

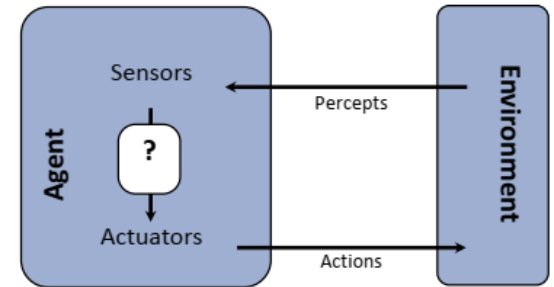
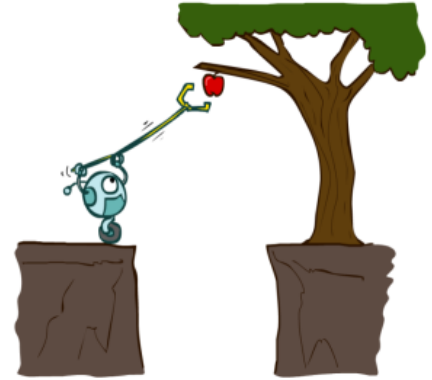
FORMULACIÓN DE PROBLEMAS

Inteligencia Artificial

Yomin E. Jaramillo Múnera, Msc. Automatización y Control Ind
vejaramilm@eafit.edu.co

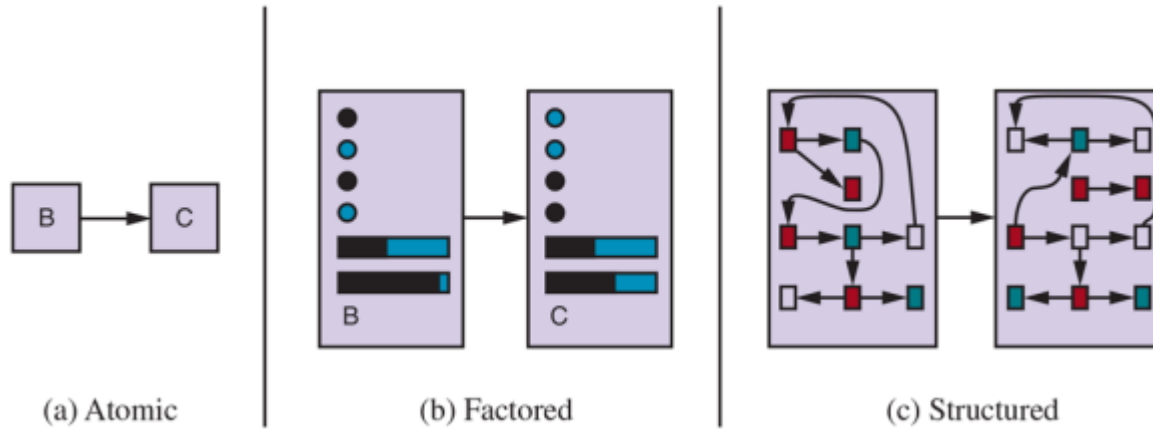
Agentes en la solución de problemas

Cuando la acción correcta a tomar no es inmediatamente obvia, un agente puede necesitar planificar con anticipación: considerar una **secuencia de acciones** que formen un camino hacia un estado objetivo. A tal agente se le llama agente de resolución de problemas, y el proceso computacional que lleva a cabo se llama búsqueda.



