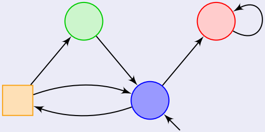
Quantitative Temporal Logics for Multi-Agent Systems

Nicolas Markey
CNRS – IRISA (Univ. Rennes, France)

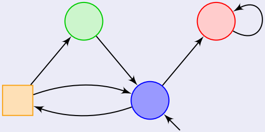
Reasoning about multi-agent systems

Games on graphs



Reasoning about multi-agent systems

Games on graphs

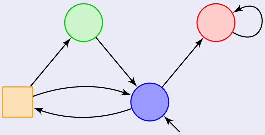


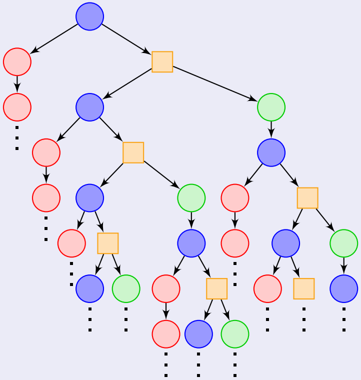
Strategy for player ▢

alternately go to ◯ and ◯ (starting with ◯).

Reasoning about multi-agent systems

Games on graphs



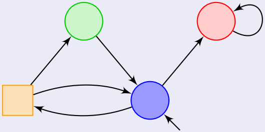


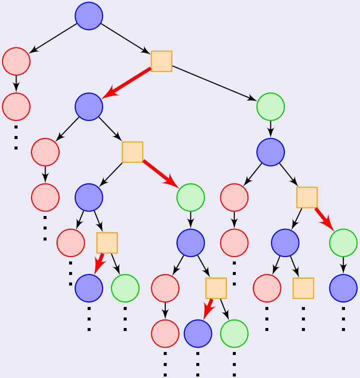
Strategy for player

alternately go to and
(starting with).

Reasoning about multi-agent systems

Games on graphs



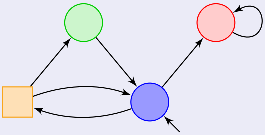


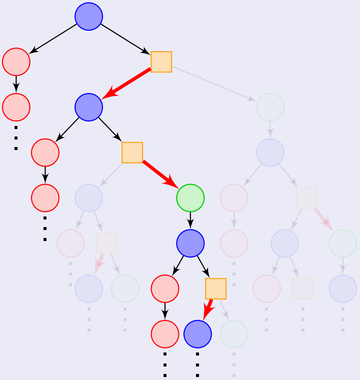
Strategy for player

alternately go to and
(starting with).

Reasoning about multi-agent systems

Games on graphs





Strategy for player

alternately go to and
(starting with).

Reasoning about strategic behaviours

Strategy logic [CHP07,MMV10]

LTL + explicit quantification and assignment of strategies:

$$\exists\sigma_A.\,\forall\sigma_B.\,\mathsf{assign}(A \to \sigma_A, B \to \sigma_B).\varphi_A$$

[CHP07] Chatterjee, Henzinger, Piterman. Strategy Logic. CONCUR, 2007.
[MMV10] Mogavero, Murano, Vardi. Reasoning about strategies. FSTTCS, 2010.

Reasoning about strategic behaviours

Strategy logic [CHP07,MMV10]

LTL + explicit quantification and assignment of strategies:

$$\underbrace{\exists\sigma_A.\,\forall\sigma_B}_{\text{Quantification over strategies}}.\,\underbrace{\text{assign}(A\to\sigma_A, B\to\sigma_B)}_{\text{Assignment of strategies to players}}.\,\underbrace{\varphi_A}_{\text{LTL property}}$$

[CHP07] Chatterjee, Henzinger, Piterman. Strategy Logic. CONCUR, 2007.

[MMV10] Mogavero, Murano, Vardi. Reasoning about strategies. FSTTCS, 2010.

