COMP9414/9814/3411: Artificial Intelligence

7. Logic

[Russell & Norvig: 7.1, 7.2, 7.3, 7.4, 8.1, 8.2]
Outline

- Logical Agents
- Example: General Game-Playing Agents
- Propositional Logic
- First-Order Logic
Logical Agents

We humans know things; and what we know helps us do things

Knowledge-based agents
- use internal representations of knowledge
- reason about this knowledge to infer what to do

Logical agents
- use logical sentences to represent knowledge
- reason by inferring new sentences
Example: General Game-Playing Agents

**General Game Players** are systems

- able to understand formal descriptions of arbitrary games
- able to learn to play these games effectively.

Translation: They don't know the rules until the game starts

Unlike specialised game players (e.g. Deep Blue), they do not use algorithms designed in advance for specific games.
Variety of Games
Variants: Bughouse Chess
Variants: Kriegspiel
Other Games
How to Represent Game Rules

It is possible in principle to communicate game information in form of tables (for legal moves, state update)

Problem: Size of description. Even if everything is finite, the necessary tables can be large (e.g. $\sim 10^{44}$ states in Chess)

Solution: In many cases, worlds are best thought of in terms of atomic features that may change; e.g. “position-of-white-queen”, “black-can-castle”

We can represent features directly and use logic to describe how actions change individual features rather than entire states
Example: States in Noughts And Crosses

\[
\begin{array}{ccc}
1 & 2 & 3 \\
1 & X & \\
2 & O & \\
3 & X & \\
\end{array}
\]

- cell(1,1,x)
- cell(1,2,b)
- cell(1,3,b)
- cell(2,1,b)
- cell(2,2,o)
- cell(2,3,b)
- cell(3,1,b)
- cell(3,2,b)
- cell(3,3,x)
- control(oplayer)
Example Rules in the Game Description Language (GDL)

Roles and initial state

role(xplayer)
role(oplayer)
init(cell(1,1,b))
init(cell(1,2,b))
... 
init(cell(3,3,b))
init(control(xplayer))

Moves and their effect

legal(W,mark(M,N)) <=
true(cell(M,N,b)) ∧
true(control(W))

next(cell(M,N,x)) <=
does(xplayer,mark(M,N))

next(cell(M,N,o)) <=
does(oplayer,mark(M,N))

All highlighted symbols are pre-defined keywords in the logic-based GDL
Another Example: Wumpus World  [Russell & Norvig: 7.2]

- **Agent's actuators**
  - turn 90° left or right
  - move forward one square
  - grab the gold
  - shoot an arrow in the current direction
  - climb out of the cave from [1,1]

- **Agent's percepts**
  - a stench when adjacent to the wumpus
  - a breeze when adjacent to a pit
  - a glitter when in the square with gold
  - a bump when walking into a wall
  - a scream when the wumpus is killed
Agents in the Wumpus World

Main challenge: The initial ignorance of the configuration of the environment
Solution: Draw the right inferences from the available information
Propositional Logic
Propositional Logic: Vocabulary

Proposition symbols: \( p, q, r \)

Logical connectives:
- \( \neg p \) (negation, read: "not p")
- \( p \land q \) (conjunction, read: "p and q")
- \( p \lor q \) (disjunction, read: "p or q")
- \( p \Rightarrow q \) (implication, read: "p implies q")
- \( p \iff q \) (equivalence, read: "p if and only if q")

Sentences: built from proposition symbols and connectives
- e.g. \( \text{ready} \iff (a4paper \lor a3paper) \land \neg \text{jam} \)

Operator precedence: \( \neg, \land, \lor, \Rightarrow, \iff \)
Example: Knights and Knaves

Assumptions
- Knights always tell the truth
- Knaves always lie

Example
- Two people, A and B
- A says: One of us is a knave!
Propositional Logic: Semantics

Truth table

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>¬p</th>
<th>p ∧ q</th>
<th>p ∨ q</th>
<th>p =&gt; q</th>
<th>p &lt;=&gt; q</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>false</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
</tbody>
</table>

Example

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>¬A</th>
<th>¬B</th>
<th>¬A ∨ ¬B</th>
<th>A &lt;=&gt; ¬A ∨ ¬B</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td></td>
<td></td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Models

A model $\mathcal{M}$ (in propositional logic) fixes the truth-value (true or false) for every proposition symbol.