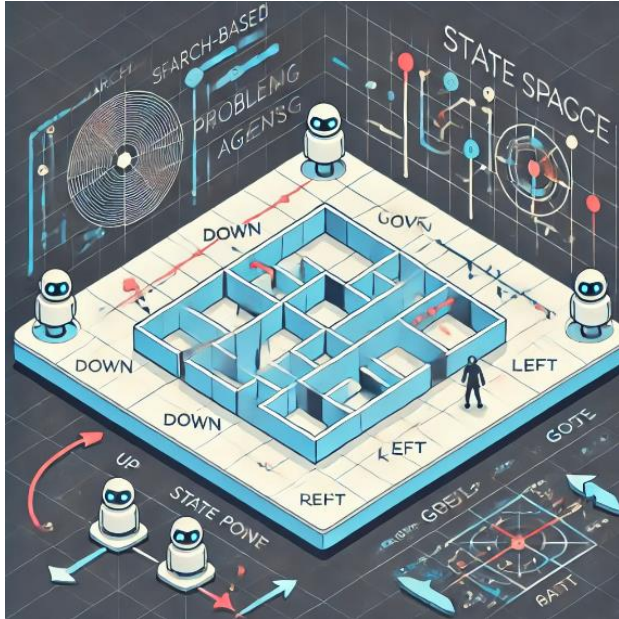
Agentes en la solución de problemas

- Los agentes de resolución de problemas utilizan representaciones atómicas.



Los agentes que usan representaciones factorizadas o estructuradas de estados se llaman agentes de planificación.

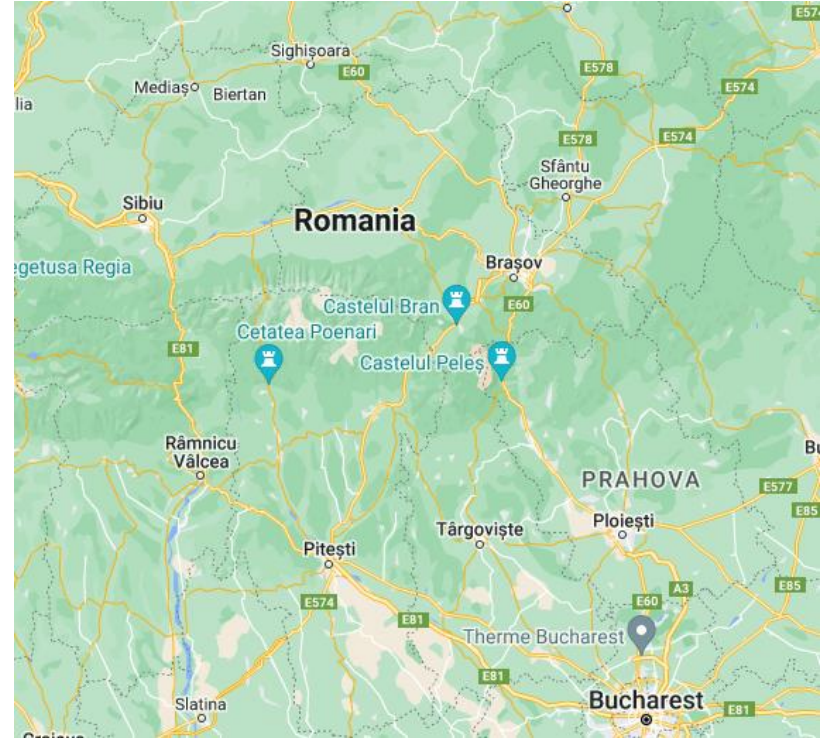
Agentes en la solución de problemas



Para los algoritmos de búsqueda iniciales, consideraremos los entornos simples: episódicos, de un solo agente, completamente observables, deterministas, estáticos, discretos y conocidos.

FORMULACIÓN DE PROBLEMAS

Imagina a un agente ubicado en Rumania. supongamos que el agente está actualmente en la ciudad de Arad y tiene un boleto no reembolsable para volar desde Bucarest al día siguiente. El agente observa las señales de tráfico y ve que hay tres caminos que salen de Arad: uno hacia Sibiu, otro hacia Timisoara y otro hacia Zerind. Ninguno de estos es el destino, por lo que, a menos que el agente esté familiarizado con la geografía de Rumania, no sabrá qué camino seguir.