Reasoning about strategic behaviours

Strategy logic [CHP07,MMV10]

LTL + explicit quantification and assignment of strategies:

$$\underline{\exists \sigma_A.\ \forall \sigma_B}.\ \underline{\mathsf{assign}(A \to \sigma_A, B \to \sigma_B)}.\ \underline{\varphi_A}$$

Quantification over strategies

Assignment of strategies to players

LTL property

Example (Characterisation of Nash equilibria)

$(\sigma_1, \ldots, \sigma_n)$ is a Nash equilibrium iff

$$\mathsf{assign}(\mathsf{Agt} \to (\sigma_1, \ldots \sigma_n)).\ \bigwedge_{A_i \in \mathsf{Agt}} ((\exists \sigma_i'.\mathsf{assign}(A_i \to \sigma_i').\ \varphi_i) \Rightarrow \varphi_i)$$

[CHP07] Chatterjee, Henzinger, Piterman. Strategy Logic. CONCUR, 2007.

[MMV10] Mogavero, Murano, Vardi. Reasoning about strategies. FSTTCS, 2010.

Reasoning about strategic behaviours

Strategy logic [CHP07,MMV10]

LTL + explicit quantification and assignment of strategies:

$$\underbrace{\exists \sigma_A.\ \forall \sigma_B}.\ \underbrace{\mathsf{assign}(A \to \sigma_A, B \to \sigma_B)}.\ \underbrace{\varphi_A}$$

Quantification over strategies

Assignment of strategies to players

LTL property

Theorem ([CHP07,DLM10])

SL model-checking is decidable (and TOWER-complete).

[CHP07] Chatterjee, Henzinger, Piterman. Strategy Logic. CONCUR, 2007.

[MMV10] Mogavero, Murano, Vardi. Reasoning about strategies. FSTTCS, 2010.

[DLM10] Da Costa, Laroussinie, Markey. ATL with Strategy Contexts: [...]. FSTTCS, 2010.

A fuzzy extension of Strategy Logic [BKM+19]

SL[$\mathcal{F}$]: a fuzzy extension of Strategy Logic

- propositions take rational values in $[0;1]$;
- allow functions $[0;1]^k \to [0;1]$

$$[\![\varphi \vee \psi]\!](\pi) = \max([\![\varphi]\!](\pi), [\![\psi]\!](\pi))$$

- quantitative semantics for **U**:

$$[\![\varphi \mathbf{U} \psi]\!](\pi) = \sup_{j \geq 0} \min([\![\psi]\!](\pi_{\geq j}), \min_{0 \leq i < j} [\![\varphi]\!](\pi_{\geq i}))$$

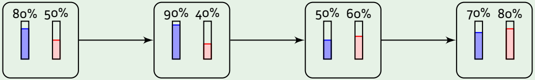
- $\exists \sigma_A.\Psi$: maximal value of Ψ over all strategies:

$$[\![\langle\!\langle x \rangle\!\rangle \varphi]\!](\pi) = \sup_{\sigma \in \mathrm{Strat}} ([\![\varphi[x \mapsto \sigma]]\!](\pi))$$

[BKM+19] Bouyer et al. Reasoning about Quality and Fuzziness of Strategic Behaviours. Submitted.

A fuzzy extension of Strategy Logic [BKM+19]

Example



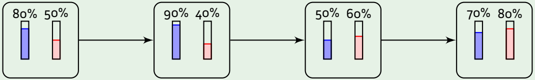
Along this run:

- $[\![\mathbf{F}\ \bigcirc]\!] =$
- $[\![\mathbf{G}\ \bigcirc]\!] =$
- $[\![\bigcirc\ \mathbf{U}\ \bigcirc]\!] =$

[BKM+19] Bouyer et al. Reasoning about Quality and Fuzziness of Strategic Behaviours. Submitted.

