

# Creación de un cliente para razonar en juegos de propósito general

**Autor:** Ignacio Navarro Martín

**Tutor:** Carlos Linares López

Ingeniería Informática  
Universidad Carlos III de Madrid

23 de octubre de 2009

# Índice

- 1 **Introducción**
  - Competiciones de juegos
  - General Game Playing
    - General Description Language
    - Competiciones Previas

- 2 **Objetivos**

- 3 **Desarrollo**

- Creación Agente GGP
  - Estructura Agente GGP
  - Elección del algoritmo de búsqueda
  - Obtención de la información
- Crítica Lenguaje GDL

- 4 **Pruebas y Resultados**

- Pruebas y Resultados

- 5 **Conclusiones**

- 6 **Líneas Futuras**

# Índice

- 1 **Introducción**
  - Competiciones de juegos
  - General Game Playing
    - General Description Language
    - Competiciones Previas

- 2 **Objetivos**

- 3 **Desarrollo**

- Creación Agente GGP
  - Estructura Agente GGP
  - Elección del algoritmo de búsqueda
  - Obtención de la información
- Crítica Lenguaje GDL

- 4 **Pruebas y Resultados**

- Pruebas y Resultados

- 5 **Conclusiones**

- 6 **Líneas Futuras**

# Competiciones de juegos

## Objetivo

Uno de los objetivos de la inteligencia artificial (IA) es encontrar un sistema capaz de razonar de forma inteligente.

## Definición

Decimos que un sistema es capaz de razonar si es capaz de resolver un problema planteado en cualquier dominio.

# Competiciones de juegos

## Objetivo

Uno de los objetivos de la inteligencia artificial (IA) es encontrar un sistema capaz de **razonar** de forma inteligente.

## Definición

Decimos que un sistema es capaz de razonar si es capaz de resolver un problema planteado en cualquier dominio.

# Competiciones de juegos

## Objetivo

Uno de los objetivos de la inteligencia artificial (IA) es encontrar un sistema capaz de **razonar** de forma inteligente.

## Definición

Decimos que un sistema es capaz de razonar si es capaz de resolver un problema planteado en cualquier dominio.

# Competiciones de juegos

## Objetivo

Uno de los objetivos de la inteligencia artificial (IA) es encontrar un sistema capaz de **razonar** de forma inteligente.

## Definición

Decimos que un sistema es capaz de razonar si es capaz de resolver un problema planteado en **cualquier** dominio.

# Competiciones de juegos

## Objetivo

Uno de los objetivos de la inteligencia artificial (IA) es encontrar un sistema capaz de **razonar** de forma inteligente.

## Definición

Decimos que un sistema es capaz de razonar si es capaz de resolver un problema planteado en algunos dominios.



# Competiciones de juegos

## Objetivo

Uno de los objetivos de la inteligencia artificial (IA) es encontrar un sistema capaz de **razonar** de forma inteligente.

## Definición

Decimos que un sistema es capaz de razonar si es capaz de resolver un problema planteado en **algunos** dominios.

# Competiciones de juegos

## Problema

¿Como obtenemos estos sistemas?

## Solución

Realizando sistemas que participen en competiciones de juegos.

# Competiciones de juegos

## Problema

¿Como obtenemos estos sistemas?

## Solución

Realizando sistemas que participen en competiciones de juegos.

# Competiciones de juegos

## Ejemplos de competiciones y sistemas

- **Othello:** El ganador actual es logistello
- **Checkers:** El ganador mundial es chinook, por Jonathan Schaeffer
- **Ajedrez:** El mejor sistema que existe actualmente es Fritz 10

# Competiciones de juegos

## Ventajas de las competiciones de juegos

- Permiten el uso de gran variedad de técnicas.
- Fomentan la participación y la resolución de problemas en diferentes dominios.
- Se puede obtener información de los sistemas creados.

## Problema

¿Realmente son útiles estas competiciones?

# Competiciones de juegos

## Ventajas de las competiciones de juegos

- Permiten el uso de gran variedad de técnicas.
- Fomentan la participación y la resolución de problemas en diferentes dominios.
- Se puede obtener información de los sistemas creados.

## Problema

¿Realmente son útiles estas competiciones?

# Competiciones de juegos

## Ventajas de las competiciones de juegos

- Permiten el uso de gran **variedad de técnicas**.
- Fomentan la participación y la resolución de problemas en diferentes dominios.
- Se puede obtener información de los sistemas creados.

## Problema

¿Realmente son útiles estas competiciones?

