

# General Game Playing

## Representing and Reasoning About the Rules of General Games With Imperfect Information<sup>1</sup>

Aldana Ramirez, Jeremías Rodríguez, Sebastián Meli

Introducción a la Inteligencia Artificial  
Universidad Nacional de Rosario

13 de marzo de 2016

---

<sup>1</sup>Escrito por Stephan Schieffel y Michael Thielscher

# Conceptos de IA involucrados

- Knowledge
- Learning
- Reasoning
- Planning
- Decision Making
- Logic Programming
- Situation Calculus

# General Game Playing

## General Game Playing (GGP)

Es un sistema que aprende a jugar juegos previamente desconocidos, sin intervención humana, solo a partir de sus reglas.

## ¿Cómo representamos las reglas?

Se ha desarrollado un lenguaje de alto nivel llamado Game Description Language (GDL) que permite describir cualquier juego con un número finito de jugadores y movimientos legales en cada estado.

Los movimientos son determinísticos y completamente observables, y todas las reglas del juego son conocidas por todos los jugadores (información completa).

Las reglas son fáciles de entender y mantener, y la semántica es precisa y totalmente procesable por una máquina.

Ejemplos: Ajedrez, Go.

# Game Description Language with Imperfect Information

Este paper propone una extensión al lenguaje GDL para incluir aleatoriedad y conocimiento incompleto del estado del juego, llamado Game Description Language with Imperfect Information (GDL-II).

Permite describir juegos con:

- Un número finito de jugadores que conocen todas las reglas de juego pero pueden tener información **incompleta o asimétrica**.
- Un número finito de movimientos legales que pueden ser **no determinísticos**.

Ejemplos: Poker, Bridge, Backgammon

La sintaxis de GDL-II está basada en la sintaxis estándar de los programas lógicos (incluyendo negación).

Una descripción de un juego es un conjunto finito de sentencias lógicas usando los predicados de la tabla 1, donde se imponen ciertas restricciones.

role(R)	R es un jugador.
init(F)	F vale en la posición inicial.
true(F)	F vale en la posición actual.
legal(R,M)	El jugador R puede hacer el movimiento M en la posición actual.
does(R,M)	El jugador R hace el movimiento M.
next(F)	F vale en la posición siguiente.
terminal	La posición actual es terminal.
goal(R,V)	El jugador R gana V puntos en la posición actual.
sees(R,P)	El jugador R percibe P en la siguiente posición.
random	El jugador aleatorio. <sup>2</sup>

Tabla 1: GDL-II keywords

---

<sup>2</sup>Notar que no necesariamente es un jugador convencional.

# Ejemplo: Krieg-Tic Tac Toe<sup>3</sup>

## Especificando las reglas en GDL-II

```
1  role(xplayer).
2  role(oplayer).
3
4  init(control(xplayer)).
5  init(cell(1,1,b)). init(cell(1,2,b)). init(cell(1,3,b)).
6  init(cell(2,1,b)). init(cell(2,2,b)). init(cell(2,3,b)).
7  init(cell(3,1,b)). init(cell(3,2,b)). init(cell(3,3,b)).
8
9  legal(xplayer,mark(M,N)) :- true(control(xplayer)), true(cell(M,N,Z)),
10                               distinct(Z,x), not true(tried(xplayer,M,N)).
11 legal(oplayer,mark(M,N)) :- true(control(oplayer)), true(cell(M,N,Z)),
12                               distinct(Z,o), not true(tried(oplayer,M,N)).
13 legal(xplayer,noop)         :- true(control(oplayer)).
14 legal(oplayer,noop)         :- true(control(xplayer)).
```

Líneas 1-7 : Se establecen los dos roles y las posiciones iniciales.

Líneas 9-14 : Se especifican los movimientos

---

<sup>3</sup>Variante del clásico Tic Tac Toe donde los jugadores no pueden percibir los movimientos de su oponente

# Ejemplo (continuación)

```
16 validmove          :- does(R,mark(M,N)), true(cell(M,N,b)).
17
18 next(F)             :- not validmove, true(F).
19 next(tried(R,M,N))  :- not validmove, does(R,mark(M,N)).
20
21 next(cell(M,N,x))    :- validmove, does(xplayer,mark(M,N)).
22 next(cell(M,N,o))    :- validmove, does(oplayer,mark(M,N)).
23 next(cell(M,N,Z))    :- validmove, true(cell(M,N,Z)),
24                        does(R,mark(I,J)), distinct(M,I).
25 next(cell(M,N,Z))    :- validmove, true(cell(M,N,Z)),
26                        does(R,mark(I,J)), distinct(N,J).
27 next(control(oplayer)) :- validmove, true(control(xplayer)).
28 next(control(xplayer)) :- validmove, true(control(oplayer)).
29 next(tried(R,M,N))   :- validmove, true(tried(R,M,N)).
30
31 sees(R, yourmove)    :- not validmove, true(control(R)).
32 sees(xplayer,yourmove) :- validmove, true(control(oplayer)).
33 sees(oplayer,yourmove) :- validmove, true(control(xplayer)).
```

Líneas 18-29 : Se especifica cómo se actualizarán las posiciones.

Líneas 31-33 : Se notificará a cada jugador cuando sea su turno.

Omitimos las condiciones de terminación y puntuación.

Sea  $G$  una descripción válida para un juego en GDL-II.

Se obtiene un único modelo para  $G$  utilizando el concepto de **modelo estable**.<sup>4</sup>

La semántica de  $G$  está dada por un sistema de transición de estados, construido a partir del modelo estable.

---

<sup>4</sup>El concepto de modelo estable fue introducido por M. Gelfond y V. Lifschitz en su paper *The Stable Model Semantics for Logic Programming* para proveer una semántica para programas lógicos con negación.



## Situation Calculus

Para construir sistemas que sean capaces de razonar con información imperfecta, se utiliza una extensión de **Situation Calculus (SC)**, que permite representar el conocimiento de los agentes.

SC es una lógica de predicados creada por McCarthy diseñada para formalizar y automatizar razonamientos y acciones. Tiene elementos predefinidos que permiten:

- Indicar la situación inicial.
- Decir que algo es cierto en determinada situación.
- Decir que una acción es posible en una situación.

## ¿Cómo razonar a partir de GDL-II?

Se utiliza una función que mapea elementos de GDL-II en elementos de esta extensión de SC.

Este paper fue escrito en 2014 por Stephan Schiffel (Reikjavík University, Islandia) y Michael Thielscher (The University of New South Wales, Australia).

Actualmente continúan investigando en el área.

Anualmente la AAAI (Association for the Advancement of Artificial Intelligence) organiza la International General Game Playing Competition, en la que participan equipos de investigación de distintos lugares del mundo.

Se realiza investigación en este área en países como Estados Unidos, Alemania y Australia.



Stephan Schieffel, Michael Thielscher (2014)

Representing and Reasoning About the Rules of General Games With Imperfect Information

*Journal of Artificial Intelligence Research* 171 – 206