Changing the Rules of the Game

Reasoning about Dynamic Phenomena in Multi-Agent Systems

Rustam Galimullin [1] Maksim Gladyshev [2] **Munyque Mittelmann** [3] Nima Motamed [2]

[1]University of Bergen (Norway)

[2]Utrecht University (The Netherlands)

[3]University of Naples Federico II (Italy)

munyque.mittelmann@unina.it

Motivation (1/2)

Alternating-time Temporal Logic (ATL)[1]: strategic ability of agents with temporal goals

- Systems are represented as Concurrent Game Structures (CGSs)
- Most research focuses on static or parameterized models

[1] R. Alur et al. (2002). "Alternating-time temporal logic". In: *Journal of the ACM* 49.5, pp. 672–713.

Motivation (1/2)

Alternating-time Temporal Logic (ATL)[1]: strategic ability of agents with temporal goals

- Systems are represented as Concurrent Game Structures (CGSs)
- Most research focuses on static or parameterized models

Realistic models of Multi-Agent Systems should accommodate **change**

- Regulations or Policy Updates
- Incorrect models
- Change of requirements

[1] R. Alur et al. (2002). "Alternating-time temporal logic". In: *Journal of the ACM* 49.5, pp. 672–713.

Motivation (1/2)

Alternating-time Temporal Logic (ATL)[1]: strategic ability of agents with temporal goals

- Systems are represented as Concurrent Game Structures (CGSs)
- Most research focuses on static or parameterized models

Realistic models of Multi-Agent Systems should accommodate **change**

- Regulations or Policy Updates
- Incorrect models
- Change of requirements

Autonomous agents are expected to strategize in dynamic settings

[1] R. Alur et al. (2002). "Alternating-time temporal logic". In: *Journal of the ACM* 49.5, pp. 672–713.

Motivation (2/2)

Scope

Reasoning about the effects of modifications on CGSs

Extend ATL:

- Nominals and hybrid logic operators (Areces et al. 2007)
- Update operators *inspired* from Dynamic Epistemic Logic (Ditmarsch et al. 2008)

Motivation (2/2)

Scope

Reasoning about the effects of modifications on CGSs

Extend ATL:

- Nominals and hybrid logic operators (Areces et al. 2007)
- Update operators *inspired* from Dynamic Epistemic Logic (Ditmarsch et al. 2008)

Contribution

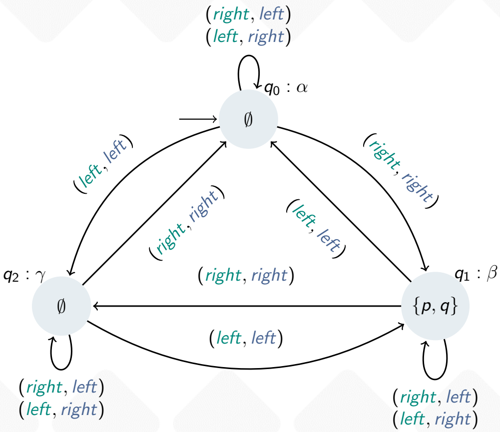
- New formalism for capturing dynamic phenomena
- Expressivity results
- Model Checking complexity

Concurrent Game Model (CGS)

A named CGS is a state transition model:

$$M = (\mathrm{AP}, \mathrm{Ag}, \mathrm{Ac}, \mathrm{S}, \mathrm{Nom}, \tau, \ell)$$

- AP — *atomic propositions*
- Ag — *agents*
- Ac — agents' *actions*
- S — *states*
- Nom — nominals
- $\tau : \mathrm{S} \times \mathrm{Ac}^{\mathrm{Ag}} \to \mathrm{S}$ — *transition function*
- $\ell : \mathrm{AP} \cup \mathrm{Nom} \to 2^{\mathrm{S}}$ — *labelling function*



