Search

if an action fails, perform a new search
the execute these actions
first mentally determine a sequence of actions
but search in the head
not physical search
A state-space problem consists of

- a function that measures the quality of a state
- a set of goal states, or equivalently, a Boolean function, $\text{goal}(s)$, that is true when $s$ is a goal
- an action to a state
- an action function that maps from a state and a set of actions to a subset of states called the start states

---

**Search**

- Uniformed Search
- Heuristic Search
- Directed Graphs
- Problem Basics
- Search Basics

**Analysis**

- Examples
- Generic Algorithm
cookie problem

Cookie monster wants to eat all the cookies.

CM starts in a particular location.

Locations are represented as a grid. A cookie may or may not be at a location.

CM starts at a particular location.

CM knows where the cookies are. Eats cookies when it finds them.

<table>
<thead>
<tr>
<th>O</th>
<th>C</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>L</td>
<td>L</td>
</tr>
</tbody>
</table>
| L | R | R | R
Frontier starts with (A)

(A) is removed
not a goal state
add (A, B) and (A, C) to frontier
remove (A, C) from frontier
does not reach a goal state
add (A, C, F) and (A, C, G) to frontier
Procedure Search(\(A\), \(S\), \(\text{Goal}\))

**Input:** A Graph with nodes \(N\) and arcs \(\mathbb{A}\) with \(\text{Goal}\)

**Output:** Path from \(S\) to \(\text{Goal}\), if found

\(\text{Goal}(): \text{Boolean function of nodes}\)

\(S(): \text{Set of start nodes}\)

\(F:\) a set of paths

while \(F\) is not empty

for each arc from \(S\) to \(N\)

for all states we can reach

\(F\) as a set of paths

\(\{ S \in S \mid (s) \} \rightarrow \overline{F} \)

remove a path \(p\) from \(F\)

if \(\text{Goal}(t)\) then return \(p\)

insert \((s, \ldots, t, n)\) into \(F\)

return null
Generic Search Algorithm

Inputs: Graph \( G \), set of start nodes \( S \), set of goal nodes \( \text{goal} \), arcs \( A \)

Procedure: Search \( G \), \( S \), \( \text{goal} \)

Output: Path from any node in \( S \) to \( \text{goal} \)

produce a path \( p = (s_0, \ldots, t) \) from \( \text{Frontier} \)

while \( \text{Frontier} \) is not empty

for each arc from \( t \) to \( n \)

if \( \text{goal}(t) \) then return \( p \)

insert \( (s, \ldots, t, n) \) into \( \text{Frontier} \)

if \( \text{path} \) has no cycles

return null
Uniform Tree
Heuristics Example
Example 2
8-Puzzle
End Example
Search Examples

Uniformed Example 2