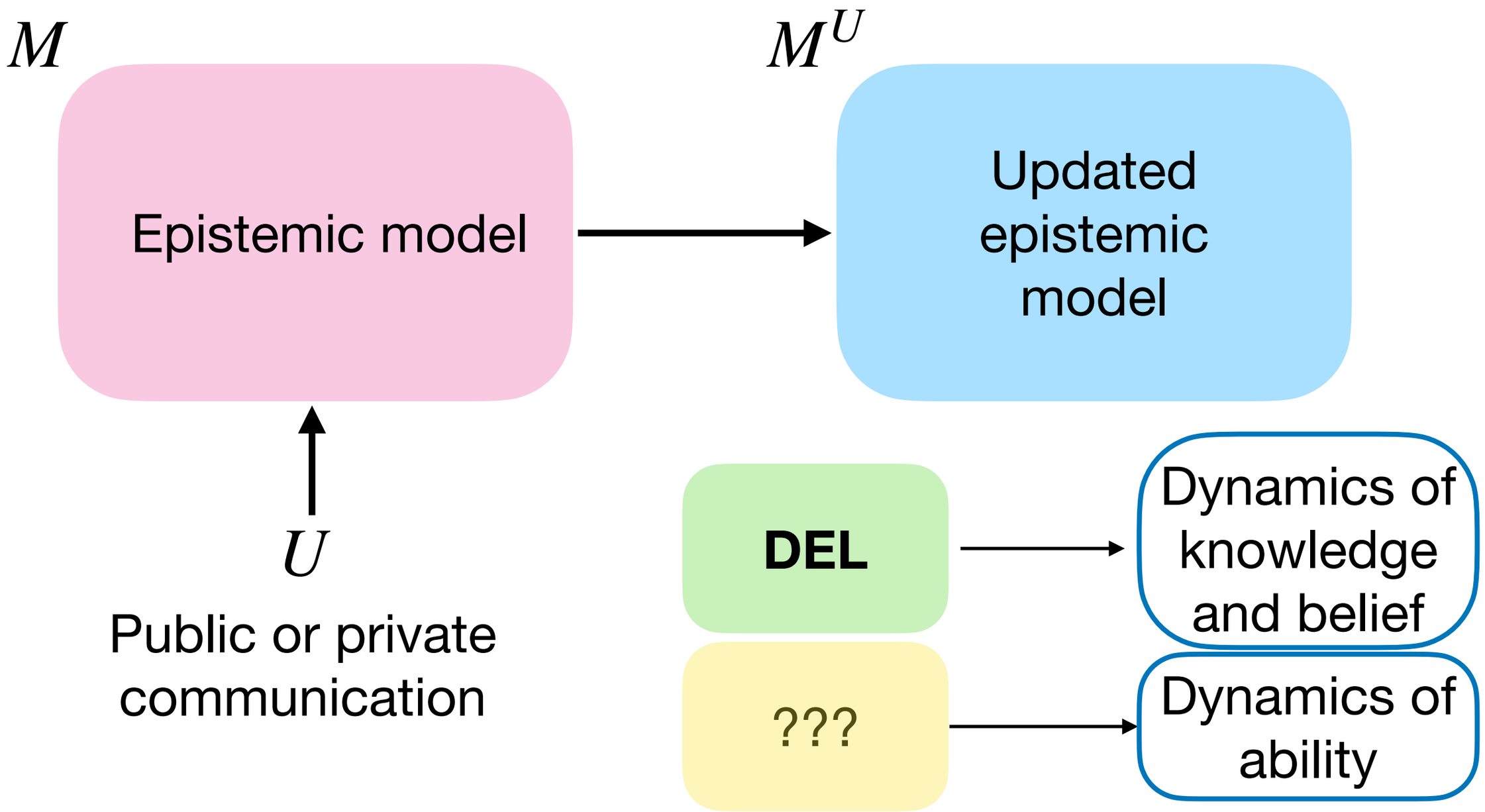
Dynamic Coalition Logic: Granting and Revoking Dictatorial Powers

Rustam Galimullin[1] and Thomas Ågotnes[1,2]

[1] University of Bergen, Bergen, Norway
{rustam.galimullin, thomas.agotnes}@uib.no
[2] Southwest University, Chongqing, China

Dynamics?

Dynamics as in Dynamic Epistemic Logic (DEL), i.e. model transformations



Ability

In the setting of CL, we treat ability as a variant of **know-how**: every time a precondition is satisfied, a given agent can achieve the desired outcome

For a single agent, having an ability means to have a **dictatorial power**

$$[U]\varphi$$

$$\downarrow$$

$$[\chi, 1, \psi]\varphi$$

‘Grant agent 1 the ability to force ψ-states from any χ-state’

Nuance

$$[\chi, 1, \psi]\varphi$$

‘Grant agent a the ability to force ψ-states from any χ-state’

Do we grant the power to force **every** ψ-state, or **some** ψ-states? (Which ones?)

We go with the first option

Do we grant only **one power** to an agent or **several powers** at the same time? Do we grant powers to agents **one at a time**, or do it for **several agents** simultaneously?

We take a general stance and allow granting multiple powers to multiple single agents at the same time