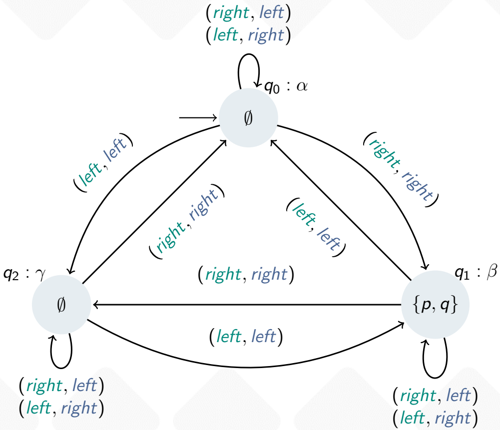
Concurrent Game Model (CGS)

A named CGS is a state transition model:

$$M = (\mathsf{AP}, \mathsf{Ag}, \mathsf{Ac}, \mathsf{S}, \mathsf{Nom}, \tau, \ell)$$

- AP *atomic propositions*
- Ag *agents*
- Ac agents' *actions*
- S *states*
- Nom nominals
- $\tau : \mathsf{S} \times \mathsf{Ac}^{\mathsf{Ag}} \to \mathsf{S}$ *transition function*
- $\ell : \mathsf{AP} \cup \mathsf{Nom} \to 2^{\mathsf{S}}$ *labelling function*

Each nominal refers to at most one state, and each state has (at least) one nominal



Strategy

Memoryless strategy

$\sigma : \mathsf{S} \rightarrow \mathsf{Ac}$

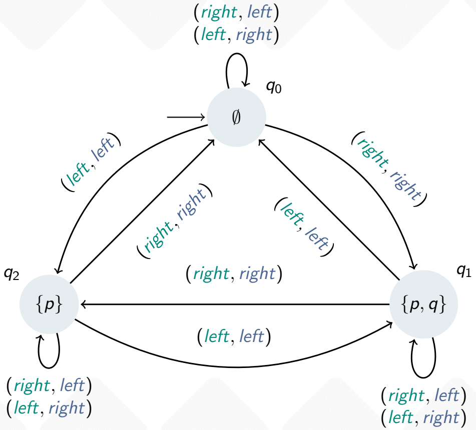
Strategy

Memoryless strategy

$\sigma : \mathsf{S} \to \mathsf{Ac}$

Example:

- $\sigma(q_0) = \textit{left}$
- $\sigma(q_1) = \textit{right}$
- $\sigma(q_2) = \textit{right}$



Hybrid ATL (HATL)

HATL Syntax

$$\varphi ::= p \mid \alpha \mid @_\alpha \varphi \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\langle A \rangle\rangle \mathbf{X}\varphi \mid \langle\langle A \rangle\rangle \varphi \mathbf{U} \varphi \mid \langle\langle A \rangle\rangle \varphi \mathbf{R} \varphi$$

where p is an atomic proposition, α is a nominal, and A is a coalition.

Semantics of the ATL operators

$\langle\langle A \rangle\rangle \psi$: there is a strategy for A enforcing ψ, independently of what the other agents do

Where ψ contains a temporal goal: next ($\mathbf{X}$), until ($\mathbf{U}$), or release ($\mathbf{R}$)

$\mathbf{F}\varphi := \top \mathbf{U} \varphi$ $\quad$ $\mathbf{G}\varphi := \bot \mathbf{R} \varphi$

Hybrid ATL (HATL)

HATL Syntax

$$\varphi ::= p \mid \alpha \mid @_\alpha \varphi \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\langle A\rangle\rangle \mathbf{X}\varphi \mid \langle\langle A\rangle\rangle \varphi \mathbf{U} \varphi \mid \langle\langle A\rangle\rangle \varphi \mathbf{R} \varphi$$

where p is an atomic proposition, α is a nominal, and A is a coalition.

Semantics of HATL operators

α: the current state is α

$@_\alpha \varphi$: at state named α, φ is true

Hybrid ATL (HATL)

HATL Syntax

$$\varphi ::= p \mid \alpha \mid @_\alpha \varphi \mid \neg \varphi \mid \varphi \vee \varphi \mid \langle\!\langle A \rangle\!\rangle \mathbf{X} \varphi \mid \langle\!\langle A \rangle\!\rangle \varphi \mathbf{U} \varphi \mid \langle\!\langle A \rangle\!\rangle \varphi \mathbf{R} \varphi$$

where p is an atomic proposition, α is a nominal, and A is a coalition.

