A quantitative semantics for Strategy Logic

Nicolas Markey
IRISA, CNRS & Inria & Univ. Rennes, France

joint work (published at IJCAI’19) with Patricia Bouyer, Orna Kupferman, Bastien Maubert, Aniello Murano, Giuseppe Perelli.
Reasoning about multi-agent systems

Games on graphs

[Diagram of a graph with nodes and arrows indicating the strategy for player to alternately go to and (starting with ).]
Reasoning about multi-agent systems

Games on graphs

Strategy for player

alternately go to \( \blacklozenge \) and \( \bullet \)
(starting with \( \blacklozenge \)).
Reasoning about multi-agent systems

Games on graphs

Strategy for player

alternately go to \(\square\) and \(\bigcirc\) (starting with \(\square\)).
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 •
(staring with ♠).
Reasonning about strategic behaviours

Strategy logic [CHP07, MMV10]

LTL + explicit quantification and assignment of strategies:

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

Reasoning about strategic behaviours

**Strategy logic [CHPo7,MMV10]**

LTL + explicit quantification and assignment of strategies:

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

- Quantification over strategies
- Assignment of strategies to players
- LTL property

Reasoning about strategic behaviours

Strategy logic [CHPo7, MMV10]

LTL + explicit quantification and assignment of strategies:

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

Example (Client-server interaction)

$\exists \sigma_S. \exists \sigma_C. \text{assign}(S \to \sigma_S)$. 

$(G \text{ mutual exclusion } \land 
\begin{array}{c}
G \\
\land \\
\bigwedge \\
C
\end{array}
\text{assign}(C \to \sigma_C). F \text{ access}_C$)


Reasoning about strategic behaviours

Strategy logic [CHP07, MMV10]

LTL + explicit quantification and assignment of strategies:

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

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

$$\text{assign}(\text{Agt} \rightarrow (\sigma_1, \ldots, \sigma_n)). \bigwedge_{A_i \in \text{Agt}} ((\exists \sigma'_i. \text{assign}(A_i \rightarrow \sigma'_i). \varphi_i) \Rightarrow \varphi_i.)$$

Reasoning about strategic behaviours

Strategy logic [CHP07, MMV10]

$LTL + explicit quantification and assignment of strategies:

$$\exists \sigma_A . \forall \sigma_B . assign(A \rightarrow \sigma_A, B \rightarrow \sigma_B) . \varphi_A$$

Quantification over strategies

Assignment of strategies to players

LTL property

Theorem ([DLM10])

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


QuantifiedCTL [ES84,Kup95,Fre01,DLM12]

QCTL extends CTL with propositional quantifiers

\[\exists p. \varphi\] means that there exists a labelling of the model with \(p\) under which \(\varphi\) holds.

Quantified CTL $[\text{ES84}, \text{Kup95}, \text{Fre01}, \text{DLM12}]$

QCTL extends CTL with propositional quantifiers

$\exists p. \varphi$ means that there exists a labelling of the model with $p$ under which $\varphi$ holds.

$$\Box E F \land \neg \left( \exists p. \left[ E F (\Box \land p) \land E F (\Box \land \neg p) \right] \right)$$

Quantified CTL \[\text{[ES84,Kup95,Fre01,DLM12]}\]

QCTL extends CTL with **propositional quantifiers**

\[
\exists p. \varphi \quad \text{means that there exists a labelling of the model with } p \text{ under which } \varphi \text{ holds.}
\]

\[
\mathbf{E} \mathbf{F} \Box \land \neg \left(\exists p. \left[\mathbf{E} \mathbf{F} (\Box \land p) \land \mathbf{E} \mathbf{F} (\Box \land \neg p)\right]\right) \equiv \text{uniq}(\Box)
\]

---


Quantified CTL [ES84,Kup95,Fre01,DLM12]

QCTL extends CTL with propositional quantifiers

\[ \exists p. \varphi \] means that there exists a labelling of the model with \( p \) under which \( \varphi \) holds.

\[
\mathsf{E} \mathsf{F} \bigcirc \land \neg \left( \exists p. \left[ \mathsf{E} \mathsf{F} (\bigcirc \land p) \land \mathsf{E} \mathsf{F} (\bigcirc \land \neg p) \right] \right) \equiv \mathsf{uniq}(\bigcirc)
\]

true if we label the Kripke structure;

false if we label the computation tree;


### Theorem

*Strategy logic can be translated into QCTL*. 

 Players has moves $m_1, ..., m_n$; from the transition table, we can compute the set $\text{Next}(q, A, m_i)$ of states that can be reached from $q$ when player $A$ plays $m_i$. 

