



TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE TLAXIACO

INTELIGENCIA ARTIFICIAL

REPORTE DE INVESTIGACION UNIDAD 3

CARRERA:

INGENIERÍA EN SISTEMAS COMPUTACIONALES

SEMESTRE:

8US

PRESENTA:

MARÍA JOSE LÓPEZ LEYVA N. CONTROL 20620143

DOCENTE

OSORIO SALINAS EDWARD

Tlaxiaco, Oax., 05 de Mayo del 2024.



INTRODUCCIÓN

En el ámbito de la inteligencia artificial, la resolución de problemas mediante la búsqueda en espacios de estados es un enfoque fundamental. Este enfoque se basa en la representación del problema como un espacio de estados, donde cada estado representa una posible configuración del problema y las acciones disponibles representan las transiciones entre estados. La búsqueda en este espacio de estados tiene como objetivo encontrar una secuencia de acciones que conduzca desde el estado inicial al estado objetivo, que representa la solución al problema.

REGLAS Y BUSQUEDA

3.2 Espacios de estados determinísticos y espacios no determinísticos.

El espacio de estados es la representación de un problema que abarca todas las posibles situaciones que se pueden presentar en la solución del problema.

Los espacios de estados se clasifican en dos categorías principales:

- + Determinísticos
- + No determinísticos

Espacios de estados determinísticos: En un espacio de estados determinista, el resultado de aplicar una acción en un estado es único y predecible. Es decir, para cada estado y acción existe un único estado sucesor. Esto significa que la evolución del sistema es completamente predecible y no hay elementos de aleatoriedad. El espacio de estados determinísticos contiene un único estado inicial y seguir la secuencia de estados para la solución. Los espacios de estados determinísticos son usados por los sistemas expertos. Se puede describir a su vez, que un sistema es determinístico si, para un estado dado, al menos aplica una regla a él y de solo una manera.

Ejemplo: Un juego de ajedrez es un ejemplo de un problema con un espacio de estados determinista. Cada movimiento del jugador representa una acción, y el estado del juego cambia de manera determinista en función de la pieza movida y la posición inicial de las piezas en el tablero.

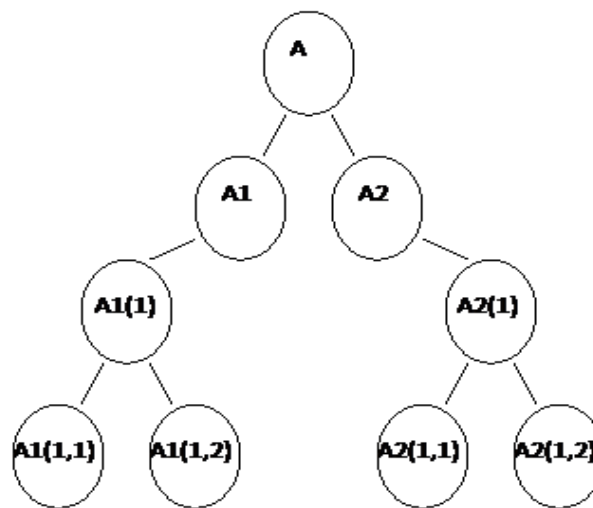


Ilustración 1 Ejemplo de espacio de estados determinísticos.

Características:

- Los espacios de estados determinísticos son usados por los sistemas expertos, mientras que los no determinísticos son usados por sistemas de lógica difusa.
- Un ambiente no determinístico presente problemas más grandes para el agente.

Espacios de estados no determinísticos: En un espacio de estados no determinista, el resultado de aplicar una acción en un estado puede ser no único o incluso aleatorio. Es decir, para un mismo estado y acción puede haber varios estados sucesores posibles. Esto significa que la evolución del sistema no es completamente predecible y puede haber elementos de aleatoriedad. El no determinístico contiene un amplio número de estados iniciales y sigue la secuencia de estados perteneciente al estado inicial del espacio. Son usados

por sistemas de lógica difusa. En otras palabras, si más de una regla aplica a cualquier estado particular del sistema, o si una regla aplica a un estado particular del sistema en más de una manera, entonces el sistema es no determinístico.

Ejemplo: Un juego de cartas como el póker es un ejemplo de un problema con un espacio de estados no determinista. El reparto de las cartas al inicio del juego es un proceso aleatorio, lo que significa que el estado inicial del juego no es determinista. Además, las decisiones de los jugadores durante el juego también pueden introducir elementos de aleatoriedad.