Nuance

$$[(\chi_1,1,\psi_1),(\chi_2,2,\psi_2),(\tau_1,1,\xi_1)]\varphi$$

We also do similarly for **revoking** dictatorial powers

Nuance

$$[(\chi_1,1,\psi_1),(\chi_2,2,\psi_2),(\tau_1,1,\xi_1)]^{+}\varphi$$

We also do similarly for **revoking** dictatorial powers

$$[(\chi_1,1,\psi_1),(\chi_2,2,\psi_2),(\tau_1,1,\xi_1)]^{-}\varphi$$

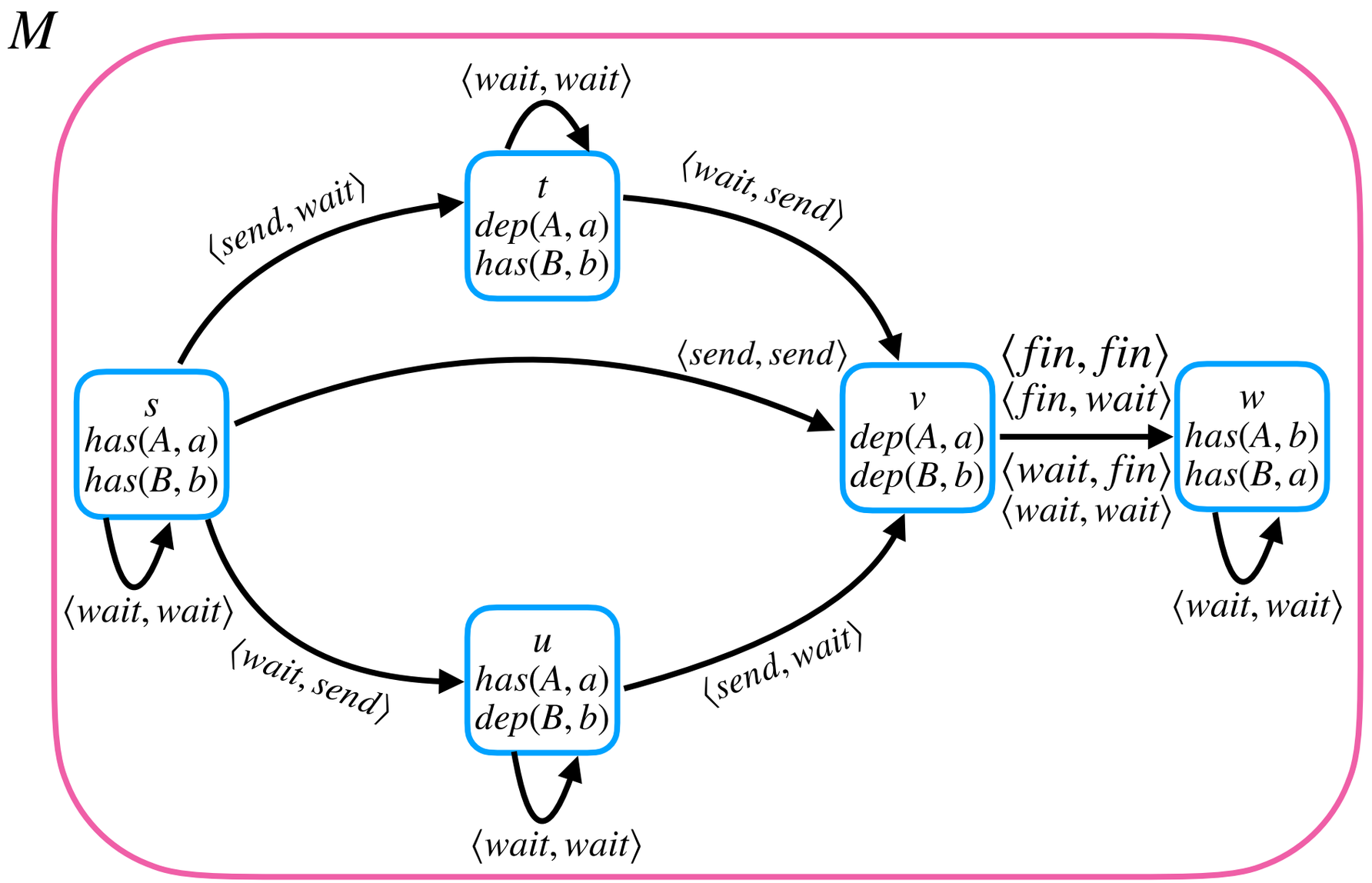
Dictatorial Dynamic CL

$$\mathsf{DDCL}^{+} \ni \varphi := p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \langle\langle C \rangle\rangle \mathsf{X}\varphi \mid [+U]\varphi$$
$$+U := (\varphi, i, \varphi) \mid (\varphi, i, \varphi), +U$$

$$\mathsf{DDCL}^{-} \ni \varphi := p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \langle\langle C \rangle\rangle \mathsf{X}\varphi \mid [-U]\varphi$$
$$-U := (\varphi, i, \varphi) \mid (\varphi, i, \varphi), -U$$

Positive updates $[+U]\varphi$**:** after implementing the positive update $+U$, φ holds in the updated model

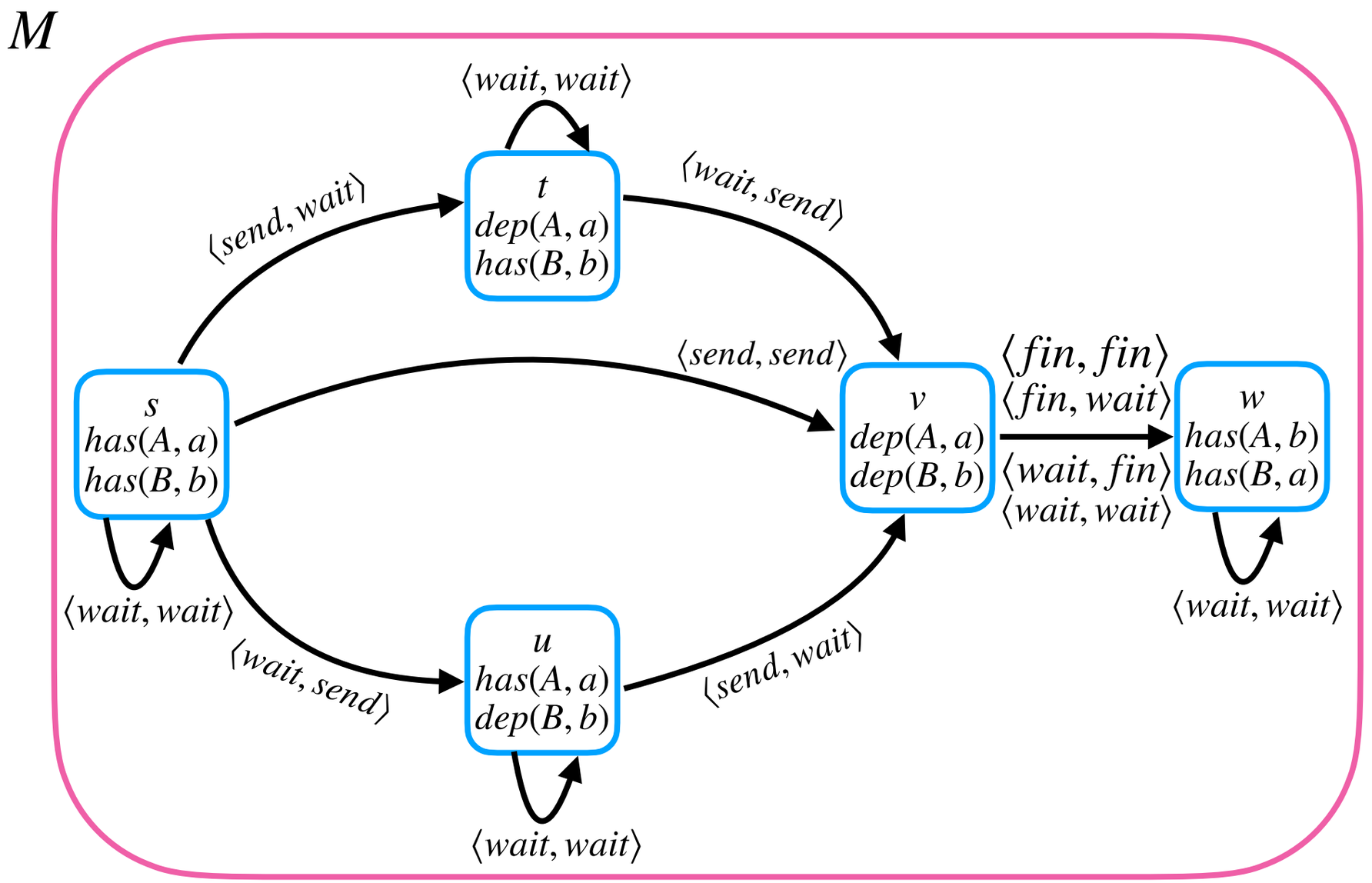
Negative updates $[-U]\varphi$**:** after implementing the negative update $-U$, φ holds in the updated model