# Competiciones de juegos

## Ventajas de las competiciones de juegos

- Permiten el uso de gran **variedad de técnicas**.
- Fomentan la participación y la resolución de problemas en **diferentes dominios**.
- Se puede obtener información de los sistemas creados.

## Problema

¿Realmente son útiles estas competiciones?



# Competiciones de juegos

## Ventajas de las competiciones de juegos

- Permiten el uso de gran **variedad de técnicas**.
- Fomentan la participación y la resolución de problemas en **diferentes dominios**.
- Se puede **obtener información** de los sistema creados.

## Problema

¿Realmente son útiles estas competiciones?

# General Game Playing

## Definición (2005)

Un sistema General Game Playing (GGP) es aquel que puede soportar la definición formal de un juego (las reglas que lo definen) **sin conocerlo previamente** y jugarlo eficazmente **sin intervención** de un ser humano para ganarlo.

# General Game Playing



# General Game Playing



# General Game Playing



5	3			7			
6			1	9	5		
	9	8					6
8				6			3
4			8		3		1
7				2			6
	6					2	8
			4	1	9		5
				8			7
							9

# General Game Playing

## Estructura

La GGP se compone de los siguientes componentes:

- **GameMaster**: Unidad central que recibe movimientos, los procesa y actualiza el estado actual del juego.
- **Game Description Language (GDL)**: Lenguaje formal usado para definir juegos.

# Game Description Language

## Lenguaje GDL

Lenguaje usado para definir juegos en GGP, basado en hechos y reglas. Es una variante de Datalog similar a Prolog.

# Game Description Language

## Restricciones GDL

Existen algunas restricciones tanto por el lenguaje como por la competición:

- Juegos deterministas
- Información completa
- Finitos
- Débilmente ganable



# Competiciones Previas

## Estructura

Las competidores previos fueron:

- **ClunePlayer (2005)**: Basado en árboles MiniMax con poda alfa-beta.
- **FluxPlayer (2006)**: Basado en Prolog con funciones heurísticas. Además utiliza lógica borrosa.
- **CadiaPlayer(2007 - 2008)**: Se basa en técnicas de Monte Carlo como UCT.

# Índice

- 1 **Introducción**
  - Competiciones de juegos
  - General Game Playing
    - General Description Language
    - Competiciones Previas

- 2 **Objetivos**

- 3 **Desarrollo**

- Creación Agente GGP
  - Estructura Agente GGP
  - Elección del algoritmo de búsqueda
  - Obtención de la información
- Crítica Lenguaje GDL

- 4 **Pruebas y Resultados**

- Pruebas y Resultados

- 5 **Conclusiones**

- 6 **Líneas Futuras**

# Objetivos

## Objetivos

Los objetivos son:

- Estudiar como desarrollar un sistema GGP.
- Comprender el lenguaje GDL y la competición y mejorarlos si fuera necesario.
- Obtener técnicas, funciones y algoritmos para competir en la GGP.
- Crear un agente para competir en la GGP.

# Índice

- 1 **Introducción**
  - Competiciones de juegos
  - General Game Playing
    - General Description Language
    - Competiciones Previas

- 2 **Objetivos**

- 3 **Desarrollo**

- Creación Agente GGP
  - Estructura Agente GGP
  - Elección del algoritmo de búsqueda
  - Obtención de la información
- Crítica Lenguaje GDL

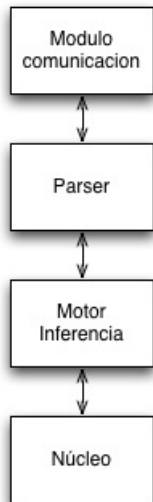
- 4 **Pruebas y Resultados**

- Pruebas y Resultados

- 5 **Conclusiones**

- 6 **Líneas Futuras**

# Estructura Agente GGP

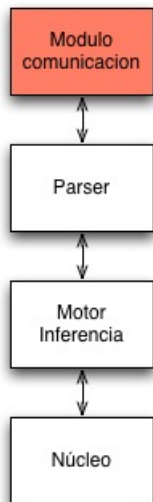


## Estructura

La mayor parte de los sistemas tienen los mismos componentes:

- **Módulo de comunicación:** Envía y recibe paquetes HTTP.
- **Parser:** Transforma la información GDL en estructuras para el sistema y viceversa.
- **Motor de inferencia:** Encargado de permitir las transiciones entre estados ejecutando acciones.
- **Núcleo del sistema:** Ejecuta los algoritmos que definen al sistema.

