Game Description Logic and Game Playing

Laurent Perrussel

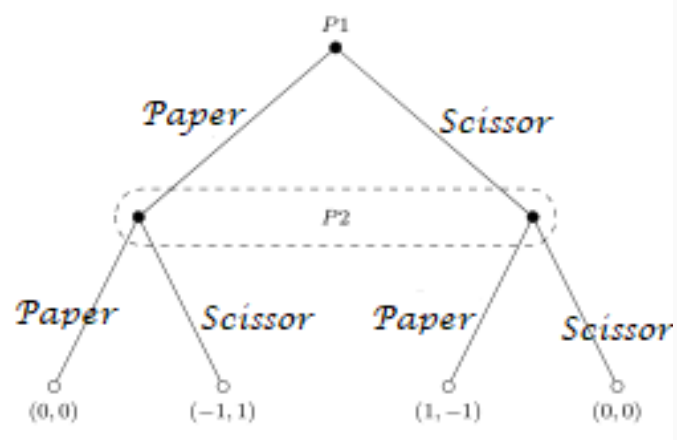
November 29 - Planning and Games workshop

IRIT – Université Toulouse Capitole

Motivation

Motivation(1/2)

- Game: describe and justify actions in a multi-agent context



- Autonomy for agent means
  - justify actions (agent rationality)

    P1 plays scissor because...

  - handling or playing in different environment (understand a game)

    P2 now plays Tic-tac-toe

Motivation (2/2)

Computer Science vs Game Theory?

- Game Theory

  Main goal: assessing the graph (i.e. the game) and find equilibrium or existence of winning strategies

- Computer Science

  Main goal: compact representation and computation of the possible next actions (Planning?)

van Benthem (2012)
Much of game theory is about the question whether strategic equilibria exist. But there are hardly any explicit languages for defining, comparing, or combining strategies.

Organization

Motivation

General Game Playing

Game Description Logic : GDL with a semantics

Reasoning for winning?

Imperfect Information

Equivalent games

Perspectives

General Game Playing

Representing games

Compact Representation

- Game rules and remarkable states.
  Initial and final states, legal actions...

- For a player: w.r.t. some context, what should be the next move?
  Rule "IF ... THEN..." (heuristics)

- General
  Language general enough for describing different games

General Game Playing

Overall goal

- Designing software agents which are able to play an unknown game

  Rules are discovered few minutes before the game starts

- Assess how an agent is general

  Assessment through a competition

- Specific game performance is not targeted

  Example: alphaGo

General Game Playing

Players

- No specific implementation

  Several implementation are available (Java, Prolog)

- No specific method to play

  Reasoning, Heuristics, Monte-Carlo, CSP

General Game Playing

Content

- language for describing games

  based on first order logic

- a platform for playing games (zero intelligence)

  Server managing a game play

- Interface for the players

  Communication about the current state and moves

- Starting point: `http://ggp.org`

Illustration

Tic-Tac-Toe

Tic Tac Toe (or Noughts and Crosses, Xs and Os) is a game for two players who take turns placing their marks in a 3x3 grid. The first player to place three of his marks in a horizontal, vertical, or diagonal row wins the game.

Ilustration

Tic-Tac-Toe en GDL (1/3)

```
;;; Components
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    (role white)
    (role black)
    ...

;;; init
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
    (init (cell 1 1 b))
    ...
    (init (cell 3 3 b))
    (init (control white))
```

Illustration

Tic-Tac-Toe en GDL (2/3)

```
;;; legal moves
    (<= (legal ?w (mark ?x ?y))
        (true (cell ?x ?y b))
        (true (control ?w)))

    (<= (legal white noop)
        (true (control black)))
    ...

;;; next (effects)
    (<= (next (cell ?m ?n x))
        (does white (mark ?m ?n))
        (true (cell ?m ?n b)))
    ...
```

Illustration

Tic-Tac-Toe en GDL (3/3)

```
;;; goal
    (<= (goal white 100)
        (line x)
        (not (line o)))

    (<= (goal white 0)
        (not (line x))
        (line o))
    ...

;;; terminal

    (<= terminal
        (line x))
    ...
```

Game Description Logic : GDL with a semantics

Signature and Language

Signature Agents, actions, propositions:

$$(N, \mathcal{A}, \Phi)$$

Language predefined symbols and temporal operators

$$\varphi ::= p \mid \mathit{initial} \mid \mathit{terminal} \mid \mathit{legal}(r, a) \mid \mathit{wins}(r) \mid \mathit{does}(r, a) \mid \neg\varphi \mid \varphi \wedge \psi \mid \bigcirc\varphi$$

Tic-Tac-Toe

GDL description of Tic-tac-Toe:

