



Proyecto de curso #2
INTELIGENCIA ARTIFICIAL

Yoshi's world

Estudiantes:

Ervin Caravall Ibarra
Hernan David Cisneros Vargas
Miguel Angel Moreno Romero
Kevin Alejandro Velez Agudelo

Profesor:

Oscar Bedoya

Escuela de Ingeniería de Sistemas y Computación
Mayo de 2024

Tabla de contenido

1. Introducción	3
2. Explicación de la Función de Utilidad Heurística y su admisibilidad.	4
2.1 Definición	4
2.2 Propósito	4
2.3 Demostrac	ión de Admisibilidad
	5
3. Conclusiones	6

1. Introducción

El algoritmo Minimax es una técnica fundamental en la inteligencia artificial, especialmente en el ámbito de los juegos de dos jugadores de suma cero, como el ajedrez, las damas o como el caso del proyecto actual, los movimientos en el juego de yoshi's world, el cual representa un desafío significativo para la inteligencia artificial debido a su naturaleza de suma cero y su complejidad estratégica.

Para este proyecto, se ha desarrollado e implementado el algoritmo Minimax con tres niveles de dificultad: principiante, amateur y experto. Cada nivel de dificultad ofrece una experiencia de juego diferente, con estrategias de decisión adaptadas para desafiar a jugadores de distintos niveles de habilidad.

Se presentarán los resultados del proceso de construcción del algoritmo Minimax desarrollado para este segundo proyecto, su funcionamiento y la capacidad o no de resolver el problema de Yoshi's world en diferentes escenarios de prueba, y un análisis posterior de resultados de ejecución. Además, se prestará especial atención en detallar la función de utilidad heurística utilizada en el algoritmo, explicando su definición, propósito y aplicación en la toma de decisiones del juego propuesto.

Se comparte el enlace hacia el repositorio del proyecto, donde se podrá visualizar el trabajo colaborativo y el control de versiones:

<https://github.com/Kvel12/Proyecto-IAII.git>

2. Explicación de la Función de Utilidad Heurística y su admisibilidad.

La función de utilidad heurística es crucial para el algoritmo Minimax, ya que proporciona una estimación del valor de un estado del juego cuando no se puede calcular un valor exacto debido a la profundidad limitada de la búsqueda.

2.1 Definición

En el contexto de nuestro juego, la función de utilidad heurística está definida de la siguiente forma:

```
def evaluar_tablero(juego):  
    return juego['puntuacion']['verde'] - juego['puntuacion']['rojo']
```

Esta función calcula la diferencia entre la puntuación de Yoshi verde y la puntuación de Yoshi rojo. Si el valor es positivo, indica que Yoshi verde está en una posición favorable, mientras que un valor negativo indica que Yoshi rojo está en una posición favorable.

2.2 Propósito

El propósito de la función de utilidad heurística es proporcionar una medida rápida y aproximada de qué tan bueno es un estado del juego para el jugador maximizador (Yoshi verde). Esta evaluación permite que el algoritmo Minimax tome decisiones informadas sin necesidad de explorar todos los estados posibles hasta el final del juego.

La admisibilidad de una heurística en el contexto de los algoritmos de búsqueda se refiere a que la heurística nunca sobreestima el costo real para alcanzar el objetivo. para nuestro juego en cuestión, la función de utilidad heurística es admisible porque cumple con la siguiente propiedad:

- No sobreestimación del valor real: La función **evaluar_tablero(juego)** simplemente calcula la diferencia entre las puntuaciones de los dos jugadores. No sobreestima el valor del estado del juego, ya que solo considera la puntuación actual y no intenta predecir futuras ganancias o pérdidas que puedan ser inciertas.

La heurística utilizada garantiza que el costo estimado desde cualquier estado hasta el objetivo (maximizar la puntuación para Yoshi verde y minimizar para Yoshi rojo) nunca es mayor que el costo real, proporcionando así una base sólida para que el algoritmo Minimax tome decisiones informadas y estratégicas en cada paso del juego

2.3 Demostración de Admisibilidad

La admisibilidad de una heurística en algoritmos de búsqueda implica que la heurística no sobreestima el costo real para alcanzar el objetivo. En nuestro juego, esto significa que la heurística utilizada no sobreestima la ventaja de un jugador sobre otro.

Para demostrar la admisibilidad de la heurística ``evaluar_tablero(juego)``, debemos demostrar que para cualquier estado s del juego, el valor de la heurística $h(s)$ no sobreestima la diferencia real en puntuaciones entre Yoshi verde y Yoshi rojo desde el estado s hasta un estado objetivo (estado final del juego).

Propiedad de Admisibilidad:

$$h(s) \leq h^*(s)$$

donde $h(s)$ es el valor estimado por la heurística en el estado s y $h^*(s)$ Es el valor real de la diferencia en puntuaciones desde s hasta el estado objetivo.

Evaluación de la Heurística:

La heurística está definida como:

$$h(s) = \text{juego}[\text{'puntuación'}][\text{'verde'}] - \text{juego}[\text{'puntuación'}][\text{'rojo'}]$$

Para cualquier estado s , esta heurística mide la diferencia actual en puntuaciones entre Yoshi verde y Yoshi rojo.

Análisis de Admisibilidad:

- Diferencia Actual en Puntuaciones:

La heurística calcula la diferencia actual en puntuaciones:

$$h(s) = \text{juego}[\text{'puntuación'}][\text{'verde'}] - \text{juego}[\text{'puntuación'}][\text{'rojo'}]$$

Este valor representa la ventaja actual de Yoshi verde sobre Yoshi rojo.

- Comparación con el Valor Real:

Dado que la heurística sólo considera las puntuaciones actuales y no futuras, el valor estimado $h(s)$ es siempre menor o igual al valor real $h^*(s)$, que incluiría cualquier futura ganancia de puntuación hasta el final del juego. Esto se puede expresar como:

$$h(s) \leq h^*(s)$$

donde $h^*(s)$ Es la diferencia real en puntuaciones desde el estado (s) hasta el estado objetivo.

3. Conclusiones

Dado que la heurística utilizada en nuestro juego mide la diferencia actual en puntuaciones sin hacer suposiciones sobre futuras ganancias o pérdidas, no sobreestima la ventaja de Yoshi verde. Por lo tanto, cumple con la propiedad de admisibilidad. Esto asegura que el algoritmo Minimax (incluyendo la técnica de poda alfa-beta), utilizando esta heurística, toma decisiones informadas y óptimas en cada paso del juego, evaluando correctamente el estado actual sin sobrevalorar la ventaja del jugador maximizador.

La heurística *evaluar_tablero(juego)* Es admisible porque proporciona una estimación que es siempre menor o igual al valor real de la diferencia en puntuaciones, garantizando que el algoritmo Minimax funcione de manera efectiva y precisa.

		A				
						B