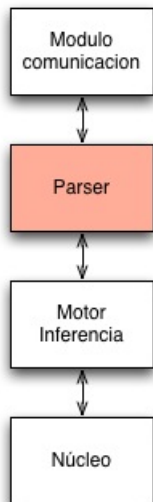
# Estructura Agente GGP



## Módulo de comunicación

Realiza la comunicación entre el sistema y el GameMaster usando los paquetes HTTP definidos previamente.

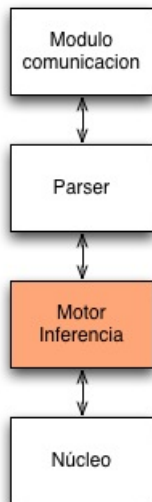
# Estructura Agente GGP



## Parser

Convierte los mensajes recibidos en estructuras internas para la aplicación.

# Estructura Agente GGP

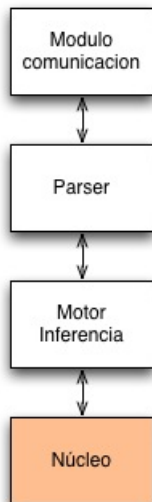


## Motor de inferencia

Crea una base de conocimiento y provee de métodos para su acceso y uso.



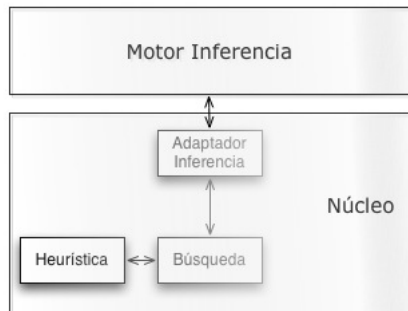
# Estructura Agente GGP



## Núcleo del sistema

Selecciona una acción de las disponibles en el estado actual.

# Estructura Agente GGP



## Núcleo del sistema

El núcleo del agente se compone de los siguientes elementos:

- Adaptador inferencia
- Heurística
- Búsqueda

# Algoritmo de búsqueda

## Supuestos

Se han realizado los siguientes supuestos:

- **Oponentes no inteligentes:** Supones que los oponentes no son inteligentes.
- **Paranoico:** En caso de existir más de dos jugadores se supondrá que todos conspiran contra el agente.
- **Juegos por turnos:** Se tratarán todos los juegos como juegos por turnos.

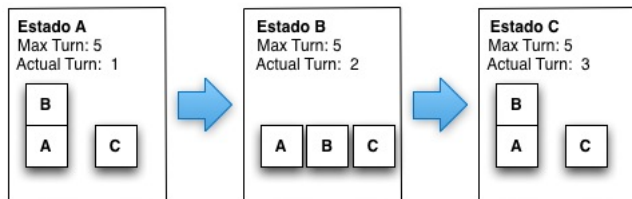
# Algoritmo de búsqueda

## Búsqueda contra la naturaleza

Las ventajas de usar el algoritmo **Learning RTA\*** son:

- Este algoritmo garantiza que converge a una solución.
- Alcanza soluciones en entornos con información limitada.
- Obtiene soluciones admisibles.
- Los consumos de memoria y/o tiempo permiten implementar el algoritmo.
- Permite retroceder para encontrar una solución

# Algoritmo de búsqueda



## Problema

Dos estados iguales son diferentes según la competición de GGP.

# Algoritmo de búsqueda

## Problema

Si los oponentes no son inteligentes, ¿nos podemos guiar por ellos?

## Solución

Nos guiamos por la estabilidad basándonos en la **robustez de la información**.

# Algoritmo de búsqueda

## Problema

Si los oponentes no son inteligentes, ¿nos podemos guiar por ellos?

## Solución

Nos guiamos por la estabilidad basándonos en la **robustez de la información**.

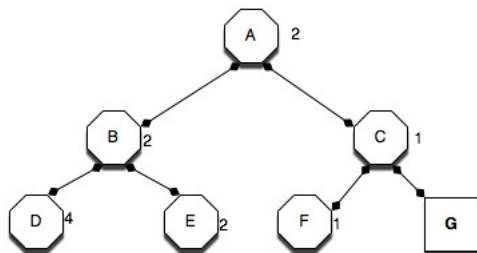
# Algoritmo de búsqueda

## Definición

Una posición estable es aquella que tiene menos probabilidad de cambiar.



# Algoritmo de búsqueda



## Búsqueda Multiagente

El algoritmo de **números conspiratorios** nos indica la estabilidad de cada nodo.

# Obtención de información

## Obtención de información

Para guiar las búsquedas es necesario dar cierta información.

## Problema

No sabemos el dominio sobre el que trabajamos.

# Obtención de información

## Obtención de información

Para guiar las búsquedas es necesario dar cierta información.

## Problema

No sabemos el dominio sobre el que trabajamos.