**SL** can be translated as follows: 

- **encoding of $\exists\sigma.\psi$:**
  
  $\exists m_{\sigma_1} \exists m_{\sigma_2} ... \exists m_{\sigma_k}. A G (m_{\sigma_i} \iff \forall m_{\sigma_j} \hat{\psi}) 

- **encoding of assignment $(\alpha).\phi$:**
  
  $A[[G(q \land m_{\alpha}(A) \Rightarrow X \text{Next}(q, A, m_{\alpha}(A)) \Rightarrow \hat{\phi})]]$
Model checking Strategy Logic

Theorem

*Strategy logic can be translated into QCTL*. 

- Players has moves $m_1, ..., m_n$;
- From the transition table, we can compute the set $\text{Next}(\bullet, A, m_i)$ of states that can be reached from $\bullet$ when player $A$ plays $m_i$. 

\[A \left[ \text{G} (q \land m_{\alpha}(A) \Rightarrow X \text{Next}(q, A, m_{\alpha}(A))) \Rightarrow \hat{\phi} \right]\]
Model checking Strategy Logic

**Theorem**

*Strategy logic can be translated into QCTL*.

- players has moves $m_1, \ldots, m_n$;
- from the transition table, we can compute the set $\text{Next}(q, A, m_i)$ of states that can be reached from $q$ when player $A$ plays $m_i$.

**SL can be translated as follows:**

- encoding of $\exists \sigma. \psi$:

$$\exists m_1^\sigma \exists m_2^\sigma \ldots \exists m_k^\sigma. A G(m_1^\sigma \Leftrightarrow \bigwedge \neg m_j^\sigma) \land \hat{\psi}$$
Model checking Strategy Logic

**Theorem**

*Strategy logic can be translated into QCTL*.  

- players has moves $m_1, ..., m_n$;  
- from the transition table, we can compute the set $\text{Next}(\bullet, A, m_i)$ of states that can be reached from $\bullet$ when player $A$ plays $m_i$.

**SL can be translated as follows:**

- encoding of $\exists \sigma. \psi$:
  
  \[
  \exists m_1^\sigma \exists m_2^\sigma \ldots \exists m_k^\sigma. \ A G (m_i^\sigma \iff \bigwedge \neg m_j^\sigma) \land \hat{\psi}
  \]

- encoding of $\text{assign}(\alpha). \varphi$ (for full binding $\alpha : \text{Agt} \rightarrow \text{Strat}$):
  
  \[
  A \left[ G (q \land m_i^{\alpha(A)} \Rightarrow X \text{Next}(q, A, m_i^{\alpha(A)})) \Rightarrow \hat{\varphi} \right]
  \]
QCTL with tree semantics

**Theorem**

- Model checking QCTL with \( k \) quantifiers in the tree semantics is \( k \)EXPTIME-complete.
- Satisfiability of QCTL with \( k \) quantifiers in the tree semantics is \( (k+1) \)EXPTIME-complete.
QCTL with tree semantics

Theorem

- Model checking QCTL with $k$ quantifiers in the tree semantics is $k$-EXPTIME-complete.
- Satisfiability of QCTL with $k$ quantifiers in the tree semantics is $(k+1)$-EXPTIME-complete.

Proof

Using (alternating) parity tree automata:

\[
\begin{align*}
\delta(q_0, \odot) &= (q_0, q_1) \lor (q_1, q_0) \\
\delta(q_0, \cdot) &= (q_1, q_1) \\
\delta(q_0, \Box) &= (q_2, q_2) \\
\delta(q_1, \Box) &= (q_1, q_1) \\
\delta(q_2, \Box) &= (q_2, q_2)
\end{align*}
\]

This automaton corresponds to $\mathsf{E} \odot \mathsf{U} \cdot$
QCTL with tree semantics

Theorem

- Model checking QCTL with $k$ quantifiers in the tree semantics is $k$-EXPTIME-complete.
- Satisfiability of QCTL with $k$ quantifiers in the tree semantics is $(k+1)$-EXPTIME-complete.

Proof

- polynomial-size automata for CTL;
- quantification is handled by projection, which first requires removing alternation (exponential blowup);
- an automaton equivalent to a QCTL formula can be built inductively;
- emptiness of an alternating parity tree automaton can be decided in exponential time.
A fuzzy extension of Strategy Logic

Quantitative models

In our models, atomic propositions take values in \([0; 1]\):
- quality of satisfaction of a proposition
- (discretised, normalised) quantities, e.g. energy, distance, ...
A fuzzy extension of Strategy Logic

Quantitative models

In our models, atomic propositions take values in $[0; 1]$:
- **quality** of satisfaction of a proposition
- (discretised, normalised) quantities, e.g. **energy**, **distance**, ...