A fuzzy extension of Strategy Logic [BKM+19]

Example



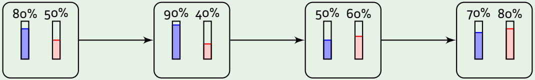
Along this run:

- ⟦**F** ●⟧ = 80%
- ⟦**G** ●⟧ =
- ⟦● **U** ●⟧ =

[BKM+19] Bouyer et al. Reasoning about Quality and Fuzziness of Strategic Behaviours. Submitted.

A fuzzy extension of Strategy Logic [BKM+19]

Example



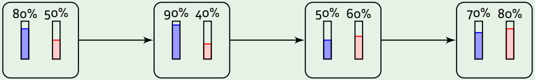
Along this run:

- $[\![\mathbf{F}\ \bigcirc]\!] = 80\%$
- $[\![\mathbf{G}\ \bigcirc]\!] = 50\%$
- $[\![\bigcirc\ \mathbf{U}\ \bigcirc]\!] =$

[BKM+19] Bouyer et al. Reasoning about Quality and Fuzziness of Strategic Behaviours. Submitted.

A fuzzy extension of Strategy Logic [BKM+19]

Example



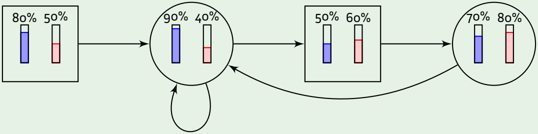
Along this run:

- $[\![\mathbf{F}\ \bigcirc]\!] = 80\%$
- $[\![\mathbf{G}\ \bigcirc]\!] = 50\%$
- $[\![\bigcirc\ \mathbf{U}\ \bigcirc]\!] = 60\%$

[BKM+19] Bouyer et al. Reasoning about Quality and Fuzziness of Strategic Behaviours. Submitted.

A fuzzy extension of Strategy Logic [BKM+19]

Example



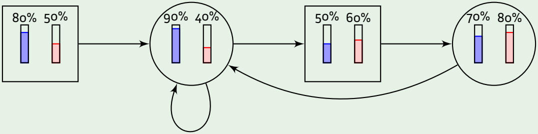
What is the best value that Player ⚪ can secure for:

- 🔵 **U** 🔴
- **F**(🔵 ∧ 🔴)

[BKM+19] Bouyer et al. Reasoning about Quality and Fuzziness of Strategic Behaviours. Submitted.

A fuzzy extension of Strategy Logic [BKM+19]

Example



What is the best value that Player $\bigcirc$ can secure for:

- $\bigcirc\ \mathbf{U}\ \bigcirc$

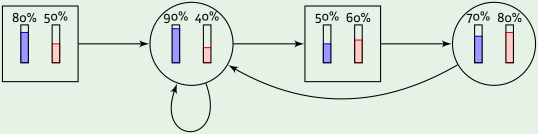
$$[\![\exists x.\mathrm{assign}(\bigcirc \mapsto x)\ \mathbf{A} \bigcirc\ \mathbf{U}\ \bigcirc]\!] = 60\%$$

- $\mathbf{F}(\bigcirc \wedge \bigcirc)$

[BKM+19] Bouyer et al. Reasoning about Quality and Fuzziness of Strategic Behaviours. Submitted.

A fuzzy extension of Strategy Logic [BKM+19]

Example



What is the best value that Player ⬤ can secure for:

- ⬤ **U** ⬤

$$[\![\exists x.\mathrm{assign}(\bullet \mapsto x)\ \mathbf{A} \bullet\ \mathbf{U}\ \bullet]\!] = 60\%$$

- **F**(⬤ ∧ ⬤)

$$[\![\exists x.\mathrm{assign}(\bullet \mapsto x)\ \mathbf{A}\ \mathbf{F}(\bullet \wedge \bullet)]\!] = 70\%$$

[BKM+19] Bouyer et al. Reasoning about Quality and Fuzziness of Strategic Behaviours. Submitted.