# Obtención de información

## Ejemplo

```
(at k1)
(s k1 k2)
(s k2 k3)
...
(s kn-1 kn)
```

## Reconocimiento de patrones

Identificar estructuras temporales nos aporta más información.

# Obtención de información

## Simulaciones

Se realizan simulaciones para obtener valores a usar en la función heurística:

- Profundidad máxima
- Profundidad media
- Estabilidad profundidad
- Ramificación máxima
- Ramificación media
- Nodos terminales
- Nodos árbol
- Tamaño medio solución

# Obtención de información

## Función heurística

$$h(x) = c * \sum_{i=1}^n f_i(x, y) * \alpha_i$$

Además debe cumplirse la siguiente condición:

$$1 = \sum_{i=1}^n \alpha_i$$

## Simulaciones

Las funciones a ajustar para obtener la función heurística son

- Función de ramificación
- Función de profundidad
- Similitud de tamaño

# Crítica Lenguaje GDL

## Problema

¿Se pueden representar todos los juegos en GDL?

## Excepciones

Existen algunas características no representables:

- Finitud de juegos
- Determinismo
- Información imperfecta
- Débilmente ganable

# Crítica Lenguaje GDL

## Problema

¿Se pueden representar todos los juegos en GDL?

## Excepciones

Existen algunas características no representables:

- Finitud de juegos
- Determinismo
- Información imperfecta
- Débilmente ganable



# Crítica Lenguaje GDL

## Problema

¿Se pueden representar todos los juegos en GDL?

## Excepciones

Existen algunas características no representables:

- Finitud de juegos
- **Determinismo**
- **Información imperfecta**
- Débilmente ganable

## Juegos No deterministas

Un juego no determinista es aquel que dado un estado y tomada una acción no sabes el estado de llegada.

# Crítica Lenguaje GDL

## Ejemplo

```
(init (ude 1 3))

(<= (next (seq ?y ?z))
    (true (seq ?x ?k))
    (con ?x ?y ?z))

(ite 1 2 5)
(ite 2 3 7)
(ite 3 4 9)
(ite 4 1 8)
```

## Juegos No deterministas

Una estructura pseudoaleatoria no necesita cambiar nada del lenguaje/competición, pero esta a la vista de los jugadores.

# Crítica Lenguaje GDL

## Juegos No deterministas

Un hecho aleatorio soluciona el problema de la pseudoaleatoriedad y visibilidad, pero se debe cambiar el servidor.

## Ejemplo

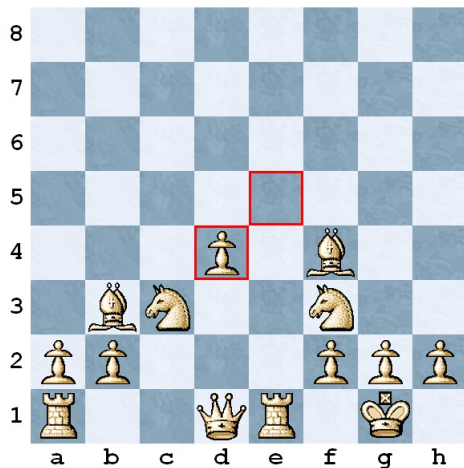
```
(<= (next (rfact ?x))  
     (does randomplayer (rmove ?x)))
```

# Crítica Lenguaje GDL

## Información imperfecta

En un juego de información imperfecta, ciertos hechos, jugadores o acciones están ocultos. No confundir con no determinismo.

# Crítica Lenguaje GDL



## Información imperfecta

Kriegspiel es una variante del ajedrez donde cada jugador desconoce algunas piezas.

# Crítica Lenguaje GDL

## Información imperfecta

La solución planteada implica ocultar reglas o hechos a ciertos jugadores. Creando la relación nueva **mark-info**.

## Solución

```
(mark-info (players) (rules/facts))
```

# Crítica Lenguaje GDL

## Información imperfecta

Un ejemplo de esto para el caso del Kriegspiel sería:

## Ejemplo

```
(mark-info black-player ( (cell a 1 br)
                          (cell d 3 br)))

(mark-info (black-player)
  (<= (next (cell ?p ?x ?y))
    (mark-info white-player (next (cell ?u ?v b)))
      (does ?player (move ?p ?u ?v ?x ?y))))
)
```



# Crítica Lenguaje GDL

## Información imperfecta

Una posible solución para evitar el número de intentos infinitos es limitarlos.