$$M \models dep(A, a) \rightarrow [\![A, B]\!] \mathsf{X} \neg has(A, a)$$

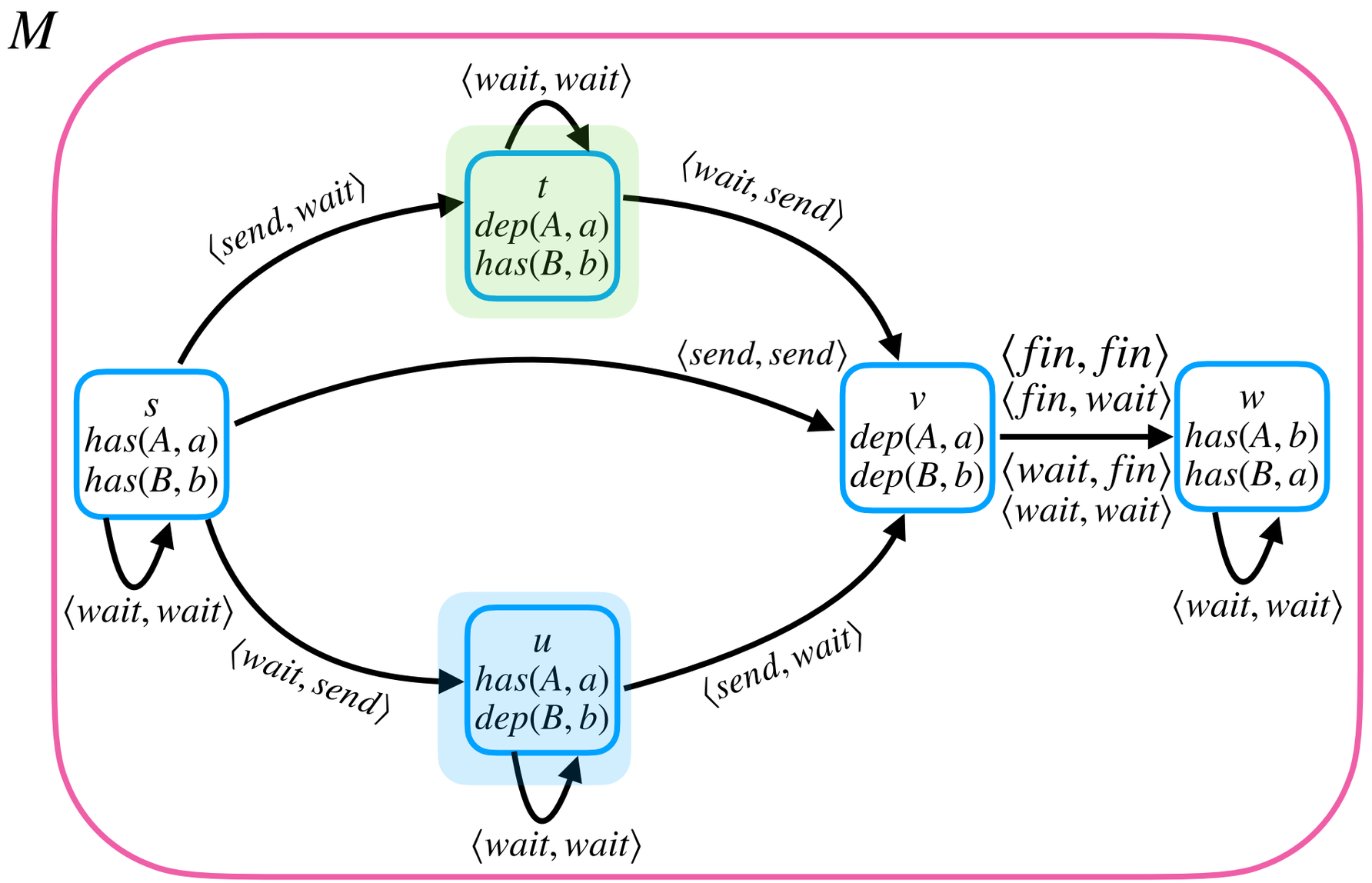
If an agent deposits her asset, she cannot get it back even if the swapped is not finalised