Example

How to reach $B$ while staying as far as possible from opponent?

![Diagram showing two robots, A and B, with a path marked as a dashed line to reach B while staying as far as possible from the opponent.](image)
A fuzzy extension of Strategy Logic

Quantitative models
In our models, atomic propositions take values in $[0; 1]$:
- quality of satisfaction of a proposition
- (discretised, normalised) quantities, e.g. energy, distance, ...

Example
How to reach $B$ while staying as far as possible from opponent?

$$
\begin{align*}
A & \rightarrow (0, 4) \rightarrow (5, 6) \rightarrow (0, 5) \rightarrow (5, 5) \\
& \quad \text{d = 0.35 B = 0} \\
& \quad \text{d = 0.31 B = 0} \\
& \quad \text{d = 0.27 B = 0}
\end{align*}
$$
A fuzzy extension of Strategy Logic

LTL[\mathcal{F}]: formally reasoning about quality [ABK16]

- quantitative semantics for LTL [FPS14]:

\[
[[\pi, \varphi \mathbf{U} \psi]] = \sup_i \left( \min(\min(\pi_i, \psi), \min(\min(\pi_j, \varphi))) \right).
\]

\[
\begin{align*}
\text{p} &= 0.5 \\
\text{q} &= 0.2 \\
\text{p} &= 0.3 \\
\text{q} &= 0.4 \\
\text{p} &= 0.7 \\
\text{q} &= 0.6 \\
\text{p} &= 0.8 \\
\text{q} &= 0.2
\end{align*}
\]

\[[p \mathbf{U} q] = 0.4\]


A fuzzy extension of Strategy Logic

LTL[\mathcal{F}]: formally reasoning about quality [ABK16]

- quantitative semantics for LTL [FPS14]:

\[
[\pi, \varphi U \psi] = \sup_i \left( \min(\pi_i, \psi), \min \left( \min_j (\pi_j, \varphi) \right) \right).
\]

\[\begin{align*}
\text{[}\pi, \varphi U \psi]\text{] = sup_{i} \left( \min(\pi_i, \psi), \min \left( \min_{j<i} (\pi_j, \varphi) \right) \right). \quad & \quad \text{[}p U q\text{]} = 0.4 \\
\text{[}\pi, \varphi U \psi]\text{] = sup_{i} \left( \min(\pi_i, \psi), \min \left( \min_{j<i} (\pi_j, \varphi) \right) \right). \quad & \quad \text{[}p U q\text{]} = 0.3
\end{align*}\]


A fuzzy extension of Strategy Logic

**LTL[\mathcal{F}]: formally reasoning about quality** \[\text{[ABK16]}\]

- **quantitative semantics for LTL** \[\text{[FPS14]}\]:

\[
\llbracket \pi, \varphi \mathbf{U} \psi \rrbracket = \sup_i \left( \min \left( \llbracket \pi_i, \psi \rrbracket, \min_{j<i} \left( \min \left( \llbracket \pi_j, \varphi \rrbracket \right) \right) \right).
\]

- **extension with functions:**
  - \text{max} (disjunction), \text{min} (conjunction);
  - any other function \( f : [0; 1]^m \rightarrow [0; 1] \).

---


A fuzzy extension of Strategy Logic

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

SL[$\mathcal{F}$] extends LTL[$\mathcal{F}$] with quantification over strategies:

$$\exists \sigma. \varphi_K \chi = \sup_{strategy \ s} \inf_{outcome \ \rho} J_{\rho, \varphi} K_{\chi}[\sigma \mapsto s]$$
A fuzzy extension of Strategy Logic

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

SL[\mathcal{F}] extends LTL[\mathcal{F}] with quantification over strategies:

\[
[s, \exists \sigma. \varphi]_\chi = \sup_{\text{strategy } s} \inf_{\text{outcome } \rho} \left[ \rho, \varphi \right]_\chi[\sigma \mapsto s].
\]
A fuzzy extension of Strategy Logic

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

SL[\mathcal{F}] extends LTL[\mathcal{F}] with quantification over strategies:

\[[s, \exists \sigma. \varphi]_\chi = \sup_{\text{strategy } s} \inf_{\text{outcome } \rho} \left[ [\rho, \varphi]_\chi[\sigma \mapsto s] \right].

**Example (Robot example)**

How to reach \(B\) while staying as far as possible from opponent?