## Solución

```
(n-illegal 5)
```

# Índice

- 1 **Introducción**
  - Competiciones de juegos
  - General Game Playing
    - General Description Language
    - Competiciones Previas

- 2 **Objetivos**

- 3 **Desarrollo**

- Creación Agente GGP
  - Estructura Agente GGP
  - Elección del algoritmo de búsqueda
  - Obtención de la información
- Crítica Lenguaje GDL

- 4 **Pruebas y Resultados**

- Pruebas y Resultados

- 5 **Conclusiones**

- 6 **Líneas Futuras**

# Pruebas y Resultados

## Pruebas

El agente se probará primero contra:

- Agentes pequeños para testear su validez
- Agentes en la competición GGP

# Pruebas y Resultados

## Agentes

Los agentes usados son:

- LegalPlayer
- RandomPlayer
- Minimax
- **Hex**

## Dominios

Los dominios usados son:

- **1 Jugador:** Mundo de los bloques
- **2 Jugadores:** Tic-Tac-Toe (Tres en raya)
- **4 Jugadores:** Othello 4

# Pruebas y Resultados

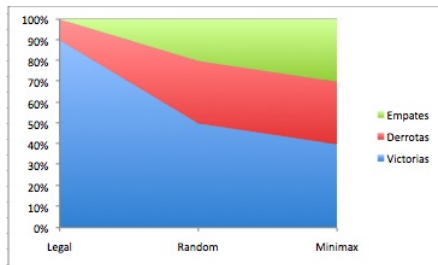


Figura: Tic-Tac-Toe

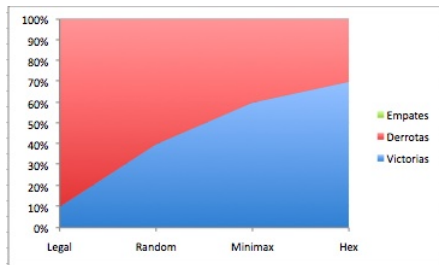


Figura: Othello 4

# Pruebas y Resultados

Posición	Nombre	Puntuación	Partidos Jugados	Partidos sin error	Tasa error
1	CadiaPlayerlite	70.1856	862	848	0.0162
2	ary	66.9859	924	899	0.0271
3	TurboTurtle	56.1768	164	136	0.1707
4	Fluxplayer	52.2034	875	678	0.2251
5	centurio	49.3675	721	647	0.1026
6	Gamer	31.0576	330	171	0.4818
7	Hex	0.1053	19	2	0.8947
8	Ethan	0	0	0	0
9	Maligne	0	0	0	0

CUADRO 5.4: Resultados de las pruebas clasificatorias GGP Competition 2009

# Pruebas y Resultados

Posición	Nombre	Puntuación
1	ary-distant	339
2	Fluxplayer	256
3	Maligne	255
4	Centurio	153
5	Ethos	318
6	CadiaPlayer	306
7	Gamer	261
8	TurboTurtle	143

CUADRO 5.5: Resultados de la GGP Competition 2009

# Índice

- 1 **Introducción**
  - Competiciones de juegos
  - General Game Playing
    - General Description Language
    - Competiciones Previas

- 2 **Objetivos**

- 3 **Desarrollo**

- Creación Agente GGP
  - Estructura Agente GGP
  - Elección del algoritmo de búsqueda
  - Obtención de la información
- Crítica Lenguaje GDL

- 4 **Pruebas y Resultados**

- Pruebas y Resultados

- 5 **Conclusiones**

- 6 **Líneas Futuras**



# Conclusiones

## Resumen

- Se ha creado un sistema GGP que compitió en la fase previa de la competición GGP 2009
- Se ha realizado una recopilación de todos los materiales necesarios para desarrollar la competición
- Además, se realizó un estudio sobre el lenguaje GDL
- Se realizó un estudio de cómo crear un agente y qué método de búsqueda fue el más adecuado

# Conclusiones

## Conclusiones

- El objetivo de crear un agente todavía está demasiado abierto
- Existen muchas formas de crear un agente que no han sido investigadas

# Índice

- 1 **Introducción**
  - Competiciones de juegos
  - General Game Playing
    - General Description Language
    - Competiciones Previas

- 2 **Objetivos**

- 3 **Desarrollo**

- Creación Agente GGP
  - Estructura Agente GGP
  - Elección del algoritmo de búsqueda
  - Obtención de la información
- Crítica Lenguaje GDL

- 4 **Pruebas y Resultados**

- Pruebas y Resultados

- 5 **Conclusiones**

- 6 **Líneas Futuras**

# Líneas Futuras

## Líneas

- Creación de una función heurística general.
- Ampliación de GDL y la competición GGP.
- Estudio de funciones de búsqueda.