Semantics of HATL operators

α: the current state is α

$@_\alpha \varphi$: at state named α, φ is true

$\langle\!\langle \emptyset \rangle\!\rangle \mathbf{F} \beta$: no matter what the agents do, a state named β will eventually be visited

Logic for ATL Model Building (LAMB)

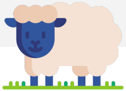
LAMB Syntax

$$\varphi ::= p \mid \alpha \mid @_\alpha\varphi \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\!\langle A \rangle\!\rangle \mathbf{X}\varphi \mid \langle\!\langle A \rangle\!\rangle \varphi\mathbf{U}\varphi \mid \langle\!\langle A \rangle\!\rangle \varphi\mathbf{R}\varphi \mid [\pi]\varphi$$

where p is an atomic proposition, α is a nominal, Act is an action profile, and A is a coalition.

Semantics of LAMB

$[\pi]\varphi$ means “after the update π, φ holds”



Logic for ATL Model Building (LAMB)

LAMB Syntax

$$\varphi ::= p \mid \alpha \mid @_\alpha \varphi \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\!\langle A \rangle\!\rangle \mathbf{X}\varphi \mid \langle\!\langle A \rangle\!\rangle \varphi \mathbf{U} \varphi \mid \langle\!\langle A \rangle\!\rangle \varphi \mathbf{R} \varphi \mid [\pi]\varphi$$

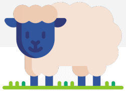
$$\pi ::= (p_\alpha := \psi) \mid \alpha \xrightarrow{Act} \alpha \mid @$$

where p is an atomic proposition, α is a nominal, Act is an action profile, and A is a coalition.

Semantics of LAMB

$[\pi]\varphi$ means "after the update π, φ holds", where π is either

- $p_\alpha := \psi$ the proposition p in α gets the current truth value of ψ



Logic for ATL Model Building (LAMB)

LAMB Syntax

$$\varphi ::= p \mid \alpha \mid @_\alpha \varphi \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\!\langle A \rangle\!\rangle \mathbf{X}\varphi \mid \langle\!\langle A \rangle\!\rangle \varphi \mathbf{U} \varphi \mid \langle\!\langle A \rangle\!\rangle \varphi \mathbf{R} \varphi \mid [\pi]\varphi$$

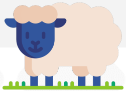
$$\pi ::= (p_\alpha := \psi) \mid \alpha \xrightarrow{Act} \alpha \mid @$$

where p is an atomic proposition, α is a nominal, *Act* is an action profile, and A is a coalition.

Semantics of LAMB

$[\pi]\varphi$ means "after the update π, φ holds", where π is either

- $p_\alpha := \psi$ the proposition p in α gets the current truth value of ψ
- $\alpha \xrightarrow{Act} \beta$ the *Act*-labeled arrow that starts in α is redirected to β



Logic for ATL Model Building (LAMB)

LAMB Syntax

$$\varphi ::= p \mid \alpha \mid @_\alpha \varphi \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\langle A\rangle\rangle \mathbf{X}\varphi \mid \langle\langle A\rangle\rangle \varphi \mathbf{U} \varphi \mid \langle\langle A\rangle\rangle \varphi \mathbf{R} \varphi \mid [\pi]\varphi$$

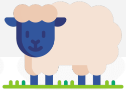
$$\pi ::= (p_\alpha := \psi) \mid \alpha \xrightarrow{Act} \alpha \mid ⓐ$$

where p is an atomic proposition, α is a nominal, Act is an action profile, and A is a coalition.

Semantics of LAMB

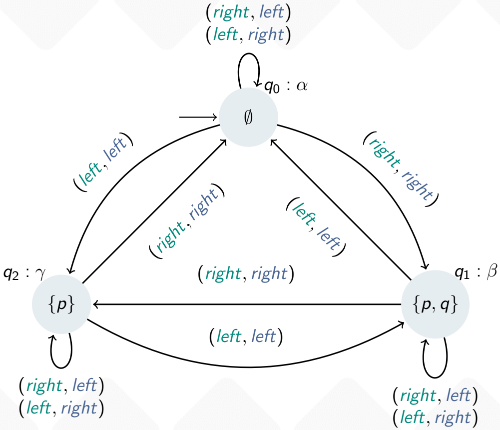
$[\pi]\varphi$ means "after the update π, φ holds", where π is either

