Tema 2. Búsqueda en Espacios de Estados

Introducción y Modelado de Espacios de Búsqueda

Objetivos

1. **Conocer los fundamentos de los algoritmos de búsqueda y el papel que juegan en la Inteligencia Artificial**
2. Conocer el paradigma de Búsqueda en **Espacios de Estados** y los algoritmos básicos de búsqueda a ciegas y sobre todo de búsqueda inteligente o heurística
3. Saber cómo modelar **problemas** para resolverlos con Búsqueda en **Espacios de Estados**, en particular cómo introducir conocimiento específico del dominio del problema

Contenidos

1. **Introducción**
2. **Espacios de búsqueda**
3. Algoritmos de búsqueda no informada
4. Algoritmos de búsqueda informada o heurística
5. Técnicas de diseño de funciones heurísticas

2.1. Introducción

1. Ubicuidad de la búsqueda en Inteligencia Artificial

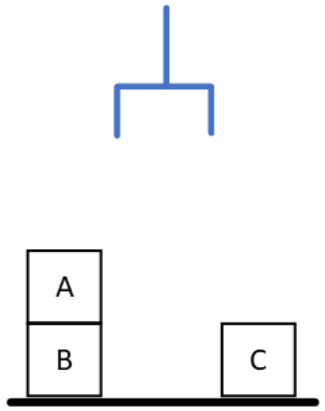
   El núcleo de un sistema inteligente es un algoritmo de búsqueda

2. Para resolver un problema con un sistema de búsqueda hay que establecer dos elementos principales
   - Un espacio de búsqueda: representa distintas formas de resolver el problema
   - Un algoritmo de búsqueda: trata de encontrar la mejor solución del espacio de búsqueda

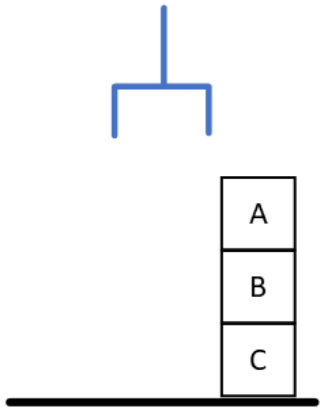
3. Ejemplos de problemas de búsqueda
   - *Planificación de actuaciones de robots*
   - *Cálculo de rutas*
   - *Reconocimiento de formas*
   - *Estrategias de juego*
   - *Cálculo de árboles de decisión*

Ejemplo: Robot en un Mundo de Bloques
Definición del problema

- Estado inicial



- Estado objetivo



- El objetivo es encontrar una secuencia mínima de acciones del robot que partiendo de la situación inicial nos permitan llegar a la situación objetivo, para ello es necesario
  - *Conocer lo que el agente (el robot) es capaz de entender, es decir el nivel de detalle de las acciones*
  - *Y definir un modelo de representación de las situaciones (estados) coherente con este nivel de detalle*

Problema: Robot en un Mundo de Bloques
Modelado de estados

- Suponemos que las acciones elementales del robot permiten generar situaciones como las siguientes

- En este caso, la descripción de los estados se puede hacer instanciando adecuadamente los siguientes predicados

```
SobreLaMesa(X)        ;; El bloque X está sobre la mesa
Sobre(X,Y)            ;; El bloque X está sobre el bloque Y
Libre(X)              ;; El bloque X no tiene otro bloque encima
Cogido(X)             ;; El robot tiene cogido el bloque X
ManoLibre             ;; La mano del robot está libre
```

Problema: Robot en un Mundo de Bloques
Modelado de acciones

- Teniendo en cuenta cómo es la representación de los estados y lo que es capaz de entender el agente (robot), las acciones elementales se pueden expresar mediante las reglas siguientes:

Desapilar(X,Y)

P: Sobre(X,Y),Libre(X),ManoLibre

E: ¬ Sobre(X,Y), ¬ ManoLibre, Cogido(X)

Apilar(X,Y)

P: Libre(Y), Cogido(X)

E: ¬ Libre(Y), ¬ Cogido(X), ManoLibre, Sobre(X,Y), Libre(X)

Coger(X)

P: SobreLaMesa(X), Libre(X), ManoLibre

E: ¬ SobreLaMesa(X), ¬Libre(X), ¬ ManoLibre

Posar(X)

P: Cogido(X)

