AI Lab - Lesson 2

Informed Search

Davide Corsi
Alessandro Farinelli

University of Verona
Department of Computer Science

October 27^{th} 2021

UNIVERSITÀ
di VERONA
Dipartimento
di INFORMATICA

Start Your Working Environment

Start the previously installed (Session 1) conda environment *ai-lab*

Listing 1: Update Environment

```
cd AI-Lab
git stash (NB: remember to backup the previous lessons before this step!)
git pull
git stash pop
conda activate ai-lab
jupyter notebook
```

Listing 2: Open Lesson

To open the tutorial navigate with your browser to:
lesson_2/lesson_2_problem.ipynb

Uniform-Cost Search Example

At the beginning of *lesson_2/lesson_2_problem.ipynb* you can find an implementation of the last uninformed search algorithm you have seen in class, the Uniform-Cost Search (UCS). The pseudocode is in the next slide.

Uniform-Cost Search (UCS)

```
Input: $problem$
Output: $solution$
$node \leftarrow$ a node with $\textsc{State} = problem.\textsc{Initial-State}, \textsc{Path-Cost} = 0$
$frontier \leftarrow \textsc{Priority-Queue}$ ordered by $\textsc{Path-Cost}$, with $node$ as the only element
$explored \leftarrow \emptyset$
loop
    if $\textsc{Is-Empty}(frontier)$ then return $\textsc{Failure}$
    $node \leftarrow \textsc{Remove}(frontier)$                ▷ Remove node with highest priority
    if $problem.\textsc{Goal-Test}(node.\textsc{State})$ then return $\textsc{Solution}(node)$
    $explored \leftarrow explored \cup node.STATE$
    for each $action$ in $problem.\textsc{Actions}(node.\textsc{State})$ do
        $child \leftarrow \textsc{Child-Node}(problem, node, action)$    ▷ Increase path cost over parent
        if $child.\textsc{State}$ not in $explored$ or $frontier$ then
            $frontier \leftarrow \textsc{Insert}(child, frontier)$
        else
            if $child.\textsc{State}$ is in $frontier$ with higher $path - cost$ then
                replace that $frontier$ $node$ with $child$
```

Note: this is a graph search version

Assignments

- Your assignments for this session are at: *lesson_2/lesson_2_problem.ipynb*. You will be required to implement some informed search algorithms
- The pseudocodes are variations of the Uniform-Cost Search (UCS) where the *priority queue* is ordered by h and $f = g + h$ respectively