A fuzzy extension of Strategy Logic [BKM+19]

Example (Characterisation of Nash equilibria)

$(\sigma_1, ..., \sigma_n)$ is a Nash equilibrium iff formula $[\![\varphi_{NE}]\!] = 1$:

$$\varphi_{NE} = \text{assign}(\text{Agt} \rightarrow (\sigma_1, ...\sigma_n)). \bigwedge_{A_i \in \text{Agt}} [(\exists \sigma_i'.\text{assign}(A_i \rightarrow \sigma_i'). \varphi_i) \preceq \varphi_i.)$$

where

$$[\![\varphi \preceq \psi]\!](\pi) = \begin{cases} 1 & \text{if } [\![\varphi]\!](\pi) \leq [\![\psi]\!](\pi) \\ 0 & \text{otherwise} \end{cases}$$

[BKM+19] Bouyer et al. Reasoning about Quality and Fuzziness of Strategic Behaviours. Submitted.

A fuzzy extension of Strategy Logic [BKM+19]

Example (Characterisation of ε-Nash equilibria)

Formula $\Phi_{\bar{NE}}$ measures how far $(\sigma_1, \ldots, \sigma_n)$ is from being a NE:

$$\Phi_{\bar{NE}} = \mathsf{assign}(\mathsf{Agt} \to (\sigma_1, \ldots \sigma_n)). \bigvee_{A_i \in \mathsf{Agt}} \left[(\exists d.\mathsf{assign}(A_i \to d).\varphi_i) - \varphi_i \right]$$

Proposition

$(\sigma_1, \ldots, \sigma_n)$ is an ε-Nash equilibrium iff $\Phi_{\bar{NE}} \leq \varepsilon$.

Proposition

$(\sigma_1, \ldots, \sigma_n)$ is a Nash equilibrium iff $\Phi_{\bar{NE}} = 0$.

[BKM+19] Bouyer et al. Reasoning about Quality and Fuzziness of Strategic Behaviours. Submitted.

SL[$\mathcal{F}$] model checking is decidable

Theorem

Model checking SL[$\mathcal{F}$] is decidable (and TOWER-complete).

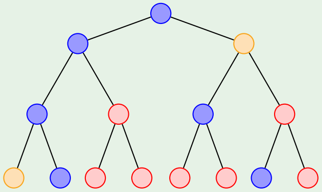
SL[$\mathcal{F}$] model checking is decidable

Theorem

Model checking SL[$\mathcal{F}$] is decidable (and TOWER-complete).

Proof (main ingredients)

- (alternating) tree automata:



SL[$\mathcal{F}$] model checking is decidable

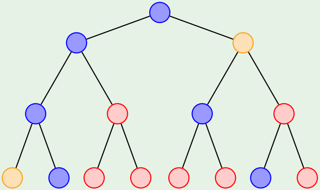
Theorem

Model checking SL[$\mathcal{F}$] is decidable (and TOWER-complete).

Proof (main ingredients)

- (alternating) tree automata:

$$\delta(q_0, {\color{blue}\bullet}) = (q_0, q_1) \vee (q_1, q_0)$$
$$\delta(q_0, {\color{red}\bullet}) = (q_1, q_1)$$
$$\delta(q_0, {\color{orange}\bullet}) = (q_2, q_2)$$
$$\delta(q_1, \circledast) = (q_1, q_1)$$
$$\delta(q_2, \circledast) = (q_2, q_2)$$



SL[$\mathcal{F}$] model checking is decidable

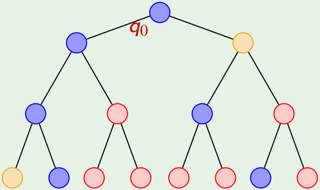
Theorem

Model checking SL[$\mathcal{F}$] is decidable (and TOWER-complete).