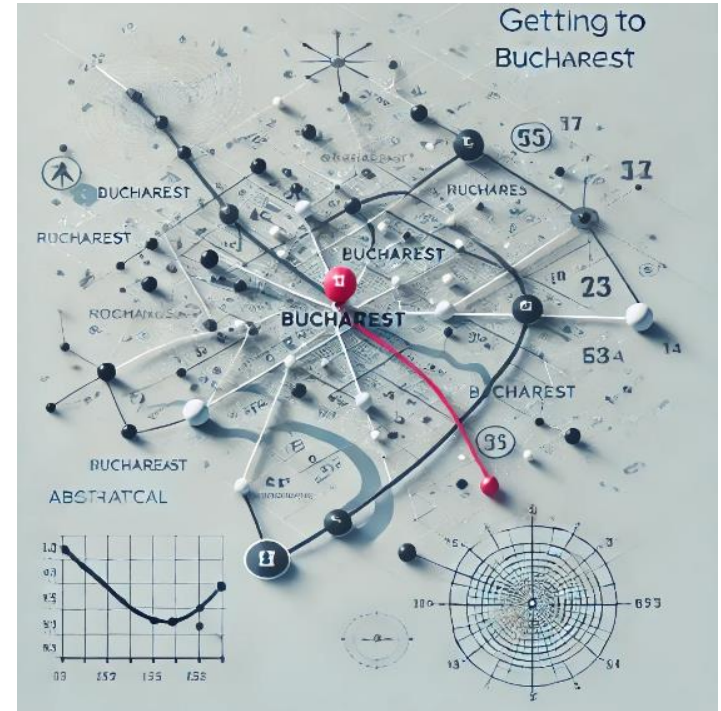


FORMULACIÓN DE PROBLEMAS

DEBEMOS TENER EN CUENTA... El proceso de abstracción

Nuestra formulación del problema de llegar a Bucarest es un modelo, una descripción matemática abstracta, y no la realidad.

El proceso de eliminar detalles de una representación se llama abstracción. Una buena formulación del problema tiene el nivel de detalle adecuado.



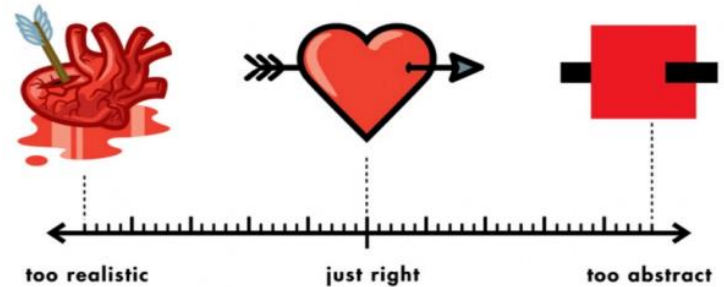
FORMULACIÓN DE PROBLEMAS

ABSTRACCIÓN

Comparar la simple descripción atómica del estado Arad con un viaje real a través del país, donde el estado del mundo incluye muchas cosas: los compañeros de viaje, la distancia a la próxima parada, el estado de la carretera, el clima, el tráfico, etc.

Todas estas consideraciones se dejan fuera de nuestro modelo porque son irrelevantes para el problema de encontrar una ruta a Bucarest

THE ABSTRACT-O-METER



Formulación de problemas

Si el agente no tiene información adicional, es decir, si el entorno es desconocido, entonces el agente no puede hacer nada mejor que ejecutar una de las acciones al azar.

Sin embargo, si proporcionamos información del mundo el agente puede seguir este proceso de resolución de problemas en cuatro fases:

Solución de problemas

PASO 1: Formulación del objetivo

- El agente adopta el objetivo de llegar a Bucarest.