Let us give an option to agents to opt out of the swap…



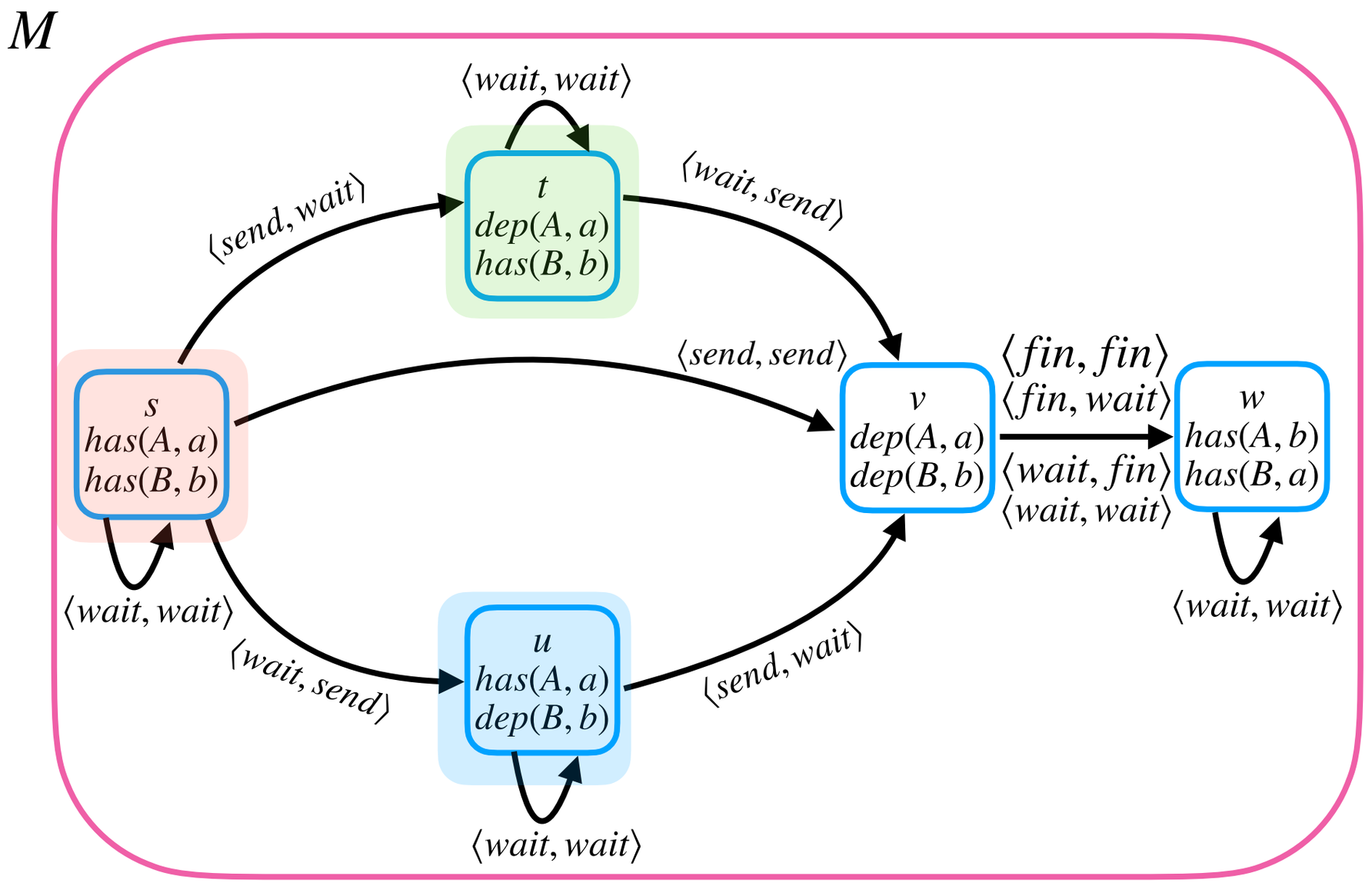
Let us give an option to agents to opt out of the swap…

$$+U = \Big(d(A,a) \wedge h(B,b), A, h(A,a) \wedge h(B,b)\Big), \Big(d(B,b) \wedge h(A,a), B, h(A,a) \wedge h(B,b)\Big)$$



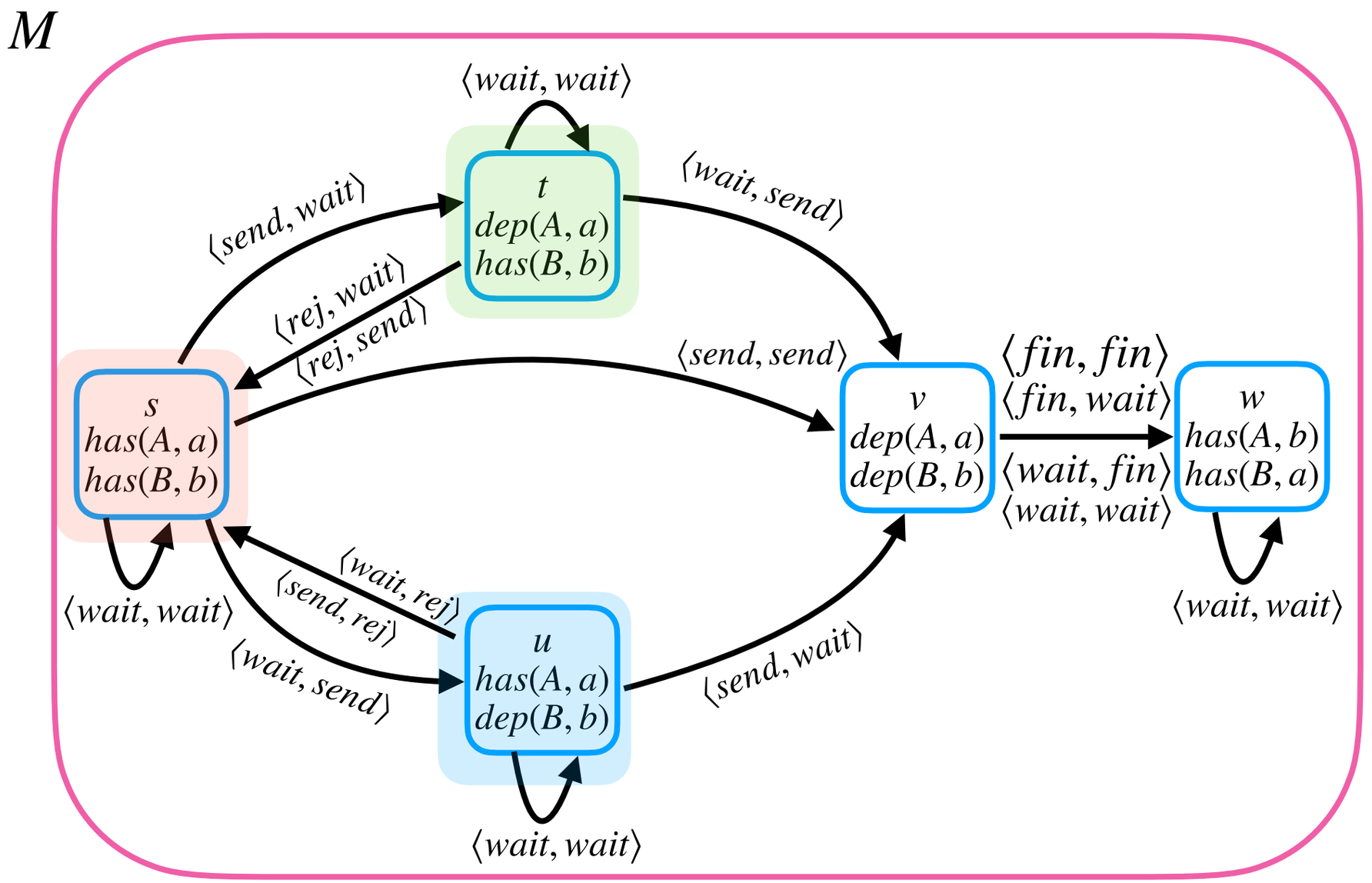
Let us give an option to agents to opt out of the swap…

$$+U = \Big(d(A,a) \wedge h(B,b), A, h(A,a) \wedge h(B,b)\Big), \Big(d(B,b) \wedge h(A,a), B, h(A,a) \wedge h(B,b)\Big)$$