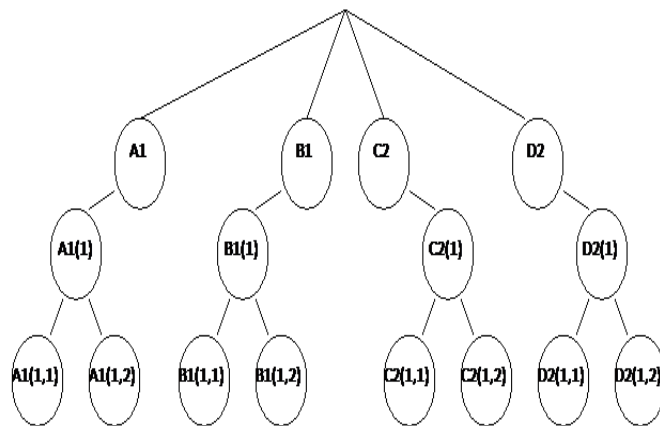


Ilustración 2 Ejemplo de espacio de estados no determinísticos.

Clasificación de los espacios de estados:

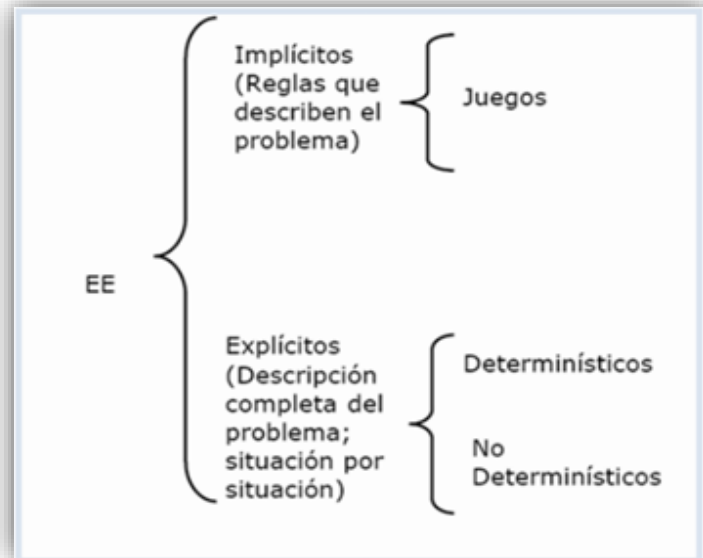


Ilustración 3 Implícitos y explícitos

Los espacios de estados implícitos normalmente utilizan un sistema de producción para generar sobre la marcha los posibles estados siguientes de un estado dado. Los juegos suelen crear un espacio de estados implícito ya que un juego puede variar dependiendo de las reglas que lo describen.

Los espacios de estados explícitos son aquellos en los se define, previo al inicio de la búsqueda, todos los estados posibles y las conexiones entre ellos.

3.3 Búsqueda sistemática

La búsqueda sistemática es un método de búsqueda en espacios de estados que explora todos los estados del espacio de búsqueda de manera ordenada y exhaustiva. Este tipo de búsqueda garantiza encontrar una solución al problema si existe una, pero puede ser computacionalmente costosa para problemas con grandes espacios de estados.

3.3.1 Búsqueda de metas a profundidad

La búsqueda de metas a profundidad (DFS por sus siglas en inglés) es un algoritmo de búsqueda sistemática que explora el espacio de estados siguiendo un camino único a partir del estado inicial. El algoritmo explora un estado hijo antes de pasar a explorar sus hermanos. Si no se encuentra una



solución en el camino actual, se retrocede al estado padre y se explora otro camino posible.

Ejemplo: La búsqueda de metas a profundidad se puede utilizar para resolver el problema de encontrar un camino en un laberinto. El algoritmo comienza en la casilla inicial y explora las casillas adyacentes. Si no se encuentra la salida en la primera casilla adyacente, se retrocede a la casilla inicial y se explora la siguiente casilla adyacente, y así sucesivamente.