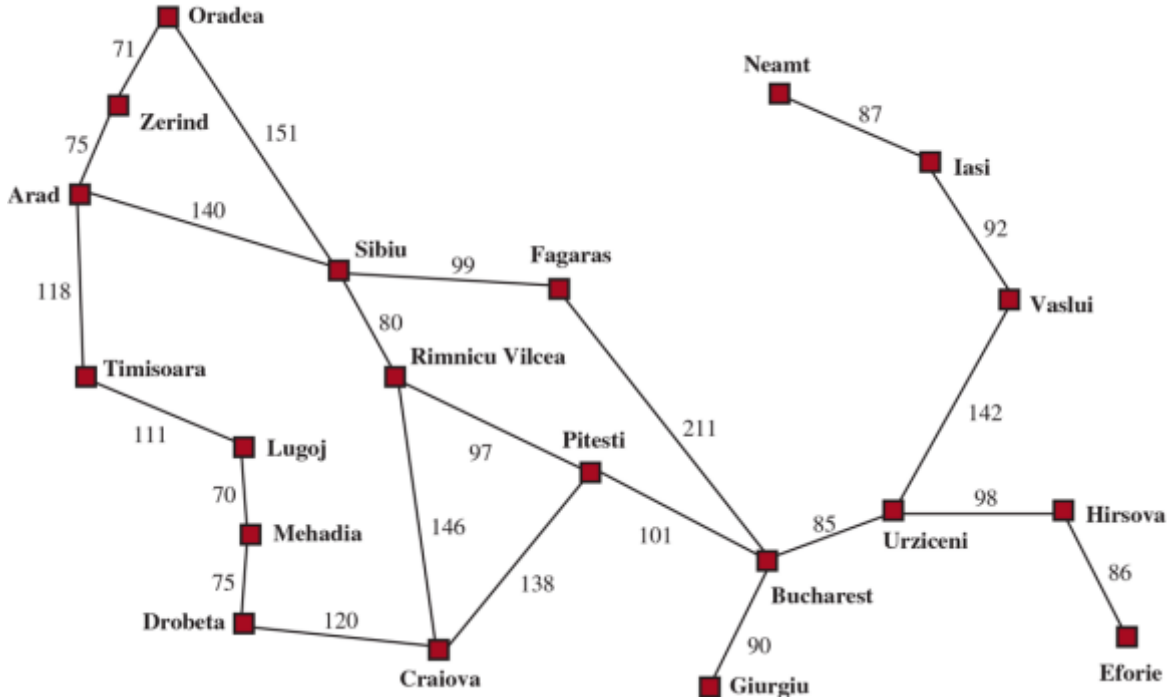
Nota: Los objetivos organizan el comportamiento al limitar los fines y, por ende, las acciones a considerar.



Solución del problema

PASO 2: FORMULACION

El agente elabora una descripción de los estados y acciones necesarios para alcanzar el objetivo, es decir, un modelo abstracto de la parte relevante del mundo.



Solución del problema

PASO 3 BUSQUEDA:

- Antes de tomar cualquier acción en el mundo real, el agente simula secuencias de acciones en su modelo, buscando hasta encontrar una secuencia de acciones que alcance el objetivo. A tal secuencia se le llama solución

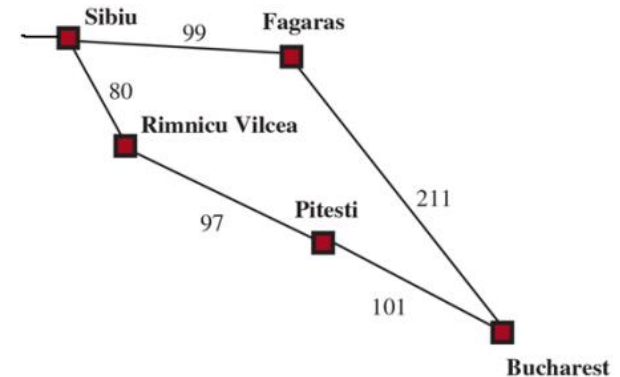
El agente podría tener que simular múltiples secuencias que no alcanzan el objetivo, pero eventualmente encontrará una solución o descubrirá que no es posible encontrar una solución.

Solución del problema

IMPORTANTE!

En un entorno totalmente observable, determinista y conocido, la solución a cualquier problema es una secuencia fija de acciones:

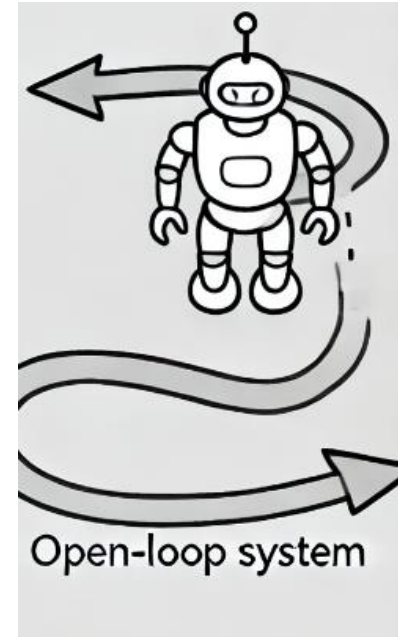
EJM: conducir a Sibiu, luego a Făgăraș, y finalmente a Bucarest.