Proof (main ingredients)

- (alternating) tree automata:

$$\begin{aligned}
\delta(q_0, \textcolor{blue}{\bigcirc}) &= (q_0, q_1) \vee (q_1, q_0) \\
\delta(q_0, \textcolor{red}{\bigcirc}) &= (q_1, q_1) \\
\delta(q_0, \textcolor{orange}{\bigcirc}) &= (q_2, q_2) \\
\delta(q_1, \circledast) &= (q_1, q_1) \\
\delta(q_2, \circledast) &= (q_2, q_2)
\end{aligned}$$



SL[$\mathcal{F}$] model checking is decidable

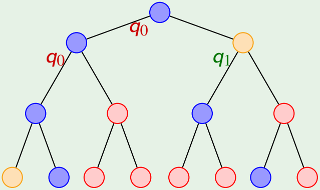
Theorem

Model checking SL[$\mathcal{F}$] is decidable (and TOWER-complete).

Proof (main ingredients)

- (alternating) tree automata:

$$\delta(q_0, \textcolor{blue}{\bigcirc}) = (q_0, q_1) \vee (q_1, q_0)$$
$$\delta(q_0, \textcolor{pink}{\bigcirc}) = (q_1, q_1)$$
$$\delta(q_0, \textcolor{orange}{\bigcirc}) = (q_2, q_2)$$
$$\delta(q_1, \circledast) = (q_1, q_1)$$
$$\delta(q_2, \circledast) = (q_2, q_2)$$



SL[$\mathcal{F}$] model checking is decidable

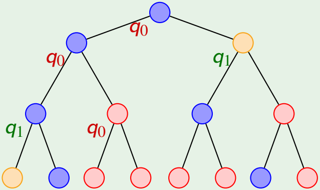
Theorem

Model checking SL[$\mathcal{F}$] is decidable (and TOWER-complete).

Proof (main ingredients)

- (alternating) tree automata:

$$
\begin{aligned}
\delta(q_0, {\color{blue}\bullet}) &= (q_0, q_1) \vee (q_1, q_0) \\
\delta(q_0, {\color{pink}\bullet}) &= (q_1, q_1) \\
\delta(q_0, {\color{orange}\bullet}) &= (q_2, q_2) \\
\delta(q_1, \circledstar) &= (q_1, q_1) \\
\delta(q_2, \circledstar) &= (q_2, q_2)
\end{aligned}
$$



SL[$\mathcal{F}$] model checking is decidable

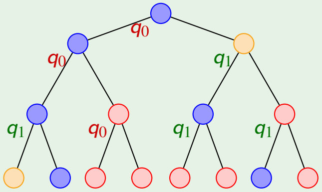
Theorem

Model checking SL[$\mathcal{F}$] is decidable (and TOWER-complete).

Proof (main ingredients)

- (alternating) tree automata:

$$\delta(q_0, \textcolor{blue}{\bullet}) = (q_0, q_1) \vee (q_1, q_0)$$
$$\delta(q_0, \textcolor{red}{\bullet}) = (q_1, q_1)$$
$$\delta(q_0, \textcolor{orange}{\bullet}) = (q_2, q_2)$$
$$\delta(q_1, \circledast) = (q_1, q_1)$$
$$\delta(q_2, \circledast) = (q_2, q_2)$$



SL[$\mathcal{F}$] model checking is decidable

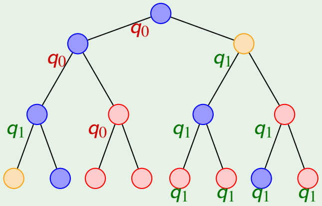
Theorem

Model checking SL[$\mathcal{F}$] is decidable (and TOWER-complete).