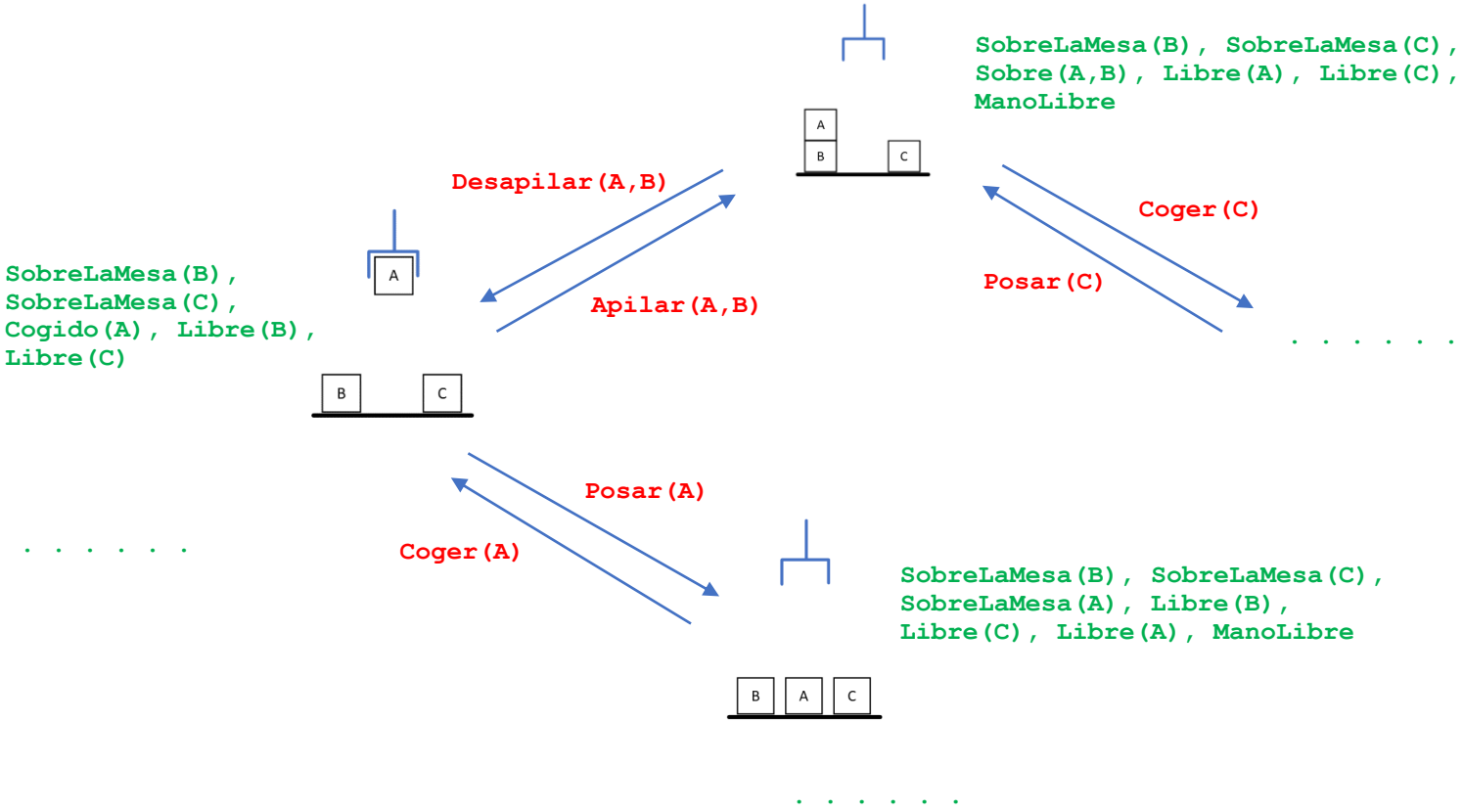
E: SobreLaMesa(X), ¬ Cogido(X), Libre(X), ManoLibre

- Cada acción se puede aplicar en los estados que cumplan los prerrequisitos (P:) y produce una serie de efectos (E:) en el nuevo estado con respecto al anterior.
- Dado que se trata de minimizar el número de acciones, el coste de aplicar una acción es 1 en todos los casos.

Problema: Robot en un Mundo de Bloques

Espacio de búsqueda para la instancia definida por la situación inicial anterior

- El espacio de búsqueda (una parte significativa) para esta instancia del problema es el siguiente, con indicación de la regla aplicada en cada transición, y la definición de cada estado



Problema: Robot en un Mundo de Bloques
Algoritmos de búsqueda adecuados al espacio anterior

- El espacio anterior tiene forma de grafo dirigido simple con costes positivos. En consecuencia, cualquier algoritmo de búsqueda en grafos que permita encontrar un camino entre el estado inicial y uno de los objetivos puede ser de utilidad
- Ejemplos:
  - Búsqueda primero en anchura
  - Búsqueda primero en profundidad
  - Algoritmo de Dijsktra
  - Backtracking
- Estos son "algoritmos de búsqueda a ciegas" ya que no utilizan información sobre el dominio del problema
- Existen otras opciones, los "algoritmos de búsqueda heurística o búsqueda inteligente" que sí pueden utilizar información específica del problema para mejorar el proceso de búsqueda de soluciones

2.2. Espacio de Búsqueda
Definición

- El espacio de búsqueda es un grafo simple dirigido con arcos etiquetados con costes positivos
  - Nodos: estados que representan subproblemas
    - 1 estado inicial
    - 1 ó más estados objetivo
  - Arcos: representan acciones elementales, operadores o reglas
    - Permiten pasar de un estado n_1 a un estado sucesor n_2
    - Tienen asociado un coste no negativo $c(n_1, accion, n_2)$ ó $c(n_1, n_2)$
  - Solución del problema
    - Es cualquier camino desde el estado inicial a uno de los estados objetivo
    - El coste de la solución es la suma de los costes de los arcos del camino