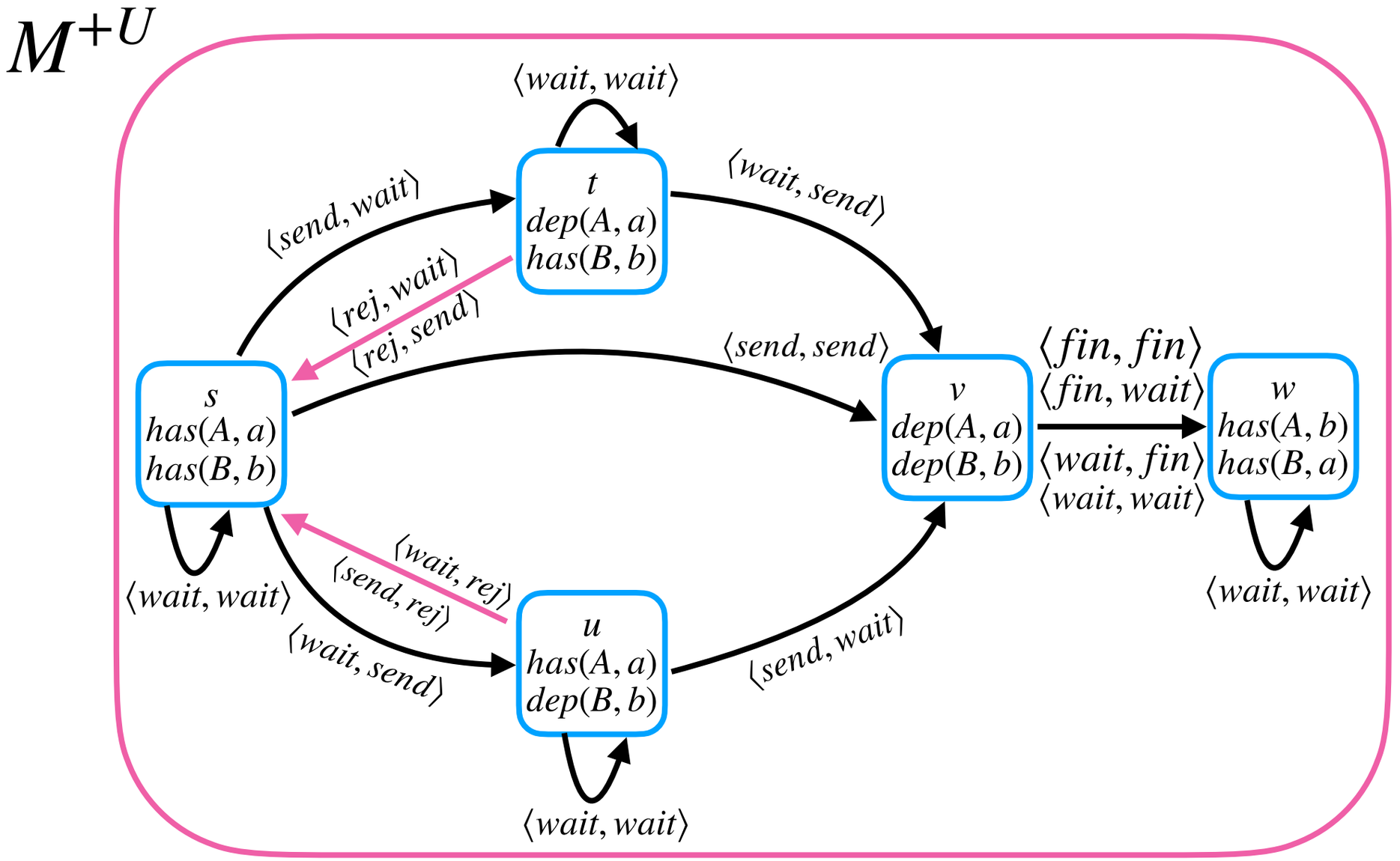
Let us give an option to agents to opt out of the swap…

$$+U = \Big(d(A,a) \wedge h(B,b), A, h(A,a) \wedge h(B,b)\Big), \Big(d(B,b) \wedge h(A,a), B, h(A,a) \wedge h(B,b)\Big)$$



Let us give an option to agents to opt out of the swap…

$$+U = \Big(d(A,a) \wedge h(B,b), A, h(A,a) \wedge h(B,b)\Big), \Big(d(B,b) \wedge h(A,a), B, h(A,a) \wedge h(B,b)\Big)$$