Proof (main ingredients)

- (alternating) tree automata:

$$\delta(q_0, \color{blue}{\bigcirc}) = (q_0, q_1) \vee (q_1, q_0)$$
$$\delta(q_0, \color{red}{\bigcirc}) = (q_1, q_1)$$
$$\delta(q_0, \color{orange}{\bigcirc}) = (q_2, q_2)$$
$$\delta(q_1, \circledast) = (q_1, q_1)$$
$$\delta(q_2, \circledast) = (q_2, q_2)$$



SL[$\mathcal{F}$] model checking is decidable

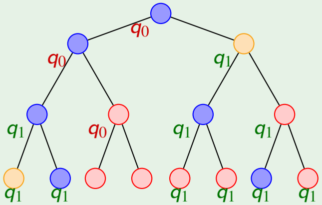
Theorem

Model checking SL[$\mathcal{F}$] is decidable (and TOWER-complete).

Proof (main ingredients)

- (alternating) tree automata:

$$\delta(q_0, {\color{blue}\bullet}) = (q_0, q_1) \vee (q_1, q_0)$$
$$\delta(q_0, {\color{red}\bullet}) = (q_1, q_1)$$
$$\delta(q_0, {\color{orange}\bullet}) = (q_2, q_2)$$
$$\delta(q_1, \circledast) = (q_1, q_1)$$
$$\delta(q_2, \circledast) = (q_2, q_2)$$



SL[$\mathcal{F}$] model checking is decidable

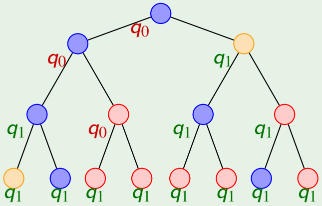
Theorem

Model checking SL[$\mathcal{F}$] is decidable (and TOWER-complete).

Proof (main ingredients)

- (alternating) tree automata:

$$\delta(q_0, {\color{blue}\bullet}) = (q_0, q_1) \vee (q_1, q_0)$$
$$\delta(q_0, {\color{pink}\bullet}) = (q_1, q_1)$$
$$\delta(q_0, {\color{orange}\bullet}) = (q_2, q_2)$$
$$\delta(q_1, \circledast) = (q_1, q_1)$$
$$\delta(q_2, \circledast) = (q_2, q_2)$$



SL[$\mathcal{F}$] model checking is decidable

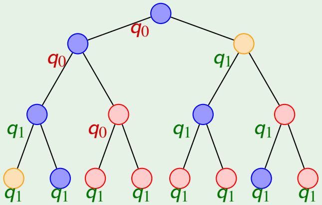
Theorem

Model checking SL[$\mathcal{F}$] is decidable (and TOWER-complete).

Proof (main ingredients)

- (alternating) tree automata:

$$\delta(q_0, \bigcirc) = (q_0, q_1) \vee (q_1, q_0)$$
$$\delta(q_0, \bigcirc) = (q_1, q_1)$$
$$\delta(q_0, \bigcirc) = (q_2, q_2)$$
$$\delta(q_1, \circledast) = (q_1, q_1)$$
$$\delta(q_2, \circledast) = (q_2, q_2)$$



This automaton corresponds to **E**$\bigcirc$ **U** $\bigcirc$

SL[$\mathcal{F}$] model checking is decidable

Theorem

Model checking SL[$\mathcal{F}$] is decidable (and TOWER-complete).

Proof (main ingredients)

- (alternating) tree automata
- projection: encodes quantification over atomic propositions.

SL[$\mathcal{F}$] model checking is decidable

Theorem

Model checking SL[$\mathcal{F}$] is decidable (and TOWER-complete).

Proof (main ingredients)

- (alternating) tree automata
- projection: encodes quantification over atomic propositions.
- bounded number of values:

Lemma

Given a game with N different rational values and a formula φ,

$$\#\{[\![\psi]\!](s) \mid \psi \text{ subformula of } \varphi\} \leq N^{|\varphi|}.$$

SL[$\mathcal{F}$] model checking is decidable

Theorem

Model checking SL[$\mathcal{F}$] is decidable (and TOWER-complete).

Proof (main ingredients)

- (alternating) tree automata
- projection: encodes quantification over atomic propositions.
- bounded number of values
- build automata $\mathcal{A}_{[\![\psi]\!]\in P}$ for all subformulas and all relevant intervals.