\(\exists \sigma. \text{assign}(\text{robot} \rightarrow \sigma). \ d \ U \ B\)
A fuzzy extension of Strategy Logic

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

\( \text{SL}[\mathcal{F}] \) extends \( \text{LTL}[\mathcal{F}] \) with quantification over strategies:

\[
\left[ s, \exists \sigma.\varphi \right]_\chi = \sup_{\text{strategy } s} \inf_{\text{outcome } \rho} \left[ \rho, \varphi \right]_{\chi[\sigma \mapsto s]}.
\]

**Example (Characterisation of Nash equilibria)**

Formula \( \Phi_{NE} \) expresses the fact that \((\sigma_1, \ldots, \sigma_n)\) is a NE:

\[
\Phi_{NE} = \text{assign}(\text{Agt} \rightarrow (\sigma_1, \ldots \sigma_n)). \bigwedge_{A_i \in \text{Agt}} \left[ (\exists d.\text{assign}(A_i \rightarrow d).\varphi_i) \right] \preceq \varphi_i
\]

where the function \( \preceq: [0; 1]^2 \rightarrow \{0, 1\} \) is such that

\[
\preceq (\alpha, \beta) = 1 \text{ whenever } \alpha \leq \beta.
\]
A fuzzy extension of Strategy Logic

**SL[^F^]: a fuzzy extension of Strategy Logic**

SL[^F^] extends LTL[^F^] with quantification over strategies:

\[
[s, \exists \sigma. \varphi]_\chi = \sup_{\text{strategy } s} \inf_{\text{outcome } \rho} [\rho, \varphi]_\chi[\sigma \mapsto s].
\]

**Example (Characterisation of ε-Nash equilibria)**

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

\[
\Phi_{NE} = \text{assign}(\text{Agt} \rightarrow (\sigma_1, \ldots \sigma_n)). \bigvee_{A_i \in \text{Agt}} [(\exists d. \text{assign}(A_i \rightarrow d). \varphi_i) - \varphi_i]
\]

**Proposition**

\((\sigma_1, \ldots, \sigma_n)\) is an ε-Nash equilibrium iff \(\lbrack \Phi_{NE} \rbrack \leq \varepsilon\).
Model checking $\text{SL}[\mathcal{F}]$

**Theorem**

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

<table>
<thead>
<tr>
<th>Theorem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model checking $\text{SL}[\mathcal{F}]$ is decidable (and TOWER-complete).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sketch of proof</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lift the classical approach for SL to $\text{SL}[\mathcal{F}]$, using QCTL*.</td>
</tr>
</tbody>
</table>
## Theorem

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

## Sketch of proof

Lift the classical approach for $SL$ to $SL[F]$, using $QCTL^*$.

### Definition (Booleanly-Quantified CTL ($BQCTL^*[F]$))

$$
\varphi ::= p \mid \exists p. \varphi \mid E\psi \mid f(\varphi, \ldots, \varphi)
$$

$$
\psi ::= \varphi \mid X\psi \mid \psi U \psi \mid f(\psi, \ldots, \psi)
$$
Model checking SL[$\mathcal{F}$]

**Theorem**

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

**Sketch of proof**

Lift the classical approach for SL to SL[$\mathcal{F}$], using QCTL*.

**Definition (Booleanly-Quantified CTL (BQCTL*[$\mathcal{F}$]))**

\[ \varphi ::= p | \exists p. \varphi | E\psi | f(\varphi, \ldots, \varphi) \]
\[ \psi ::= \varphi | X\psi | \psi U \psi | f(\psi, \ldots, \psi) \]

**Remark**

BQCTL* [$\mathcal{F}$] is interpreted over quantitative trees, but quantification is boolean!
Model checking SL[$\mathcal{F}$]

**Theorem**

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

**Sketch of proof**

Key lemma:

**Lemma**

For any $\varphi \in BQCTL^*[\mathcal{F}]$ and any finite $V \subseteq [0; 1]$, the set

$$V_\varphi = \{[t, \varphi] \mid t \text{ quantitative tree with values in } V\}$$

is finite.
Model checking $\text{SL}[\mathcal{F}]$

**Theorem**

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

**Sketch of proof**

Key lemma:

**Lemma**

For any $\varphi \in BQCTL^* [\mathcal{F}]$ and any finite $V \subseteq [0; 1]$, the set

$$V_\varphi = \{ [[t, \varphi]] \mid t \text{ quantitative tree with values in } V \}$$

is finite.

$\sim$ For any $\varphi$ and $P \subseteq [0; 1]$, we can build tree automata characterizing $V$-trees $t$ for which $[[t, \varphi]] \in P$. 
Conclusions and future works

**Contributions**
- quantitative extension of Strategy Logic;
- (semi-)quantitative extension of QCTL*;
- model checking remains decidable.

**Future works**
- more applications, analysis of expressive power;
- specialized efficient algorithms for fragments of SL[\(\mathcal{F}\)];
- fully-quantitative extension of QCTL*.