Solución del problema

PASO 4: EJECUCION

Si el modelo es correcto, una vez que el agente ha encontrado una solución, puede ignorar sus percepciones mientras ejecuta las acciones (COMO SIGUIENDO LA SOLUCIÓN A OJOS CERRADOS) porque la solución está garantizada para llevar al objetivo.



Solución del problema



Otro enfoque es que el sistema constantemente este monitoreando las percepciones. Es preferible cuando existe la posibilidad de que el modelo sea incorrecto o el entorno sea no determinista.

HABLEMOS DE PROBLEMAS!!!

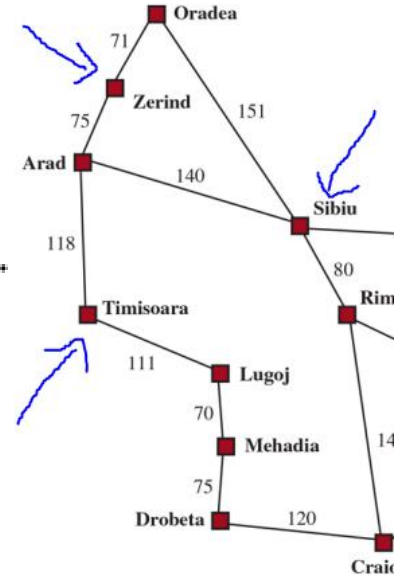
¿Cómo se define un problema de búsqueda?

- Un conjunto de posibles estados en los que el entorno puede estar. A esto lo llamamos el espacio de estados.
- El estado inicial en el que el agente comienza. Por ejemplo: Arad.
- Un conjunto de uno o más estados objetivo. A veces hay un único estado objetivo (por ejemplo, Bucarest), a veces hay un pequeño conjunto de estados objetivo alternativos
- Las acciones disponibles para el agente.

PROBLEMAS DE BUSQUEDA

- Dado un estado (s), **ACCIONES**(s) devuelve un conjunto finito de acciones que se pueden ejecutar. Decimos que cada una de estas acciones es aplicable en ese estado. Un ejemplo:

$$\text{ACTIONS}(\text{Arad}) = \{\text{ToSibiu}, \text{ToTimisoara}, \text{ToZerind}\}.$$



PROBLEMAS DE BUSQUEDA

- Un modelo de transición, que describe lo que hace cada acción. **RESULTADO** (s , a) devuelve el estado que resulta de realizar la acción a en el estado s . Por ejemplo:

$$\text{RESULT}(\textit{Arad}, \textit{ToZerind}) = \textit{Zerind}.$$

PROBLEMAS DE BUSQUEDA

- Una función de costo asociada a la acción, denotada por:

$$\text{ACTION-COST}(s, a, s')$$

Cuando estamos programando o haciendo matemáticas, buscamos el costo numérico de aplicar la acción “a” en el estado s para alcanzar el estado s' .