- $p_\alpha := \psi$ the proposition p in α gets the current truth value of ψ
- $\alpha \xrightarrow{Act} \beta$ the *Act*-labeled arrow that starts in α is redirected to β
- ⓐ adds a new state and name it α



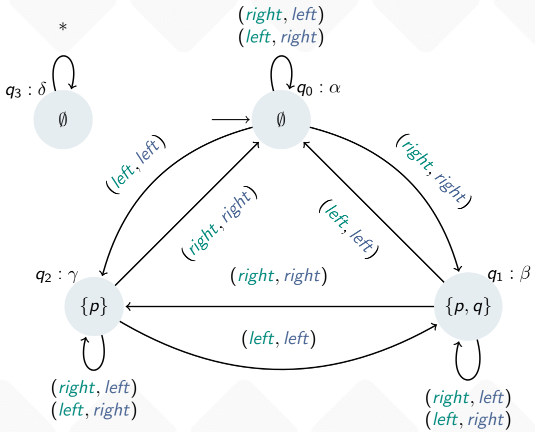
Update Operators

$\rightarrow$ δ



Update Operators

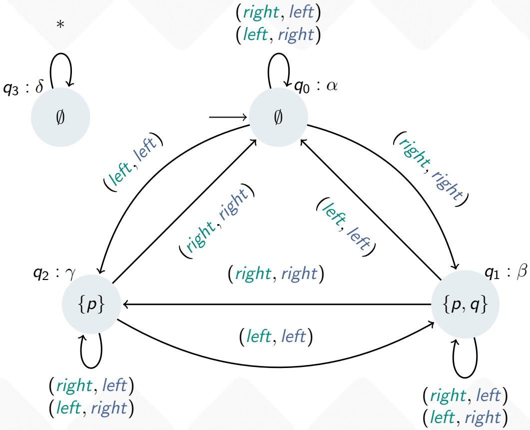
- $\textcircled{\delta}$



Update Operators

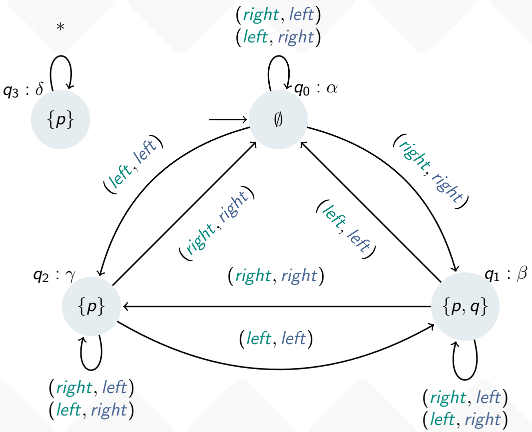
- $\textcircled{\delta}$

$\rightarrow$ $p_\delta := \neg q$ (assume $@_\alpha$)



Update Operators

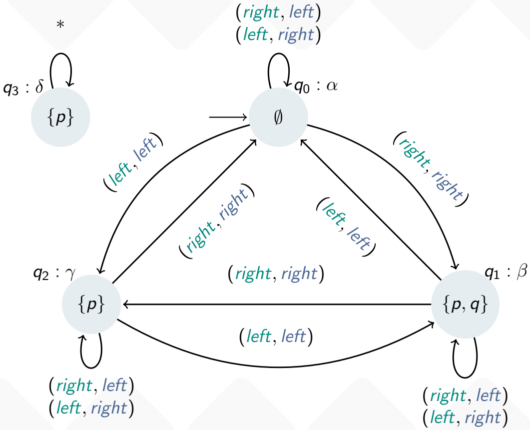
- Ⓓδ
- $p_\delta := \neg q \quad$ (assume $@_\alpha$)