Example: $\mathcal{M} = \{A=\text{true}, B=\text{false}\}$

- $\mathcal{M}$ satisfies a proposition $p$ iff $p=\text{true} \in \mathcal{M}$
- $\mathcal{M}$ satisfies a sentence $\neg \alpha$ iff $\mathcal{M}$ does not satisfy $\alpha$
- $\mathcal{M}$ satisfies a sentence $\alpha \land \beta$ iff $\mathcal{M}$ satisfies $\alpha$ and $\mathcal{M}$ satisfies $\beta$
- $\mathcal{M}$ satisfies a sentence $\alpha \lor \beta$ iff $\mathcal{M}$ satisfies $\alpha$ or $\mathcal{M}$ satisfies $\beta$ (or both)
- $\mathcal{M}$ satisfies a sentence $\alpha \Rightarrow \beta$ iff $\mathcal{M}$ satisfies $\beta$ whenever $\mathcal{M}$ satisfies $\alpha$
- $\mathcal{M}$ satisfies a sentence $\alpha \Leftrightarrow \beta$ iff $\mathcal{M}$ satisfies $\alpha$ just in case $\mathcal{M}$ satisfies $\beta$

Example: $\mathcal{M} = \{A=\text{true}, B=\text{false}\}$ satisfies $A \Leftrightarrow \neg A \lor \neg B$
Logical Reasoning

β follows logically from α (or: β is a logical consequence of α), written $\alpha \models \beta$
iff every model that satisfies $\alpha$ also satisfies $\beta$

Example: $(A \land \neg B)$ follows logically from $(A \iff \neg A \lor \neg B)$

Let's consider some more examples.

- One person, A. A says: I am a knight!
- How many models satisfy this sentence?
- What follows logically from this sentence?

- One person, A. A says: I am a knave!
- How many models satisfy this sentence?
Logical Reasoning in the Wumpus World

\[ P_{x,y} \] there is a pit in \([x,y]\)
\[ W_{x,y} \] there is a wumpus in \([x,y]\)
\[ B_{x,y} \] the agent perceives a breeze in \([x,y]\)
\[ S_{x,y} \] the agent perceives a stench in \([x,y]\)

- There is no pit in \([1,1]\)
  \[ \neg P_{1,1} \]

- A breeze in \([1,1]\) means there is a pit in \([1,2]\) or \([2,1]\)
  \[ B_{1,1} \iff P_{1,2} \lor P_{2,1} \]

- A breeze in \([2,1]\) means there is a pit in \([1,1]\), \([2,2]\) or \([3,1]\)
  \[ B_{2,1} \iff P_{1,1} \lor P_{2,2} \lor P_{3,1} \]

- The breeze percepts for the first two squares visited are
  \[ \neg B_{1,1} \text{ and } B_{2,1} \]

Let \( KB \) be the conjunction of these sentences. Then \( KB \models \neg P_{2,1} \land (P_{2,2} \lor P_{3,1}) \)
First-Order Logic
First-Order Logic: Vocabulary

Quantifiers:  \( \forall, \exists \)
Constants:  richard, john
Variables:  \( X, Y, Z \)
Predicates:  king, married, loves
Functions:  father, mother
Logical Connectives:  \( \neg, \land, \lor, \Rightarrow, \Leftrightarrow \)

The **arity** of a function or predicate is the number of arguments that can be supplied

**Term**
- a variable, \( X \)
- a constant, richard
- or a functional term \( \text{mother}(\text{father}(\text{john})) \)

**Sentence**
- an atomic sentence, \( \text{married}(\text{father}(\text{richard}),\text{mother}(\text{john})) \)
- or a complex sentence \( \forall X \exists Y \text{loves}(X,Y) \Rightarrow \exists Y \forall X \text{loves}(X,Y) \)
Examples

Represent the following sentences in first-order logic, using a consistent vocabulary.

