

Introducción a Búsqueda

Jorge Baier

Departamento de Ciencia de la Computación
Pontificia Universidad Católica de Chile

Santiago, Chile



- Introducir los problemas de búsqueda
- Conocer los elementos que componen un Mundo Determinístico con un agente
- Comprender el algoritmo de búsqueda genérica



- Todo problema en el que es necesario *encontrar* una solución es un problema de búsqueda.
- Un algoritmo se dice *de búsqueda* se mueve a través de un espacio de búsqueda para encontrar una solución.
- Se usa un algoritmo de búsqueda en problemas en donde *no se tiene* una solución algorítmica.
- Posibles ejemplos: planificar un viaje, jugar ajedrez, resolver un puzle.



Diferencias con Programación en Lógica

- Muchos problemas de búsqueda se pueden modelar usando programación en lógica.
- Sin embargo las herramientas que veremos en este capítulo tienen mejor rendimiento en una familia muy amplia de problemas.
- La programación en lógica es una técnica muy buena para resolver problemas altamente combinatoriales, pero su rendimiento es bajo en los (importantes) problemas que veremos en este capítulo



Un solo agente:

- Generación de lenguaje
- Cubo Rubik, Puzle de $(n^2 - 1)$
- Sudoku, Atomix
- Navegación de Robots, Planificación de Movimientos
- Razonamiento Hipotético
- Verificación de Software

Múltiples agentes:

- Damas, Ajedrez, Go, ...
- Bridge, Poker, ...
- Backgammon



Mundos Determinísticos, con Un Agente

- Un espacio de estados \mathcal{S} .
- Un conjunto \mathcal{A} de operadores. Un operador $a \in \mathcal{A}$ es una función *parcial*

$$a : \mathcal{S} \mapsto \mathcal{S}.$$

- Por cada estado, un conjunto $A(s) \subseteq \mathcal{A}$ de *operadores aplicables* en s . Si $a \in A(s)$, entonces $a(s)$ está definida. Definimos

$$Succ(s) = \{a(s) \mid a \in A(s)\}$$

- Una función de costo $c : \mathcal{A} \rightarrow \mathbb{R}^+$.
- Un estado inicial s_{init} .
- Un conjunto de estados finales G .



Solución a un Problema de Búsqueda

- Una secuencia de operadores $o_0 o_1 \dots o_n$ es aplicable en s_0 ssi $s_{i+1} = o_i(s_i)$ está definido, para todo $i \in \{0, \dots, n\}$.
- Una secuencia aplicable de operadores $o_0 o_1 \dots o_n$ es una solución al problema ssi cuando $s_{i+1} = o_i(s_i)$, para todo $i \in \{0, \dots, n\}$, $s_{n+1} \in G$.



Búsqueda Genérica

El siguiente es un algoritmo de búsqueda genérico.

Input: Un problema de búsqueda (S, A, s_{init}, G)

Output: Un nodo objetivo

- 1 *Open* es un contenedor vacío
- 2 *Closed* es un conjunto vacío
- 3 Inserta s_{init} a *Open*
- 4 $parent(s_{init}) = null$
- 5 **while** *Open* $\neq \emptyset$
- 6 $u \leftarrow \text{Extraer}(\textit{Open})$
- 7 Inserta u en *Closed*
- 8 **for each** $v \in Succ(u) \setminus (\textit{Open} \cup \textit{Closed})$
- 9 $parent(v) = u$
- 10 **if** $v \in G$ **return** v
- 11 Inserta v a *Open*



- Introducir los problemas de búsqueda
- Conocer los elementos que componen un Mundo Determinístico con un agente
- Comprender el algoritmo de búsqueda genérica