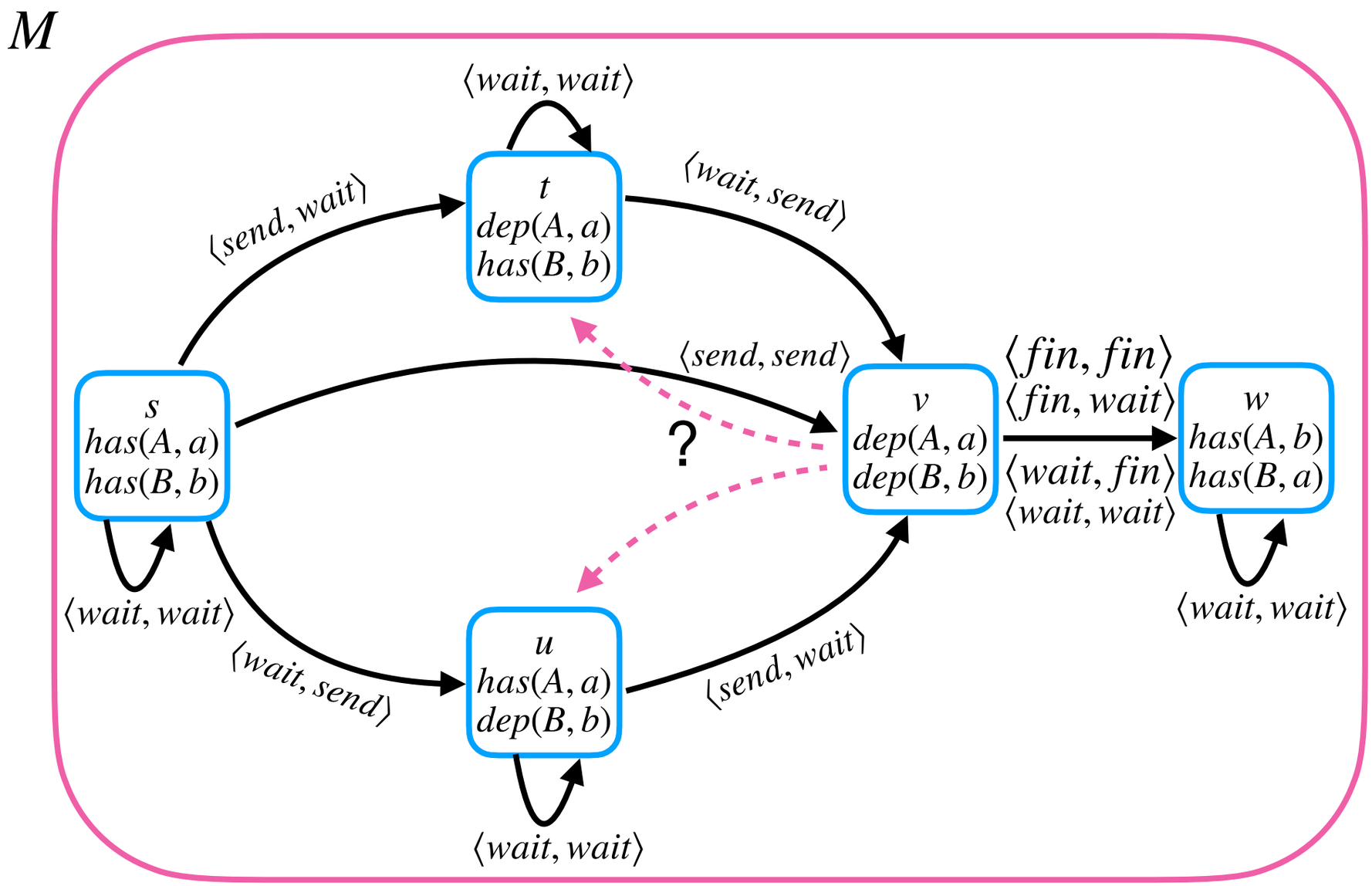
Let us give an option to agents to opt out of the swap…

$$+U = \big(d(A,a) \wedge h(B,b), A, h(A,a) \wedge h(B,b)\big), \big(d(B,b) \wedge h(A,a), B, h(A,a) \wedge h(B,b)\big)$$

$$M \models dep(A,a) \rightarrow [\![A,B]\!]\mathsf{X} \neg has(A,a)$$

$$M^{+U} \not\models dep(A,a) \rightarrow [\![A,B]\!]\mathsf{X} \neg has(A,a)$$

$$M \models [+U]\big(dep(A,a) \wedge h(B,b) \rightarrow \langle\!\langle A \rangle\!\rangle \mathsf{X} has(A,a)\big)$$



However, not every update can be implemented

$$+U = \big(d(A,a) \wedge d(B,b), A, h(A,a) \wedge d(B,b)\big), \big(d(A,a) \wedge d(B,b), B, d(A,a) \wedge h(B,b)\big)$$

Update $+U$ forces us to have two dictators in a single state and both of them force different states

No priority on agents; update $+U$ is not executable

Executability

We call update $+U$ **executable** in CGM M iff for all (φ, i, ψ), $(\chi, j, \tau) \in +U$: $\|\varphi\|_M \cap \|\chi\|_M = \varnothing$ whenever $i \neq j$

$Pairs^{+U}_{(\varphi,i,\psi)} = \{s \mid M, s \vDash \varphi\} \times \{s \mid M, s \vDash \psi\}$ is the set of pairs between which we need to add forcing transitions according to $(\varphi, i, \psi) \in +U$. $Force(i, s)$ contains new actions $a^{(s,t)}$ for each $(s, t) \in Pairs^{+U}_{(\varphi,i,\psi)}$

Given CGM M and update $+U$, and updated CGM M^{+U} is $(Act^{+U}, S, act^{+U}, \tau, {}^{+U}V)$, where

$$Act^{+U} = Act \cup \bigcup_{i \in A, s \in S} Force(i, s)$$

Executability

We call update $+U$ **executable** in CGM M iff for all (φ, i, ψ), $(\chi, j, \tau) \in +U$: $\|\varphi\|_M \cap \|\chi\|_M = \varnothing$ whenever $i \neq j$

$Pairs^{+U}_{(\varphi,i,\psi)} = \{s \mid M, s \vDash \varphi\} \times \{s \mid M, s \vDash \psi\}$ is the set of pairs between which we need to add forcing transitions according to $(\varphi, i, \psi) \in +U$. $Force(i, s)$ contains new actions $a^{(s,t)}$ for each $(s, t) \in Pairs^{+U}_{(\varphi,i,\psi)}$

Given CGM M and update $+U$, and updated CGM M^{+U} is $(Act^{+U}, S, act^{+U}, \tau, {}^{+U}V)$, where

$$act^{+U}(i, s) = act(i, s) \cup Force(i, s)$$

Executability

We call update $+U$ **executable** in CGM M iff for all (φ, i, ψ), $(\chi, j, \tau) \in +U$: $\|\varphi\|_M \cap \|\chi\|_M = \varnothing$ whenever $i \neq j$

$Pairs^{+U}_{(\varphi,i,\psi)} = \{s \mid M, s \vDash \varphi\} \times \{s \mid M, s \vDash \psi\}$ is the set of pairs between which we need to add forcing transitions according to $(\varphi, i, \psi) \in +U$. $Force(i, s)$ contains new actions $a^{(s,t)}$ for each $(s, t) \in Pairs^{+U}_{(\varphi,i,\psi)}$

Given CGM M and update $+U$, and updated CGM M^{+U} is $(Act^{+U}, S, act^{+U}, \tau, {}^{+U}V)$, where

$\tau^{+U}(\alpha, s) = t$, if $\exists \alpha_i^{(s,t)} \sqsubseteq \alpha$ with $\alpha_i^{(s,t)} \in Force(i, s)$, and $\tau(\alpha, s)$ otherwise