1) Emily is either a surgeon or a lawyer.
2) Joe is an actor, but he also holds another job.
3) All surgeons are doctors.
4) Some doctors are also lawyers.

"Every dog who loves one of its brothers is happy."

Are the following first-order formulae correct/incorrect/meaningless?

- $\forall X \ Dog(X) \land (\exists Y \ Brother(Y,X) \land Loves(X,Y)) \Rightarrow \ Happy(X)$
- $\forall X, Y \ Dog(X) \land Brother(Y,X) \land Loves(X,Y) \Rightarrow \ Happy(X)$
- $\forall X \ Dog(X) \land [\forall Y \ Brother(Y,X) \Leftrightarrow Loves(X,Y)] \Rightarrow \ Happy(X)$
- $\forall X \ Dog(X) \land (\exists Y \ Loves(X, Y \land Brother(X)) \Rightarrow \ Happy(X)$
Other Examples of First-Order Logic Sentences

\[ \forall W, M, N \quad \text{legal}(W, \text{mark}(M, N)) \leq \begin{align*} &\text{true}(\text{cell}(M, N, b)) \land \text{true}(\text{control}(W)) \\ &\forall M, N \quad \text{next}(\text{cell}(M, N, x)) \leq \text{does}(\text{xplayer}, \text{mark}(M, N)) \\ &\forall M, N \quad \text{next}(\text{cell}(M, N, o)) \leq \text{does}(\text{oplayer}, \text{mark}(M, N)) \end{align*} \]

Note the reverse implication in GDL, as in Prolog.
First-Order Logic: Semantics

A model $M$ (in first-order logic) fixes the truth-value (true or false) for every atomic sentence without variables.

Example:

$$M = \{ \text{king(richard)} = \text{true}, \text{king(john)} = \text{false}, \text{king(father(john))} = \text{true}, \text{king(mother(richard))} = \text{false}, \text{married(richard, john)} = \text{false}, \text{married(richard, mother(john))} = \text{false}, \ldots \}$$
Logical Entailment in First-Order Logic

- $\mathcal{M}$ satisfies a variable-free atomic sentence $p$ iff $p=true \in \mathcal{M}$
- $\mathcal{M}$ satisfies a sentence $\neg \alpha$ iff $\mathcal{M}$ does not satisfy $\alpha$
- $\mathcal{M}$ satisfies a sentence $\alpha \land \beta$ iff $\mathcal{M}$ satisfies $\alpha$ and $\mathcal{M}$ satisfies $\beta$
- $\mathcal{M}$ satisfies a sentence $\alpha \lor \beta$ iff $\mathcal{M}$ satisfies $\alpha$ or $\mathcal{M}$ satisfies $\beta$ (or both)
- $\mathcal{M}$ satisfies a sentence $\alpha \Rightarrow \beta$ iff $\mathcal{M}$ satisfies $\beta$ whenever $\mathcal{M}$ satisfies $\alpha$
- $\mathcal{M}$ satisfies a sentence $\alpha \iff \beta$ iff $\mathcal{M}$ satisfies $\alpha$ just in case $\mathcal{M}$ satisfies $\beta$
- $\mathcal{M}$ satisfies a sentence $\forall x \alpha$ iff $\mathcal{M}$ satisfies $\alpha\{x/t\}$ for all variable-free terms $t$
- $\mathcal{M}$ satisfies a sentence $\exists x \alpha$ iff $\mathcal{M}$ satisfies $\alpha\{x/t\}$ for some variable-free term $t$

$\alpha\{x/t\}$ means to replace each occurrence of $x$ by $t$ in $\alpha$.