Si el conjunto open se maneja como una lista LIFO, es decir, como un stack, siempre se estará visitando primero los últimos estados en ser generados. Esto significa que si A genera B y C , y B genera D , antes de visitar C se visita D , que está más alejado de la raíz A , o sea más profundo en el árbol de búsqueda. El algoritmo tiene en este caso la tendencia de profundizar la búsqueda en una rama antes de explorar ramas alternativas.

```
procedure Búsqueda_en_profundidad {  
  open () [estado_inicial]  
  closed () {}  
  while (open no está vacía) {  
    remover el primer estado X de la lista open  
    if (X es un estado objetivo) return éxito  
    else {  
      generar el conjunto de sucesores del estado X  
      agregar el estado X al conjunto closed  
      eliminar sucesores que ya están en open o en closed  
      agregar el resto de los sucesores al principio de open  
    }  
  }  
  return fracaso  
}
```

Considerando que la cantidad promedio de sucesores de los nodos visitados es B (llamado en inglés el *branching factor* y en castellano el *factor de ramificación*), y suponiendo que la profundidad máxima alcanzada es n , este algoritmo tiene una complejidad en tiempo de $O(B^n)$ y, si no se considera el conjunto *closed*, una complejidad en espacio de $O(B \times n)$. En vez de usar el conjunto *closed*, el control de ciclos se puede hacer descartando aquellos estados que aparecen en el camino generado hasta el momento (basta que cada estado generado tenga un puntero a su padre). El mayor problema de este algoritmo es que puede "perdersse" en una rama sin encontrar la solución.



Además, si se encuentra una solución no se puede garantizar que sea el camino más corto.

3.3.2 Búsqueda de metas en anchura

La búsqueda de metas en anchura (BFS por sus siglas en inglés) es otro algoritmo de búsqueda sistemática que explora el espacio de estados nivel por nivel. El algoritmo explora todos los estados del nivel actual antes de pasar al siguiente nivel. Si se encuentra una solución en cualquier nivel, el algoritmo se detiene.

Ejemplo: La búsqueda de metas en anchura también se puede utilizar para resolver el problema de encontrar un camino en un laberinto. El algoritmo comienza en la casilla inicial y explora todas las casillas adyacentes a la casilla inicial. Luego, explora todas las casillas adyacentes a las casillas adyacentes a la casilla inicial, y así sucesivamente.

Si el conjunto open se maneja como una lista FIFO, es decir, como una cola, siempre se estará visitando primero los primeros estados en ser generados. El recorrido del espacio de estados se hace por niveles de profundidad.

```
procedure Búsqueda_en_amplitud {  
  open ()[estado_inicial]  
  closed () {}  
  while (open no está vacía) {  
    remover el primer estado X de la lista open  
    if (X es un estado objetivo) return éxito  
    else {  
      generar el conjunto de sucesores del estado X  
      agregar el estado X al conjunto closed  
      eliminar sucesores que ya están en open o en closed  
      agregar el resto de los sucesores al final de open  
    }  
  }  
  return fracaso  
}
```

Si el factor de ramificación es B y la profundidad a la cual se encuentra el estado objetivo más cercano es n , este algoritmo tiene una complejidad en tiempo y espacio de $O(B^n)$. Contrariamente a la búsqueda en profundidad, la búsqueda en amplitud garantiza encontrar el camino más corto.



CONCLUSIÓN

La búsqueda en espacios de estados es una técnica fundamental en la inteligencia artificial para la resolución de problemas. Los espacios de estados determinísticos y no determinísticos representan dos tipos de problemas con diferentes características. La búsqueda sistemática, incluyendo la búsqueda de metas a profundidad y la búsqueda de metas en anchura, son métodos de búsqueda que exploran el espacio de estados de manera ordenada y exhaustiva. La elección del algoritmo de búsqueda adecuado depende de las características del problema y de los recursos computacionales disponibles.



REFERENCIAS

Unidad 2. Técnicas de búsqueda. (s. f.).

<http://inteligenciaartificialisc.blogspot.com/p/unidad-2.html>

GeeksforGeeks. (2024, 3 mayo). *Difference between Deterministic and Non-deterministic*

Algorithms. GeeksforGeeks. [https://www.geeksforgeeks.org/difference-between-](https://www.geeksforgeeks.org/difference-between-deterministic-and-non-deterministic-algorithms/)

[deterministic-and-non-deterministic-algorithms/](https://www.geeksforgeeks.org/difference-between-deterministic-and-non-deterministic-algorithms/)

<https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://hopel>

[chen.tecnm.mx/principal/sylabus/fpdb/recursos/r98095.PPTX&ved=2ahUKEwiVgoyam](https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://hopel)

[iFAxVU4MkDHU4LD3IQFnoECA8QAw&usg=AOvVaw0XU9xlhrtFCui51N03LjUr](https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://hopel)