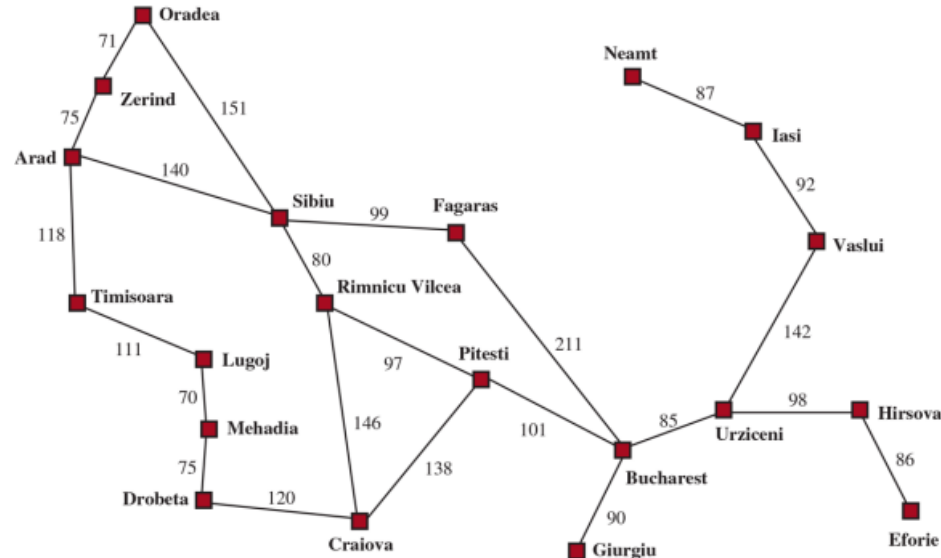
Un agente que resuelve problemas debería usar una función de costo que refleje su propia medida de desempeño; por ejemplo, para agentes de búsqueda de rutas, el costo de una acción podría ser la longitud en millas o el tiempo que toma.

PROBLEMAS DE BUSQUEDA

Una secuencia de acciones forma un camino (**path**), y una solución es un camino desde el estado inicial hasta un estado objetivo. Suponemos que los costos de las acciones son aditivos; es decir, el costo total de un camino es la suma de los costos individuales de las acciones. Una **solución óptima** tiene el costo de camino más bajo entre todas las soluciones.

PROBLEMAS DE BUSQUEDA

El espacio de estados puede representarse como un grafo en el que los vértices son estados y los arcos dirigidos entre ellos son acciones. El mapa de Rumania mostrado en la figura es un grafo de este tipo, donde cada carretera indica dos acciones, una en cada dirección.