$\beta$ follows logically from $\alpha$ (or: $\beta$ is a logical consequence of $\alpha$), written $\alpha \models \beta$ iff every model that satisfies $\alpha$ also satisfies $\beta$.
Examples

All humans are mortal. \( \forall X \text{human}(X) \Rightarrow \text{mortal}(X) \) (\( \alpha \))

Socrates is human. \( \text{human}(\text{socrates}) \) (\( \beta \))

Some cards are red. \( \exists X \text{card}(X) \land \text{red}(X) \) (\( \alpha \))

\( \heartsuit 7 \) and \( \spadesuit 8 \) are cards. \( \text{card}(\heartsuit 7) \land \text{card}(\spadesuit 8) \) (\( \beta \))
Examples

All humans are mortal. \( \forall X \text{human}(X) \Rightarrow \text{mortal}(X) \) \((\alpha)\)
Socrates is human. \( \text{human}(\text{socrates}) \) \((\beta)\)

- \( M_1 = \{\text{human}(\text{socrates}) = \text{false}, \text{mortal}(\text{socrates}) = \text{false}\} \)
  satisfies \( \alpha \) (why?) but not \( \beta \)
- \( M_2 = \{\text{human}(\text{socrates}) = \text{true}, \text{mortal}(\text{socrates}) = \text{true}\} \) satisfies \( \alpha \) and \( \beta \)
- \( M_2 \) is the only model that satisfies \( \alpha \) and \( \beta \). Hence \( \alpha \wedge \beta \models \text{mortal}(\text{socrates}) \)

Some cards are red. \( \exists X \text{card}(X) \wedge \text{red}(X) \) \((\alpha)\)
\( \heartsuit 7 \) and \( \clubsuit 8 \) are cards. \( \text{card}(\heartsuit 7) \wedge \text{card}(\clubsuit 8) \) \((\beta)\)

- \( \{\text{card}(\heartsuit 7) = \text{true}, \text{card}(\clubsuit 8) = \text{true}, \text{red}(\heartsuit 7) = \text{true}, \text{red}(\clubsuit 8) = \text{false}\} \) satisfies \( \alpha \wedge \beta \)
- \( \{\text{card}(\heartsuit 7) = \text{true}, \text{card}(\clubsuit 8) = \text{true}, \text{red}(\heartsuit 7) = \text{false}, \text{red}(\clubsuit 8) = \text{true}\} \) satisfies \( \alpha \wedge \beta \)
- Hence \( \alpha \wedge \beta \models \not\text{red}(\heartsuit 7) \)
Applied Logic:
Describing Games in GDL
Elements of a Game Description

A complete game description consists of:

- names of the players
- initial game state
- which moves are possible ("move generator")
- the effects of the moves ("game physics")
- termination conditions and result
Noughts and Crosses: Players and Initial State

\[
\text{role(xplayer)} \\
\text{role(oplayer)} \\
\text{init(cell(1,1,b))} \\
\text{init(cell(1,2,b))} \\
\text{init(cell(1,3,b))} \\
\text{init(cell(2,1,b))} \\
\text{init(cell(2,2,b))} \\
\text{init(cell(2,3,b))} \\
\text{init(cell(3,1,b))} \\
\text{init(cell(3,2,b))} \\
\text{init(cell(3,3,b))} \\
\text{init(control(xplayer))}
\]
Move Generator

\[ \text{legal}(W, \text{mark}(M,N)) \iff \begin{align*} \text{true}(\text{cell}(M,N,b)) \land \text{true}(\text{control}(W)) \end{align*} \]

\[ \text{legal}(xplayer, \text{noop}) \iff \text{true}(\text{control}(xplayer)) \]

\[ \text{legal}(o\text{p}layer, \text{noop}) \iff \text{true}(\text{control}(o\text{p}layer)) \]

Logical consequences:

\[ \alpha \land \beta \models \text{legal}(x\text{p}layer, \text{noop}) \]

\[ \alpha \land \beta \models \text{legal}(o\text{p}layer, \text{mark}(1,2)) \]

\[ \ldots \]

\[ \alpha \land \beta \models \text{legal}(o\text{p}layer, \text{mark}(3,2)) \]
Physics (Example)

cell(1,1,x)  cell(1,1,x)
cell(1,2,b)  cell(1,2,b)
cell(1,3,b)  cell(1,3,o)
cell(2,1,b)  cell(2,1,b)
cell(2,2,o)  cell(2,2,o)
cell(2,3,b)  cell(2,3,b)
cell(3,1,b)  cell(3,1,b)
cell(3,2,b)  cell(3,2,b)
cell(3,3,x)  cell(3,3,x)
control(oplayer) control(xplayer)

xplayer: noop  oplayer: mark(1,3)