1. $\mathit{initial} \leftrightarrow \mathit{turn}(\mathrm{x}) \wedge \neg \mathit{turn}(\mathrm{o}) \wedge \bigwedge_{i,j=1}^{3} \neg(p^{\mathrm{x}}_{i,j} \vee p^{\mathrm{o}}_{i,j})$
2. $\mathit{wins}(r) \leftrightarrow \bigvee_{i=1}^{3} \bigwedge_{l=0}^{2} p^{r}_{i,1+l} \vee \bigvee_{j=1}^{3} \bigwedge_{l=0}^{2} p^{r}_{1+l,j} \bigwedge_{l=0}^{2} p^{r}_{1+l,1+l} \vee \bigwedge_{l=0}^{2} p^{r}_{1+l,3-l}$
3. $\mathit{terminal} \leftrightarrow \mathit{wins}(\mathrm{x}) \vee \mathit{wins}(\mathrm{o}) \vee \bigwedge_{i,j=1}^{3} (p^{\mathrm{x}}_{i,j} \vee p^{\mathrm{o}}_{i,j})$
4. $\mathit{legal}(r, a_{i,j}) \leftrightarrow \neg(p^{\mathrm{x}}_{i,j} \vee p^{\mathrm{o}}_{i,j}) \wedge \mathit{turn}(r) \wedge \neg \mathit{terminal}$
5. $\mathit{legal}(r, \mathit{noop}) \leftrightarrow \mathit{turn}(-r)$
6. $\bigcirc p^{r}_{i,j} \leftrightarrow p^{r}_{i,j} \vee (\mathit{does}(r, a_{i,j}) \wedge \neg(p^{\mathrm{x}}_{i,j} \vee p^{\mathrm{o}}_{i,j}))$
7. $\mathit{turn}(r) \rightarrow \bigcirc \neg \mathit{turn}(r) \wedge \bigcirc \mathit{turn}(-r)$

State-Transition Model (Perfect-Information Game)

$$M = (W, w_0, T, L, U, g, \pi)$$

- W is a non-empty finite set of *possible states*.
- $w_0 \in W$, representing the unique *initial* state.
- $T \subseteq W$, representing a set of *terminal* states.
- $L \subseteq W \backslash T \times N \times 2^{\mathcal{A}}$ is a *legality* relation, specifying legal actions for each agent at non-terminal states. Let $L_r(w) = \{a \in \mathcal{A} \ : \ (w, r, a) \in L\}$ be the set of all legal actions for agent r at state w. To make the game playable, we require $L_r(w) \neq \emptyset$ for every $r \in N$ and $w \in W \backslash T$.
- $U : W \times \mathcal{A}^{|N|} \to W \backslash \{w_0\}$ is an *update* function, specifying the state transition for each state and joint action.
- $g : N \to 2^W$ is a *goal* function, specifying the winning states of each agent.
- $\pi : W \to 2^{\Phi}$ is a standard valuation function.

Semantics

W.r.t. M, some path δ and index j

$$
\begin{array}{lll}
M, \delta, j \models p & \text{iff} & p \in \pi(\delta[j]) \\
M, \delta, j \models \neg\varphi & \text{iff} & M, \delta, j \not\models \varphi \\
M, \delta, j \models \varphi_1 \wedge \varphi_2 & \text{iff} & M, \delta, j \models \varphi_1 \text{ and } M, \delta, j \models \varphi_2 \\
M, \delta, j \models \mathit{initial} & \text{iff} & \delta[j] = w_0 \\
M, \delta, j \models \mathit{terminal} & \text{iff} & \delta[j] \in T \\
M, \delta, j \models \mathit{wins}(r) & \text{iff} & \delta[j] \in g(r) \\
M, \delta, j \models \mathit{legal}(r, a) & \text{iff} & a \in L_r(\delta[j]) \\
M, \delta, j \models \mathit{does}(r, a) & \text{iff} & \theta_r(\delta, j) = a \\
M, \delta, j \models \bigcirc\varphi & \text{iff} & \text{if } j < |\delta|, \text{ then } M, \delta, j+1 \models \varphi
\end{array}
$$

Reasoning for winning?