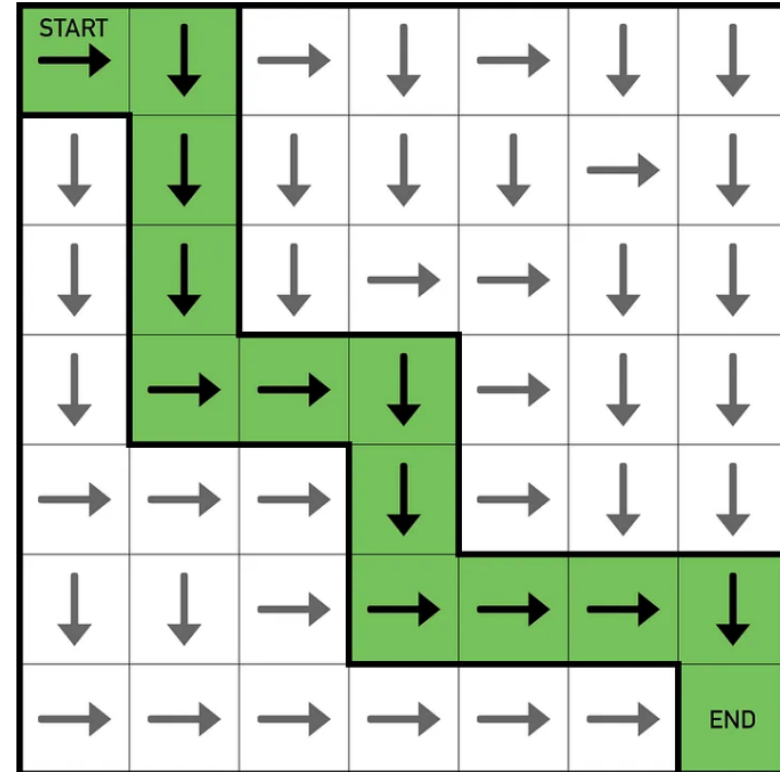




PROBLEMAS ESTANDARIZADOS

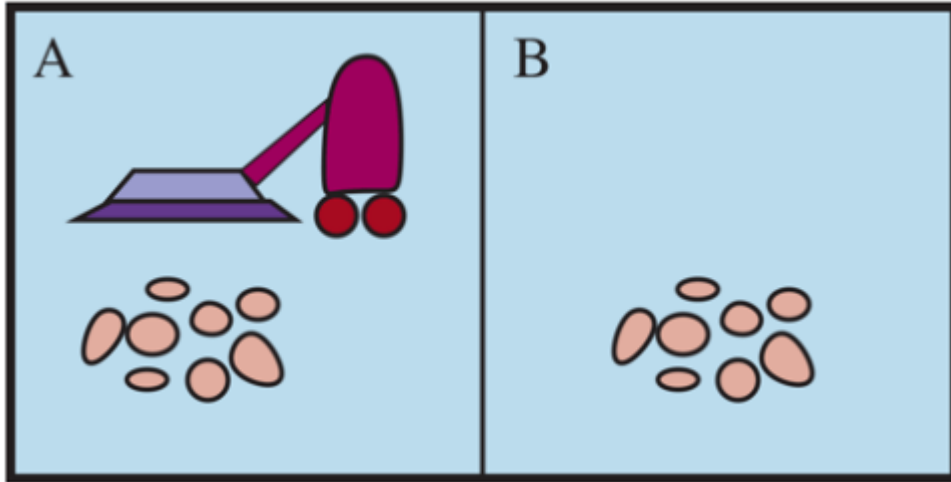
GRID WORLD

Es una matriz rectangular bidimensional de celdas cuadradas en la que los agentes pueden moverse de celda a celda. Normalmente, el agente puede moverse a cualquier celda adyacente libre de obstáculos, ya sea horizontal o verticalmente, y en algunos problemas, diagonalmente. Las celdas pueden contener objetos, que el agente puede recoger, empujar o actuar sobre ellos de alguna otra manera; una pared u otro obstáculo infranqueable en una celda impide que el agente se mueva hacia esa celda.



VEAMOS UN EJEMPLO

El mundo de una aspiradora, que consiste en un agente de limpieza robótico en un mundo compuesto por cuadrados que pueden estar sucios o limpios.

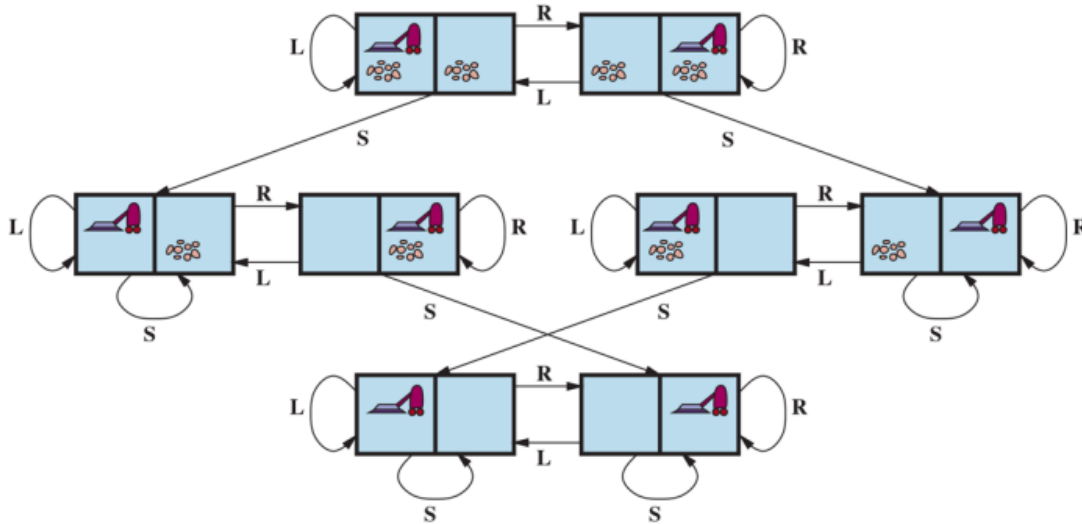


El agente de la aspiradora percibe en qué cuadrado se encuentra y si hay suciedad en el cuadrado. El agente comienza en el cuadrado A.