X |   |   
---|---|---
  | O |   
---|---|---
  |   | X

X |   |   
---|---|---
  | O |   
---|---|---
  |   | X
Physics

\[
\begin{align*}
\text{next}(\text{cell}(M,N,x)) & \iff \text{does}(xplayer,\text{mark}(M,N)) \\
\text{next}(\text{cell}(M,N,o)) & \iff \text{does}(oplayer,\text{mark}(M,N)) \\
\text{next}(\text{cell}(M,N,W)) & \iff \text{does}(W,\text{mark}(J,K)) \land \\
& \quad \text{true}(\text{cell}(M,N,W)) \land \\
& \quad (\text{distinct}(M,J) \lor \text{distinct}(N,K)) \\
\text{next}(\text{control}(xplayer)) & \iff \text{true}(\text{control}(ooperator)) \\
\text{next}(\text{control}(ooperator)) & \iff \text{true}(\text{control}(xplayer))
\end{align*}
\]

Logical consequences:

\[\alpha \ (\text{slide 32}) \land \beta \land \text{does}(xplayer,\text{noop}) \land \text{does}(ooperator,\text{mark}(1,3)) \vDash \text{next}(\text{cell}(1,3,o)) \land \text{next}(\text{control}(xplayer)) \land \ldots\]
Termination and Result

**terminal** <= line(x)
**terminal** <= line(o)
**terminal** <= ¬open

line(X) <= row(M,X)
line(X) <= column(M,X)
line(X) <= diagonal(X)

open <= true(cell(M,N,b))

goal(xplayer,100) <= line(x)
goal(xplayer, 50) <= ¬line(x) ∧ ¬line(o) ∧ ¬open
goal(xplayer,  0) <= line(o)

goal(oplayer,100) <= line(o)
goal(oplayer, 50) <= ¬line(x) ∧ ¬line(o) ∧ ¬open
goal(oplayer,  0) <= line(x)

row(M,X) <= true(cell(M,1,X)) ∧ true(cell(M,2,X)) ∧ true(cell(M,3,X))
column(N,X) <= true(cell(1,N,X)) ∧ true(cell(2,N,X)) ∧ true(cell(3,N,X))
diagonal(X) <= true(cell(1,1,X)) ∧ true(cell(2,2,X)) ∧ true(cell(3,3,X))
diagonal(X) <= true(cell(1,3,X)) ∧ true(cell(2,2,X)) ∧ true(cell(3,1,X))
Summary: GDL Keywords

- **role(r)** means that $r$ is a role (i.e. a player) in the game
- **init(f)** means that $f$ is true in the initial position (state)
- **true(f)** means that $f$ is true in the current state
- **does(r,a)** means that role $r$ does action $a$ in the current state
- **next(f)** means that $f$ is true in the next state
- **legal(r,a)** means that it is legal for $r$ to play $a$ in the current state
- **goal(r,v)** means that $r$ gets goal value $v$ in the current state
- **terminal** means that the current state is a terminal state
- **distinct(s,t)** means that terms $s$ and $t$ are syntactically different

Note: Like in Prolog, all variables are implicitly universally quantified

\[
\text{next} \left( \text{cell}(M,N,x) \right) \leq \text{does}(xplayer, \text{mark}(M,N))
\]

means \[
\forall M,N \ \text{next} \left( \text{cell}(M,N,x) \right) \leq \text{does}(xplayer, \text{mark}(M,N))
\]
Knowledge Interchange Format (a.k.a. KIF) is a standard for programmatic exchange of knowledge represented in relational logic.