Update Operators

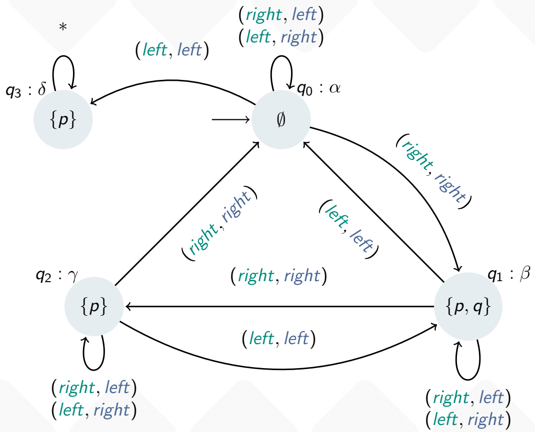
- ⓓ δ
- $p_\delta := \neg q$ (assume $@_\alpha$)

$\rightarrow \alpha \xrightarrow{(left, left)} \delta$



Update Operators

- $\textcircled{\delta}$
- $p_\delta := \neg q \quad$ (assume $@_\alpha$)
- $\alpha \xrightarrow{(left, left)} \delta$

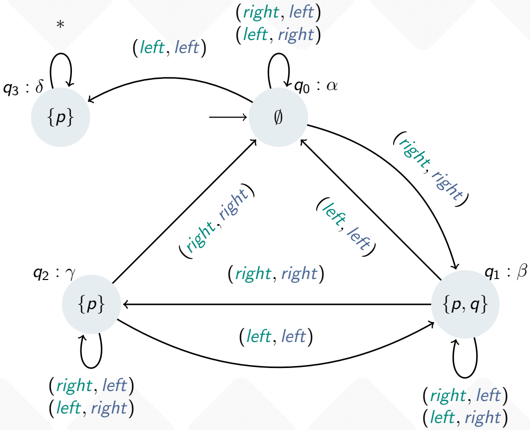


Update Operators

- $\textcircled{\delta}$
- $p_\delta := \neg q \quad$ (assume $@_\alpha$)
- $\alpha \xrightarrow{(left, left)} \delta$

Let [upd] denote the sequence of updates above and M be the original model

$M_{q_0} \models [\text{upd}]\langle\langle green, blue \rangle\rangle \mathbf{X}\delta$



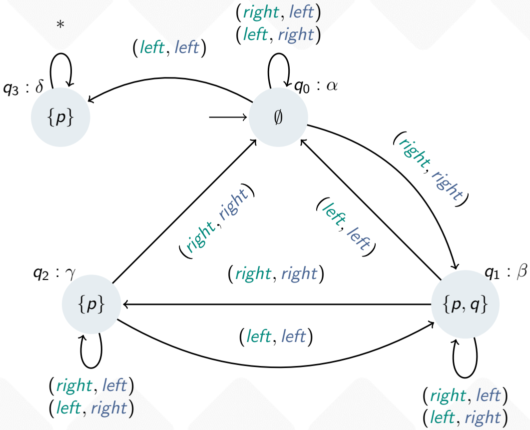
Update Operators

- δ
- $p_\delta := \neg q \quad$ (assume $@_\alpha$)
- $\alpha \xrightarrow{(left, left)} \delta$

Let [upd] denote the sequence of updates above and M be the original model

$M_{q_0} \models [\text{upd}]\langle\langle green, blue\rangle\rangle \mathbf{X}\delta$

$M_{q_0} \not\models [\text{upd}]@_\delta\langle\langle green, blue\rangle\rangle \mathbf{F} q$



Expressivity of LAMB

Expressivity

ATL $\prec$ HATL $\prec$ LAMB

Expressivity of LAMB

SLAMB:
The logic with only updates in the form $p_\alpha := \varphi$ has the same expressivity as HATL

ALAMB:
The logic with only updates in the form $\alpha \xrightarrow{Act} \beta$ is more expressive than HATL

$$\begin{tikzcd} \mathrm{HATL} \arrow[r, leftrightarrow] & \mathrm{SLAMB} \arrow[r] & \mathrm{ALAMB} \arrow[r, bend right] & \mathrm{LAMB} \arrow[l, bend right, dashed, "?"'] \end{tikzcd}$$

Model Checking

Model Checking

Given a model M, a state q, and a formula φ, the *model checking problem* is to decide whether

$$M, q \models \varphi$$

(i.e., whether φ holds in M at q)