Las acciones disponibles son moverse a la derecha, moverse a la izquierda, aspirar la suciedad o no hacer nada.

APLIQUEMOS LOS CONCEPTOS

ESTADOS: Un estado del mundo indica qué objetos están en qué celdas. Los objetos son el agente y cualquier suciedad. En la versión simple de dos celdas, el agente puede estar en cualquiera de las dos celdas, y cada celda puede contener suciedad o no , entonces tenemos: $2 \times 2 \times 2 = 8$ **ESTADOS POSIBLES**



The state-space graph for the two-cell vacuum world. There are 8 states and three actions for each state: L = *Left*, R = *Right*, S = *Suck*.

VEAMOS UN EJEMPLO

Percept sequence	Action
<i>[A, Clean]</i>	<i>Right</i>
<i>[A, Dirty]</i>	<i>Suck</i>
<i>[B, Clean]</i>	<i>Left</i>
<i>[B, Dirty]</i>	<i>Suck</i>
<i>[A, Clean], [A, Clean]</i>	<i>Right</i>
<i>[A, Clean], [A, Dirty]</i>	<i>Suck</i>
<i>⋮</i>	<i>⋮</i>
<i>[A, Clean], [A, Clean], [A, Clean]</i>	<i>Right</i>
<i>[A, Clean], [A, Clean], [A, Dirty]</i>	<i>Suck</i>
<i>⋮</i>	<i>⋮</i>

function REFLEX-VACUUM-AGENT(*[location,status]*) **returns** an action

if *status = Dirty* **then return** *Suck*
else if *location = A* **then return** *Right*
else if *location = B* **then return** *Left*

APLIQUEMOS LOS CONCEPTOS

ESTADO INICIAL: Cualquiera de los 8.

ACCIONES: En el mundo de dos celdas definimos tres acciones: Aspirar, mover a la Izquierda y mover a la Derecha. En un mundo bidimensional de múltiples celdas, necesitamos más acciones de movimiento.

MODELO DE TRANSICIÓN: Aspirar elimina cualquier suciedad de la celda del agente; Adelante mueve al agente una celda en la dirección en la que está mirando, a menos que choque con una pared, en cuyo caso la acción no tiene efecto. Atrás mueve al agente en la dirección opuesta,

ESTADOS OBJETIVO: Los estados en los que cada celda está limpia.

COSTO DE ACCIÓN: Cada acción cuesta 1.

ROMPECABEZAS DE SOKOBAN:

Es un problema estandarizado en el que el objetivo del agente es empujar una cantidad de cajas, dispersas por la cuadrícula, a ubicaciones de almacenamiento designadas. Con la condición de que solo puede haber una caja por celda.

7	2	4
5		6
8	3	1

Start State

	1	2
3	4	5
6	7	8

Goal State

Cuando un agente se mueve hacia adelante a una celda que contiene una caja y hay una celda vacía en el otro lado de la caja, tanto la caja como el agente se mueven hacia adelante. El agente no puede empujar una caja hacia otra caja o hacia una pared.

Para un mundo con **n** celdas sin obstáculos y **b** cajas, hay
 $n \times n! / (b!(n-b)!)$ Estados

Ejm: en una cuadrícula de 8×8 con una docena de cajas, hay más de 200 billones de estados.

- **ESTADOS:** Una descripción del estado especifica la ubicación de cada una de las fichas.
- **ESTADO INICIAL:** Cualquier estado puede designarse como el estado inicial.
- **ACCIONES:** Aunque en el mundo físico es una ficha la que se desliza, la forma más sencilla de describir una acción es pensar en el espacio en blanco moviéndose hacia la Izquierda, Derecha, Arriba o Abajo.
Si el espacio en blanco está en un borde o esquina,
no todas las acciones serán aplicables.
- **MODELO DE TRANSICIÓN:** Mapea un estado y una acción a un estado resultante
- **COSTO:** 1



GRACIAS