- Syntax is prefix version of standard syntax
- Some operators are renamed: `not`, `and`, `or`
- Case-independent. Variables are prefixed with `?`

\[
\text{r}(X, Y) \leq p(X, Y) \land \neg q(Y)
\]

\[
(\leq (r ?x ?y) \land (p ?x ?y) \neg q ?y)))
\]

or, equivalently,

\[
(\leq (r ?x ?y) (p ?x ?y) \neg q ?y)))
\]

The meaning is the same.
Noughts And Crosses in KIF

(role xplayer)
(role oplayer)

(init (cell 1 1 b))
(init (cell 1 2 b))
(init (cell 1 3 b))
(init (cell 2 1 b))
(init (cell 2 2 b))
(init (cell 2 3 b))
(init (cell 3 1 b))
(init (cell 3 2 b))
(init (cell 3 3 b))
(init (control xplayer))

(<= (next (cell ?m ?n x))
  (does xplayer (mark ?m ?n))
  (true (cell ?m ?n b))
  (true (control ?w)))
(<= (legal ?w (mark ?m ?n))
  (true (cell ?m ?n b))
  (true (control ?w)))
(<= (line ?x) (row ?m ?x))
(<= (line ?x) (column ?n ?x))
(<= (line ?x) (diagonal ?x))
(<= open
  (true (cell ?m ?n b)))
  <= terminal (line x))
  <= terminal (line o))
  <= terminal (not open))
<= goal xplayer 100)
  (line x))
  <= goal xplayer 50)
  (not (line x))
  (not (line o))
  (not open))
<= goal oplayer 0)
  (line o))
  <= goal oplayer 100)
  (line o))
  <= goal oplayer 50)
  (not (line x))
  (not (line o))
  (not open))
<= goal oplayer 0)
  (line x))
Game Manager

Player

Player

Player

TCP/IP
Communication Protocol

- Manager sends **START** message to players
  (START <ID> <ROLE> <GAME RULES> <STARTCLOCK> <PLAYCLOCK>)
  - ID: Match Identifier
  - Role: the name of the role you are playing (e.g. xplayer or oplayer)
  - Game rules: the axioms describing the game
  - Start/play clock: how much time you have before the game begins/per turn

- Manager sends **PLAY** message to players
  (PLAY <ID> <PRIOR MOVES>)
  Prior moves is a list of moves, one per player
  - The order is the same as the order of roles in the game description
  - e.g. ((mark 1 1) noop)
  - Special case: for the first turn, prior moves is nil

Players send back a message of the form **MOVE**, e.g. (mark 3 2)

- When the previous turn ended the game, Manager sends a **STOP** message
  (STOP <ID> <PRIOR MOVES>)
www.general-game-playing.de/downloads.html

Download Game Manager ("GameController")

Download Basic Players
GameController App

![GameController App Screenshot]

**MatchID** | TestMatch_1
---|---
**Startclock** | 10
**Playclock** | 5

<table>
<thead>
<tr>
<th>Role</th>
<th>Type</th>
<th>Host</th>
<th>Port</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>XPLAYER</td>
<td>RANDOM</td>
<td>-</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>OPLAYER</td>
<td>RANDOM</td>
<td>-</td>
<td>0</td>
<td>100</td>
</tr>
</tbody>
</table>

INFO(12:43:15.123): match:TestMatch_1, GDL v1
INFO(12:43:15.129): game:tictactoe
INFO(12:43:15.129): starting game with startclock=10, playclock=5
INFO(12:43:15.131): step:1
INFO(12:43:15.135): role: XPLAYER => player: local(Random)
INFO(12:43:15.136): role: OPLAYER => player: local(Random)
INFO(12:43:15.137): Sending start messages ...
INFO(12:43:15.153): time after gameStart's runThreads: Mon May 02 12:43:15 EST
More on General Game Playing

Websites – games.stanford.edu  ggp.org  general-game-playing.de
- Games
- Game Manager
- Reference Players & Development Tools
- Literature

World Cup, administered by Stanford
- 2005 – Cluneplayer (USA)
- 2006 – Fluxplayer (Germany)
- 2009, 2010 – Ary (France)
- 2011, 2013 – TurboTurtle (USA)
- 2014 – Sancho (USA)

MOOC (starts today(!))
- www.coursera.org/course/ggp