Model Checking

Model Checking

Given a model M, a state q, and a formula φ, the *model checking problem* is to decide whether

$$M, q \models \varphi$$

(i.e., whether φ holds in M at q)

Model checking LAMB

PTime-complete

(lower bound from ATL)

Model checking algorithm for LAMB

procedure $\mathrm{MC}(M, s, \varphi)$
    case $\varphi = \alpha$
        return $s \in \ell(\alpha)$
    case $\varphi = @_\alpha \psi$
        if $\ell(\alpha) \neq \emptyset$ **then**
            return $\mathrm{MC}(M, \ell(\alpha), \psi)$
        else
            return *false*
    case $\varphi = [\pi]\psi$ with $\pi \in \{p_\alpha := \psi, \alpha \xrightarrow{Act} \beta, @\}$
        return $\mathrm{MC}(\textsc{Update}(M, s, \pi), s, \psi)$
        * Other cases are standard *
end procedure

Computing Updated Models

procedure $\textsc{Update}(M, s, \pi)$
    case $\pi = p_\alpha := \psi$
        if $\ell(\alpha) \neq \emptyset$ **then**
            if $\mathrm{MC}(M, s, \psi)$ **then**
                $\ell^\pi(p) = \ell(p) \cup \ell(\alpha)$
            else
                $\ell^\pi(p) = \ell(p) \setminus \ell(\alpha)$
            return $M^\pi = \langle S, \tau, \ell^\pi \rangle$
        else
            return M
    case $\pi = \alpha \xrightarrow{Act} \beta$
        if $\ell(\alpha) \neq \emptyset$ and $\ell(\beta) \neq \emptyset$ **then**
            $\tau^\pi = \tau \setminus \{(\ell(\alpha), Act, \tau(\ell(\alpha), Act))\} \cup \{(\ell(\alpha), Act, \ell(\beta))\}$
            return $M^\pi = \langle S, \tau^\pi, \ell \rangle$
        else
            return M
    (...)
end procedure

Bounded Synthesis

Assume a model that does not satisfy a specification φ.

Can we find a bounded modification to repair it?

Bounded Synthesis

Bounded modification synthesis problem

Let M_s be a CGS, φ be a formula, and n be a natural number

The *bounded synthesis problem* finds an update $\pi_\varphi := [\pi_1, \ldots, \pi_n]$, with the size at most n, such that $M_s \models [\pi_\varphi]\varphi$

Bounded Synthesis

Bounded modification synthesis problem

Let M_s be a CGS, φ be a formula, and n be a natural number

The *bounded synthesis problem* finds an update $\pi_\varphi := [\pi_1, \ldots, \pi_n]$, with the size at most n, such that $M_s \models [\pi_\varphi]\varphi$

Bounded synthesis problem for LAMB

NP-complete

(lower bound with reduction from 3-SAT)

Conclusion

- Ideas from both strategy logics and Dynamic Epistemic Logic
- Reasoning about dynamic phenomena
- Maintain ATL complexity with more expressivity
- Synthesis of modifications $\rightarrow$ system repair

Conclusion

- Ideas from both strategy logics and Dynamic Epistemic Logic
- Reasoning about dynamic phenomena
- Maintain ATL complexity with more expressivity
- Synthesis of modifications $\rightarrow$ system repair
- Future work
  - Satisfiability problem for LAMB
  - Costs associated with model changes
  - Unbounded synthesis
  - ATL* and SL

References

Alur, R., T. A. Henzinger, and O. Kupferman (2002). "Alternating-time temporal logic". In: *Journal of the ACM* 49.5, pp. 672–713.

Areces, Carlos and Balder ten Cate (2007). "Hybrid logics". In: *Handbook of Modal Logic*. Ed. by Patrick Blackburn, Johan Van Benthem, and Frank Wolter. Vol. 3. Studies in Logic and Practical Reasoning. Elsevier, pp. 821–868.

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

Acknowledgements

MM acknowledges support from the project

- *Strategic rEasoning for sociALly good mechanisms* (SEAL), which has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 101105549.

Funded by
the European Union

This presentation uses icons made by *Freepik* from Flaticon (www.flaticon.com).