2.2. Espacio de Búsqueda
Características

- Es un modelo simplificado del problema
  - El nivel de detalle depende de las capacidades del agente que tiene que ejecutar las acciones
- No tiene que estar almacenado en una estructura, se genera a partir de las operaciones
  - EstadoInicial() ;; Genera un nodo con el estado inicial
  - Sucesores(n) ;; Calcula la lista de sucesores de un estado n, con los costes correspondientes
  - EsObjetivo(n) ;; Comprueba si un estado es o no objetivo
- En general es finito, pero puede no serlo
- Los estados son distintos, no hay estados repetidos
- Suponemos entornos simples
  - Estados totalmente observables y discretos (no continuos)
  - Acciones deterministas
- El Espacio de Búsqueda es independiente del Algoritmo de Búsqueda

Problema: Cálculo de rutas óptimas
Espacio de búsqueda

- Se trata de buscar la mejor ruta entre Arad y Bucharest en Rumanía

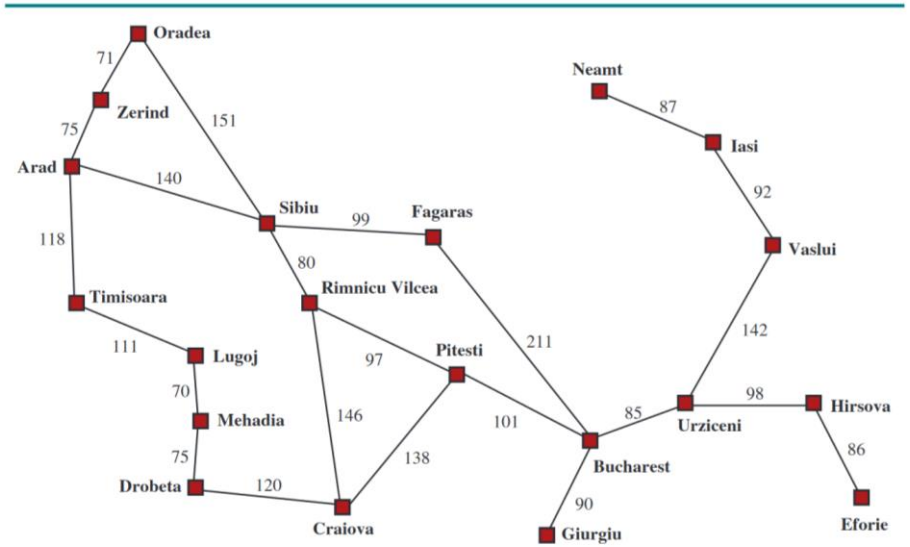
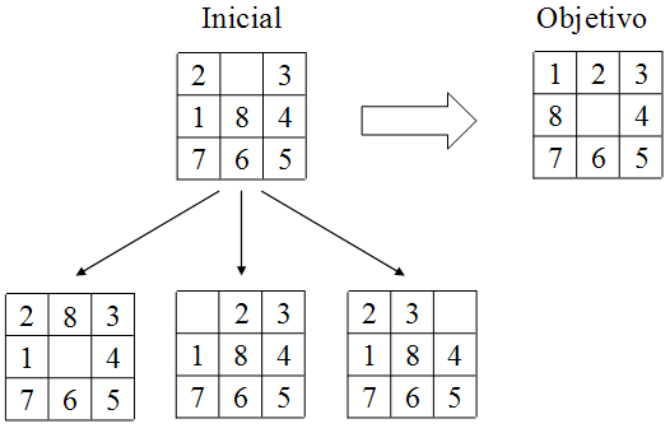


Figure 3.1 A simplified road map of part of Romania, with road distances in miles.

- Estados: ciudades en el mapa de Rumanía
  - EstadoInicial(): Arad
  - Sucesores(n): lista de ciudades conectadas con n
  - EsObjetivo(n): n es Bucharest
- Acciones: conexiones entre las ciudades
  - c(n1,n2) = distancia de la conexión directa entre n1 y n2

Problema: El problema del 8-puzzle
Espacio de búsqueda

- Dado un tablero de 3 × 3 posiciones con 8 fichas y una casilla vacía, se trata de buscar la secuencia mínima de movimientos para llegar desde una situación inicial a otra, objetivo, en la que las fichas están ordenadas como se indica en la figura
- Un movimiento elemental consiste en mover una ficha a la posición vacía si ésta es adyacente ortogonalmente



- Estados:
  - Situaciones posibles de las 8 fichas en el tablero
  - Hay un solo objetivo
- Acciones:
  - 2, 3 ó 4 para cada estado
  - El coste es 1 siempre ya que se trata de minimizar el número de movimientos

Otros problemas de búsqueda

- El problema del viajante de comercio
- El problema de las N-reinas
- El problema de los frixuelos (o pancakes) ordenados
- El problema de los misioneros y caníbales
- Cálculo de un árbol de expansión mínimo con grado limitado
- El problema de asignación cuadrática
- Coloreado de grafos
- Cálculo de la comunidad más pequeña en una red social
- . . . . .