Dictatorial Dynamic CL

$$\mathsf{DDCL}^{+} \ni \varphi := p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \langle\langle C \rangle\rangle \mathsf{X}\varphi \mid [+U]\varphi$$
$$+U := (\varphi, i, \varphi) \mid (\varphi, i, \varphi), +U$$

$$\mathsf{DDCL}^{-} \ni \varphi := p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \langle\langle C \rangle\rangle \mathsf{X}\varphi \mid [-U]\varphi$$
$$-U := (\varphi, i, \varphi) \mid (\varphi, i, \varphi), -U$$

$M, s \vDash [+U]\varphi$ iff $+U$ is executable on M implies $M^{+U}, s \vDash \varphi$
$M, s \vDash [-U]\varphi$ iff $-U$ is executable on M implies $M^{-U}, s \vDash \varphi$
$M, s \vDash \langle +U \rangle\varphi$ iff $+U$ is executable on M and $M^{+U}, s \vDash \varphi$

Some properties

Monotonicity: $\langle +U \rangle \varphi \wedge \langle +U \rangle \psi \leftrightarrow \langle +U \rangle (\varphi \wedge \psi)$ is valid

Union: $\langle +U^1 \rangle \varphi \wedge \langle +U^2 \rangle \varphi \rightarrow \langle +U^1 \cup +U^2 \rangle \varphi$ is not valid

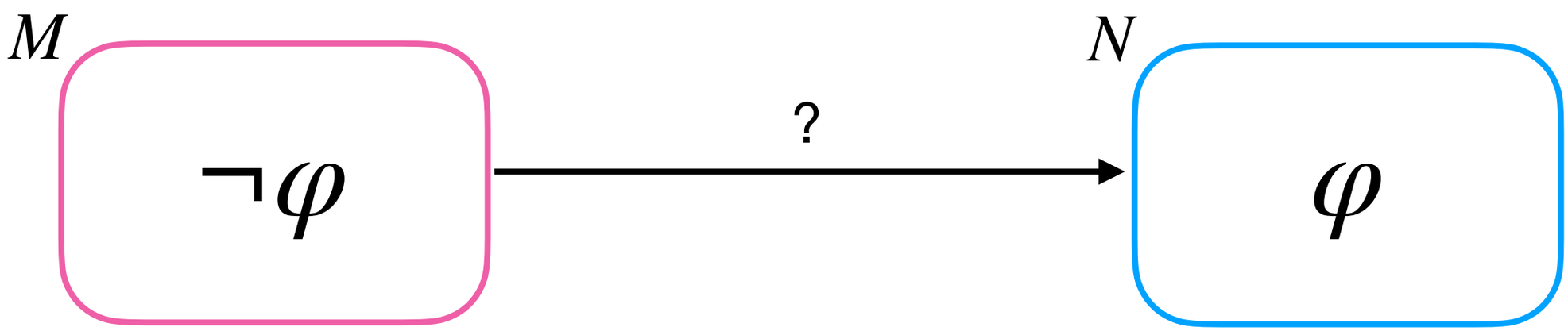
Commutativity: $\langle +U^1 \rangle \langle +U^2 \rangle \varphi \rightarrow \langle +U^2 \rangle \langle +U^1 \rangle \varphi$ is not valid

Decomposition:
$\langle (\psi_1, 1, \chi_1), (\psi_2, 2, \chi_2) \rangle \varphi \rightarrow \langle (\psi_1, 1, \chi_1) \rangle \langle (\psi_2, 2, \chi_2) \rangle \varphi$ is not valid

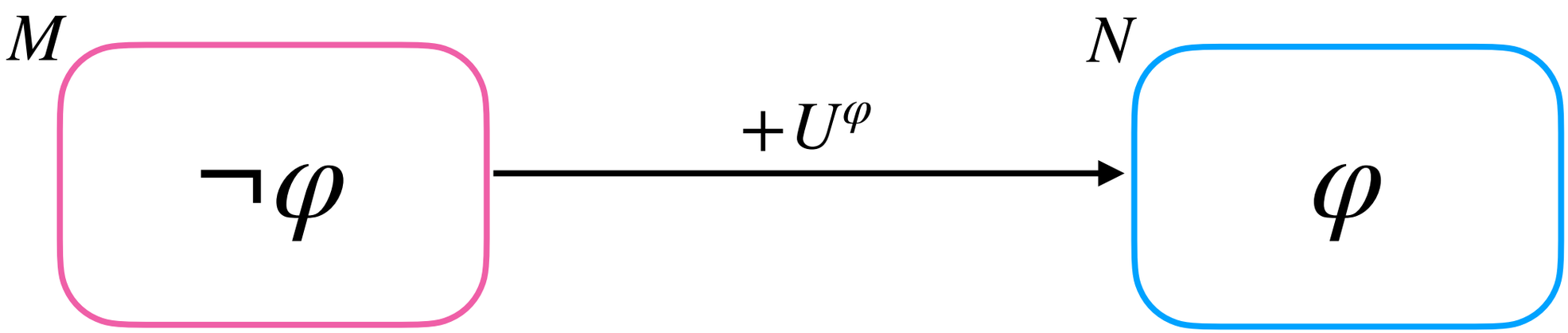
Model checking: P-complete for both DDCL^+ and DDCL^-

Expressivity: both DDCL^+ and DDCL^- are strictly more expressive than CL, are incomparable to ATL, and mutually incomparable

Towards Synthesis



Towards Synthesis



Towards Synthesis

Given a CGM $M = (Act, S, act, \tau, V)$, a binary relation $\mathscr{R} \subseteq S^2$ is a **bisimulation** if for each pair $(s_1, s_2) \in \mathscr{R}$ and for each $C \subseteq A$ we have:

Atoms: for each $p \in P$, $s_1 \in V(p)$ iff $s_2 \in V(p)$

Forth: for each α_C^1 at s_1, there is α_C^2 at s_2 s.t. for every $t_2 \in Out(\alpha_C^2, s_2)$ there exists $t_1 \in Out(\alpha_C^1, s_1)$ s.t. $(t_1, t_2) \in \mathscr{R}$

Back: like **Forth**, with 1 and 2 swapped

Theorem: if $\mathscr{R}(s_1, s_2)$, then $M, s_1 \vDash \varphi$ iff $M, s_2 \vDash \varphi$ with φ being a CL or an ATL formula

Corollary: DDCL is not invariant under bisimulation

Towards Synthesis

Input Given are a CGM M, state s, and a target CL (or ATL) formula φ, such that $M, s \not\models \varphi$

First, we run an algorithm to essentially model check the following quantified expression: $\exists + U : M, s \models [+U]\varphi$

We are in the realm of speculation

Towards Synthesis

Input Given are a CGM M, state s, and a target CL (or ATL) formula φ, such that $M, s \not\models \varphi$