GDL-based Reasoning

Extend GDL and build a player on that extension

- Connecting action and strategy

  Priority over actions (eg. no loose first)

- Imperfect Information

  New kinds of game such as battleship

- Compare games

  Games with similar rules

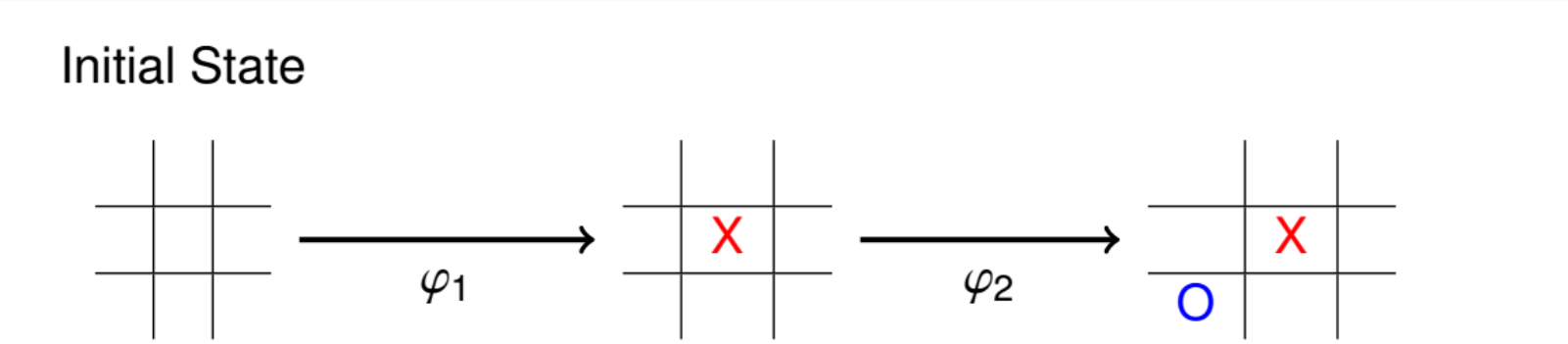
GDL and Strategic Reasoning (1/3)

"priority" operator: $\phi \nabla \psi$

ϕ *should hold; if not then* ψ *hold*

$$M, \delta, i \models \phi \text{ or } (\mathcal{P}(\phi, \delta[0, i]) = \emptyset \text{ and } M\delta, i \models \psi)$$

Example (Guifei PhD)



- $\varphi_1 := does(a^{x}_{2,2}) \ \nabla \ \bigvee_{i,j \in \{1,3\}} does(a^{x}_{i,j})$
- $\varphi_2 := does(a^{o}_{2,2}) \ \nabla \ \bigvee_{i,j \in \{1,3\}} does(a^{o}_{i,j})$

GDL and Strategic Reasoning (2/3)

Strategy for a game

- Fill the center:
  $fill\_center^r = does(a^r_{2,2})$
- Check if I can win:
  $check^r = \bigvee_{i,j=1}^{3} (does(a^r_{i,j}) \wedge \bigcirc wins(r))$
- Prevent immediate loss:
  $block^r = \bigvee_{i,j=1}^{3} (\bigcirc(does(a^{-r}_{i,j}) \wedge \bigcirc wins(\text{-}r)) \wedge does(a^r_{i,j}))$
- Fill an available corner:
  $fill\_corner^r = \bigvee_{i,j\in\{1,3\}} does(a^r_{i,j})$
- Fill anywhere available:
  $fill\_any^r = \bigvee_{i,j=1}^{3} does(a^r_{i,j})$
- Combined actions:
  $combined^r = fill\_centre^r \triangledown check^r \triangledown block^r \triangledown fill\_corner^r \triangledown fill\_any^r$

GDL and Strategic Reasoning (3/3)

Pending questions:

- How to design strategies?

  Connection with Machine Learning

- Generalize strategies?

  Are they any common points (General Strategic Reasoning)

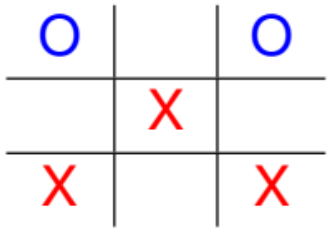
- How to implement?

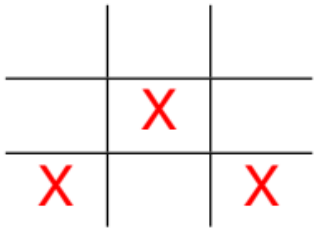
  Complexity of strategic reasoning and complexity of the game

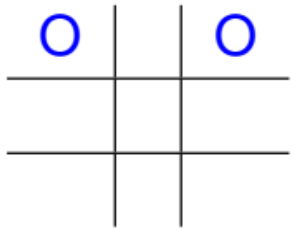
Imperfect Information

Imperfect Reasoning (1/4)

Krieg-Tictactoe







Each player can

- see her own marks, but not those of her opponent.
- know turn-taking and her own available actions.

How to describe and reason about games with imperfect information?

Imperfect Reasoning (2/4)

Epistemic extension of GDL with standard epistemic operators.

- $\mathsf{K}_r\varphi$ means "*player r knows* φ".
- $\mathsf{C}\varphi$ means "φ *is common knowledge among all the players*".

