PDL as a Multi-Agent Strategy Logic

Jan van Eijck

CWI & ILLC, Amsterdam

TARK, Chennai, Jan 8, 2013
Overview

Games and Strategies

Extending PDL to a Strategic Game Language

Epistemic Strategy Logic
## The PD Game

<table>
<thead>
<tr>
<th></th>
<th>cooperate</th>
<th>defect</th>
</tr>
</thead>
<tbody>
<tr>
<td>cooperate</td>
<td>$c, c$</td>
<td>$c, d$</td>
</tr>
<tr>
<td>defect</td>
<td>$d, c$</td>
<td>$d, d$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>cooperate</th>
<th>defect</th>
</tr>
</thead>
<tbody>
<tr>
<td>cooperate</td>
<td>2, 2</td>
<td>0, 3</td>
</tr>
<tr>
<td>defect</td>
<td>3, 0</td>
<td>1, 1</td>
</tr>
</tbody>
</table>

Output function: map from strategy pairs to outcomes.
Cooperation Strategy for Player 1 in PD
MASL Language

\[ t_i ::= a \mid ?? \mid !! \]
\[ c ::= (t_1 \ldots, t_n) \]
\[ \phi ::= \top \mid c \mid p \mid \neg \phi \mid \phi_1 \land \phi_2 \mid [\gamma] \phi \]
\[ \gamma ::= c \mid ?\phi \mid \gamma_1; \gamma_2 \mid \gamma_1 \cup \gamma_2 \mid \gamma^* \]
Term and Vector evaluation

\[
\begin{align*}
[a]_{S_i, s, i} &= \{a\} \\
[??]_{S_i, s, i} &= S_i \\
[!!]_{S_i, s, i} &= \{s[i]\}
\end{align*}
\]

\[
\left[ (t_1 \cdot t_n) \right]_{S, s} = \left[ t_1 \right]_{S_1, s, 1} \times \cdots \times \left[ t_n \right]_{S_n, s, n}
\]
Interpretation of \((c, !!)\) in PD

\[
\begin{align*}
&\quad c,!! \rightarrow cc \\
&\quad cc \rightarrow c,!! \\
&\quad c,!! \rightarrow dc \\
&\quad dc \rightarrow c,!! \\
&\quad c,!! \rightarrow cd \\
&\quad cd \rightarrow c,!! \\
&\quad c,!! \rightarrow dd \\
&\quad dd \rightarrow c,!! 
\end{align*}
\]
Truth for MASL

\( M = (N, S, o) \) with \( o : S \to P \).

\( M, s \models \top \) always

\( M, s \models c \) iff \( s \in \llbracket c \rrbracket^{S,s} \)

\( M, s \models p \) iff \( s \in o^{-1}(p) \)

\( M, s \models \lnot \phi \) iff \( M, s \not\models \phi \)

\( M, s \models \phi_1 \land \phi_2 \) iff \( \cdots \)

\( M, s \models [\gamma] \phi \) iff \( \cdots \)

\( \llbracket c \rrbracket^M = \{(s, t) \mid t \in \llbracket c \rrbracket^{S,s}\} \)

\( \llbracket ?\phi \rrbracket^M = \{(s, s) \mid M, s \models \phi\} \)

\( \llbracket \gamma_1; \gamma_2 \rrbracket^M = \llbracket \gamma_1 \rrbracket^M \circ \llbracket \gamma_2 \rrbracket^M \)

\( \llbracket \gamma_1 \cup \gamma_2 \rrbracket^M = \llbracket \gamma_1 \rrbracket^M \cup \llbracket \gamma_2 \rrbracket^M \)

\( \llbracket \gamma^* \rrbracket^M = (\llbracket \gamma \rrbracket^M)^* \).
Formula for ‘Game is Nash’

\[
\langle (??) \rangle \bigwedge_{i \in N} \bigvee_{v \in U} (u_i \geq v \land [(i, \bar{v})] \neg u_i > v).
\]
Meta-strategy: Tit-for-Tat

\((? (c, ??); (??, c) \cup ?? (d, ??); (??, d))^*\)
Axioms

\[ [c]c. \]
\[ \langle c \rangle \top. \]
\[ \langle c \rangle \phi \rightarrow [c] \phi \text{ (if } c \text{ determined).} \]
\[ [c] \phi \leftrightarrow \bigwedge_{a \in S_i} [c^i_a] \phi. \]
\[ (i_a, \parallel) \rightarrow (c \leftrightarrow c^i_a). \]

plus PDL axioms . . .
MASL is complete

Canonical model construction
MASL and coalition logic

\[
\begin{align*}
\text{Tr}(p) & := p \\
\text{Tr}(\neg \phi) & := \neg \text{Tr}(\phi) \\
\text{Tr}(\phi_1 \land \phi_2) & := \text{Tr}(\phi_1) \land \text{Tr}(\phi_2) \\
\text{Tr}([C] \phi) & := \bigvee_{c \in \dot{C}} [c] \text{Tr}(\phi).
\end{align*}
\]

\(\dot{C}\) defined by

\[
\left\{ (t_1, \ldots, t_n) \mid \begin{array}{ll}
  t_i & \in S_i & \text{if } i \in C, \\
  t_i & = ? & \text{otherwise}
\end{array} \right\}.
\]

\(M, s \models_{CL} \phi\) iff \(M, s \models_{MASL} \text{Tr}(\phi)\).
Epistemic MASL

\[ \phi ::= \cdots | [\gamma]\phi | [\alpha]\phi \]

\[ \gamma ::= \cdots \]

\[ \alpha ::= i | i \bar{\sim} | ?\phi | \alpha_1; \alpha_2 | \alpha_1 \cup \alpha_2 | \alpha^* \]

Define \( i \) as \((i \cup i \bar{\sim})^*\).
Then \( i \) is a reflexive, symmetric and transitive knowledge operator.
Intensional game forms

\[(N, W, R_1, \ldots, R_n)\]

where

- \(W\) is a set of pairs \((G, s)\) where \(G = (N, S)\) is a game form with \(s \in S\),
- each \(R_i\) is a binary relation on \(W\).
Restriction in Epistemic PD
Knowing Dictatorship

A knowing dictator is a player who *knows* that he is always able to get the best deal:

\[
[i][\neg?][v \in U \land \land j \in N - \{i\} :- u_j > v \land \langle (i, !!) \rangle u_i \geq v).
\]

Player 2 in restricted PD is a dictator, but not a knowing dictator. For all he knows, he could end up in state *dd*. 