First, we run an algorithm to essentially model check the following quantified expression: $\exists + U : M, s \models [+U]\varphi$

We can guess an update $+U$

We are in the realm of speculation

Towards Synthesis

Input Given are a finite CGM M, state s, and a target CL (or ATL) formula φ, such that $M, s \not\models \varphi$

First, we run an algorithm to essentially model check the following quantified expression: $\exists +U : M, s \models [+U]\varphi$

Second, once we have guessed update $+U$, we need to ensure that it is sufficiently ‘small’, i.e., of polynomial size

Since the model is finite, we can construct formulas that uniquely characterise states of the model (up to bisimulation)

We are in the realm of speculation

Towards Synthesis

Input Given are a CGM M, state s, and a target CL (or ATL) formula φ, such that $M, s \not\models \varphi$

First, we run an algorithm to essentially model check the following quantified expression: $\exists + U : M, s \models [+U]\varphi$

Second, once we have guessed update $+U$, we need to ensure that it is sufficiently ‘small’, i.e., of polynomial size

$$\chi_s^0 := \bigwedge_{s \in V(p)} p \wedge \bigwedge_{s \notin V(p)} \neg p$$

Full propositional description of a state

We are in the realm of speculation

Towards Synthesis

Input Given are a CGM M, state s, and a target CL (or ATL) formula φ, such that $M, s \not\models \varphi$

First, we run an algorithm to essentially model check the following quantified expression: $\exists + U : M, s \models [+U]\varphi$

Second, once we have guessed update $+U$, we need to ensure that it is sufficiently 'small', i.e., of polynomial size

$$\chi_s^{k+1} := \chi_s^0 \wedge \bigwedge_{C \subseteq Ag} \bigwedge_{\sigma_C} \langle\langle C \rangle\rangle \mathsf{X} \bigvee_{t \in Out(s,\sigma_C)} \chi_t^n$$

Forth and Back

We are in the realm of speculation

Towards Synthesis

Input Given are a CGM M, state s, and a target CL (or ATL) formula φ, such that $M, s \not\models \varphi$

First, we run an algorithm to essentially model check the following quantified expression: $\exists +U : M, s \models [+U]\varphi$

Second, once we have guessed update $+U$, we need to ensure that it is sufficiently ‘small’, i.e., of polynomial size

$$\chi_s := \chi_s^{|S|}$$

We are in the realm of speculation

Towards Synthesis

Input Given are a CGM M, state s, and a target CL (or ATL) formula φ, such that $M, s \not\models \varphi$

First, we run an algorithm to essentially model check the following quantified expression: $\exists + U : M, s \models [+U]\varphi$

Second, once we have guessed update $+U$, we need to ensure that it is sufficiently 'small', i.e., of polynomial size

While χ_s is exponential in the size of its syntax tree (due to repetitions), the size of its DAG representation is polynomial

We are in the realm of speculation

Towards Synthesis

Input Given are a CGM M, state s, and a target CL (or ATL) formula φ, such that $M, s \not\models \varphi$

First, we run an algorithm to essentially model check the following quantified expression: $\exists +U : M, s \models [+U]\varphi$

Second, once we have guessed update $+U$, we need to ensure that it is sufficiently 'small', i.e., of polynomial size

Finally, if we can guess an update $+U$ s.t. $M, s \models [+U]\varphi$, then there is an equivalent update $+U'$ of polynomial size

We are in the realm of speculation

Towards Synthesis

Input Given are a CGM M, state s, and a target CL (or ATL) formula φ, such that $M, s \not\models \varphi$

Output Update $+U$ such that $M, s \models [+U]\varphi$

Synthesis for DDCL is NP-complete

We are in the realm of speculation

Dynamics of Individual and Group Abilities

DDCL

We can grant and revoke individual forcing abilities

Updates can be used with any of your favourite CGM-based logics

Model checking is P-complete (for the logics with P model checking)

Synthesis is possible

What's next?

Group abilities has been considered

Dynamics of Individual and Group Abilities

DDCL

We can grant and revoke individual forcing abilities

Updates can be used with any of your favourite CGM-based logics

Model checking is P-complete (for the logics with P model checking)

Synthesis is possible

What's next?

Group abilities has been considered

How to grant abilities to a group such that no single agent has the ability?

Dynamics of Individual and Group Abilities

DDCL

We can grant and revoke individual forcing abilities

Updates can be used with any of your favourite CGM-based logics

Model checking is P-complete (for the logics with P model checking)

Synthesis is possible

What's next?

Group abilities has been considered

How to grant abilities to a group such that no single agent has the ability?

One solution: grant each agent in the group a special action s.t. in order to make a forcing transition the group has to synchronise on this action

Dynamics of Individual and Group Abilities

DDCL

We can grant and revoke individual forcing abilities

Updates can be used with any of your favourite CGM-based logics

Model checking is P-complete (for the logics with P model checking)

Synthesis is possible

What's next?

Group abilities has been considered

Granting abilities that are weaker than next-time (grant an agent the ability to eventually reach φ)

An agent is granted an ability to eventually make φ true while also allowing other agents to achieve their goals along the way

Dynamics of Individual and Group Abilities

DDCL

We can grant and revoke individual forcing abilities

Updates can be used with any of your favourite CGM-based logics

Model checking is P-complete (for the logics with P model checking)

Synthesis is possible

What's next?

Group abilities has been considered

Granting abilities that are weaker than next-time (grant an agent the ability to eventually reach φ)

Updates on general CGMs

The Dynamic Turn in Strategy Logics

RG, Gladyshev, Mittelmann, Motamed. *The Dynamic Turn in Strategy Logics*, AAMAS 2026.
Overview of the research programme

RG and Ågotnes. *Dynamic Coalition Logic: Granting and Revoking Dictatorial Powers*, LORI 2021.
Granting abilities to single agents

RG and Ågotnes. *Action Models for Coalition Logic*, DaLí 2022.
Complex DEL-styles updates of CGMs

RG, Gladyshev, Mittelmann, Motamed. *Changing The Rules of The Game: Reasoning about Dynamic Phenomena in Multi-Agent Systems*, AAMAS 2025.
Modular updates of CGMs

Catta, RG, and Mittelmann. *On Angels and Demons: Strategic (De)Construction of Dynamic Models*, AAMAS 2026.
Agents adding and removing arrows (see Davide Catta's DBLP for more work on obstruction)

RG, Grosinger, and Mittelmann. *I Would If I Could: Reasoning about Dynamics of Actions in Multi-Agent Systems*, KR 2026.
Adding and removing action to and from agents in perfect and imperfect information settings