Syntax

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathit{initial} \mid \mathit{terminal} \mid \mathit{wins}(r) \mid \mathit{legal}(a^r) \mid$$

$$\mathit{does}(a^r) \mid \bigcirc\varphi \mid \mathsf{K}_r\varphi \mid \mathsf{C}\varphi$$

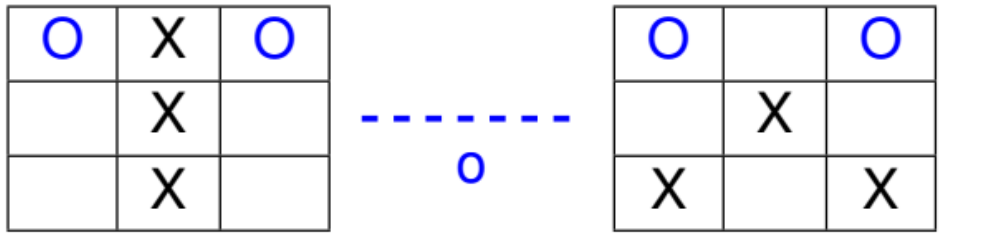
where $p \in \Phi$, $r \in N$ and $a^r \in A^r$.

Abbreviation: $\mathsf{E}\varphi =_{def} \bigwedge_{r\in N} \mathsf{K}_r\varphi$.

Imperfect Reasoning (3/4)

(1) $initial \rightarrow C initial$
(2) $legal(a^r) \rightarrow K_r(legal(a^r))$
(3) $does(a^r) \rightarrow K_r(does(a^r))$
(4) $wins(r) \rightarrow K_r(wins(r))$
(5) $terminal \rightarrow C terminal$

Krieg-Tictactoe satisfies all the properties, except (5).



terminal state non-terminal state

Imperfect Reasoning (4/4)

Pending questions:

- Benefit of the epistemic dimension?

  Compulsory for strategic reasoning (priority rules without epistemic operator)

- Different types of imperfect information?

  Perfect recall vs Memory less - Nested operator?

- How to implement

  Complexity of the game? Implementation with an epistemic reasoner?

Equivalent games

Equivalent games (1/3)

Number Scrabble:

1. $\mathit{initial} \leftrightarrow \mathit{turn}(\mathrm{b}) \wedge \neg \mathit{turn}(\mathrm{w}) \wedge \bigwedge_{i=1}^{9} \neg(s(\mathrm{b}, i) \vee s(\mathrm{w}, i))$
2. $\mathit{wins}(r) \leftrightarrow (\bigvee_{i=2}^{3}(s(r, i) \wedge s(r, 4) \wedge s(r, 11 - i)) \vee \bigvee_{i=1}^{2}(s(r, i) \wedge s(r, 6) \wedge s(r, 9 - i)) \vee \bigvee_{l=1}^{4}(s(r, 5 - l) \wedge s(r, 5) \wedge s(r, 5 + l)))$
3. $\mathit{terminal} \leftrightarrow \mathit{wins}(\mathrm{b}) \vee \mathit{wins}(\mathrm{w}) \vee \bigwedge_{i=1}^{9}(s(\mathrm{b}, i) \vee s(\mathrm{w}, i))$
4. $\mathit{legal}(r, \alpha(n)) \leftrightarrow \neg(s(\mathrm{b}, n) \vee s(\mathrm{w}, n)) \wedge \mathit{turn}(r) \wedge \neg \mathit{terminal}$
5. $\mathit{legal}(r, \mathit{noop}) \leftrightarrow \mathit{turn}(-r) \vee \mathit{terminal}$
6. $\bigcirc s(r, n) \leftrightarrow s(r, n) \vee (\neg(s(\mathrm{b}, n) \vee s(\mathrm{w}, n)) \wedge \mathit{does}(r, \alpha(n)))$
7. $\mathit{turn}(r) \wedge \neg \mathit{terminal} \rightarrow \bigcirc \neg \mathit{turn}(r) \wedge \bigcirc \mathit{turn}(-r)$

Equivalent games (2/3)

Equivalence

Semantics 2 models (State-Transition) with a bisimulation between them

Syntax Set of rules are equivalent

Number Scrabble and Tic-Tac-Toe are equivalent

Jeux équivalents (3/3)

Pending questions:

- Loose equivalence

  A game is "close" to a second one? Restricted equivalence to a sub-part of the game?

- Connecting equivalence and strategic reasoning

  "ready-to-go" strategies

- How to implement

  Complexity for deciding whether two games are equivalent. Available heuristics?

Perspectives

A lot of questions!

On GDL:

- Connecting action and strategy
- Imperfect Information
- Games comparison

Still on GDL

- Connection to planning
- Construction of a General Player ?

  Is it realistic to reason with GDL formulas?