Describing Games Correctly

Ji Ruan, Prof. W. van der Hoek, Prof. M. Wooldridge
Department of Computer Science

Http://www.csc.liv.ac.uk/~jruan

THE UNIVERSITY *of* LIVERPOOL

## Computers play games

3.5 : 2.5
11 May 1997

The victory of **IBM's Deep Blue** computer over the world champion chess player **Gary Kasparov** in 1997 was regarded as one of the most significant events in the history of Artificial Intelligence.

However, Deep Blue is **specially** designed to play Chess, and it has no clue about how to play another game like Tic-Tac-Toe.
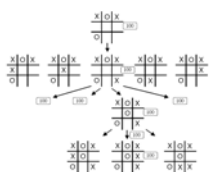
The **General Game Playing (GGP)** Competition was introduced by American Association for Artificial Intelligence in 2005 to test the ability of the computer players to play games in general, rather than a specific game.

How to represent games?
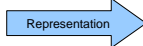
Representation

Games can be represented as trees.
For example, Tic-Tac-Toe :

Problem: too many states

Tic-Tac-Toe has around 5000 states. Chess has around 10^30 states.

In GGP, **Game Description Language (GDL)** is introduced to represent games *succinctly*. For example, Tic-Tac-Toe can be represented in GDL in 80 lines:

```
(role xplayer)           (<= (legal ?w
(role oplayer)           (mark ?x ?y))
(init (cell 1 1 b))         (true (cell ?x ?y b))
(init (cell 1 2 b))         (true (control ?w)))
(init (cell 1 3 b))      (<= (legal xplayer noop)
(init (cell 2 1 b))         (true (control
(init (cell 2 2 b))      oplayer)))
...
(init (control xplayer)) (<= (goal xplayer 100)
(<= (next (cell ?m ?n x))   (line x))
   (does xplayer        (<= (goal xplayer 50)
(mark ?m ?n))              (not (line x))
   (true (cell ?m ?n b)))   (not (line o))
...                         (not open))
(<= (next (control       (<= (goal xplayer 0)
xplayer))                   (line o))
   (true (control        ...
oplayer)))               (<= terminal
...                         (line x))
                         ...
```

Problems with GDL

GDL can describe a game, but it does not provide a tool to check whether a game is described correctly.

All well-defined games should have three basic properties:

- *Termination: The game does not go on forever.*
- *Winnability: Each player has a possibility to win the game.*
- *Playability: Each player has legal moves in every non-terminal state.*

In Our Research

We draw a connection between **GDL** and **Alternating-time Temporal Logic (ATL)**, a logic for reasoning about multi-agent systems.

- We use ATL formally specify game properties. For example, the ATL expression <<>> E Terminal, specifies *Termination*.

- We translate a GDL game description to an ATL module.

- We use a model checker called MOCHA to verify whether a game property holds in a GDL description.

GDL Description — hold? — Game Properties
ATL Module — MOCHA — ATL Specification
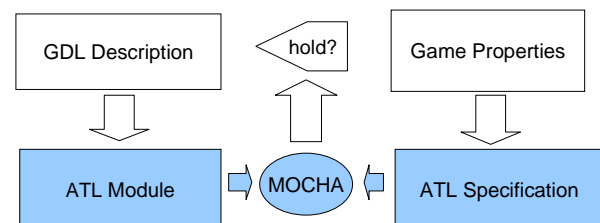
- Result: the complexity of reasoning about GDL-specified games using ATL specifications is EXPTIME-complete.

Outlook

Currently, GDL only allows for describing games with perfect information, e.g. chess. We try to extend GDL to enable the description of imperfect information games, e.g. poker.