SAPIENZA
UNIVERSITÀ DI ROMA

Artificial Intelligence

2024/2025 Prof: Sara Bernardini

Lab 11:
Exam simulation n.2

Francesco Argenziano
email: argenziano@diag.uniroma1.it

*The slides have been prepared using the textbook material available on the web, and the slides of the previous editions of the course by Prof. Luigia Carlucci Aiello, Prof. Daniele Nardi and Dott. Fabio Previtali.

Exercise 1

Two groups of 3 frogs each would like to swap positions. However, the particular arrangement of the lily pads in the river makes the movements of the frogs difficult; therefore, the swap must happen according to some specific rules: 1) a frog can jump only in the direction it is facing; 2) a frog can jump to a lily pad only if the lily pad has no frogs on it; 3) a frog can jump to a free lily pad if it is immediately in front of it; 4) if the lily pad in front of a frog is occupied by another frog but the next one is free, the frog can jump over the other frog and reach the free lily pad. No other movements are permitted. Jumping to a close lily pad has cost 1, while jumping over a frog has cost 2. The initial and final configurations of the problem are depicted in the figure below.

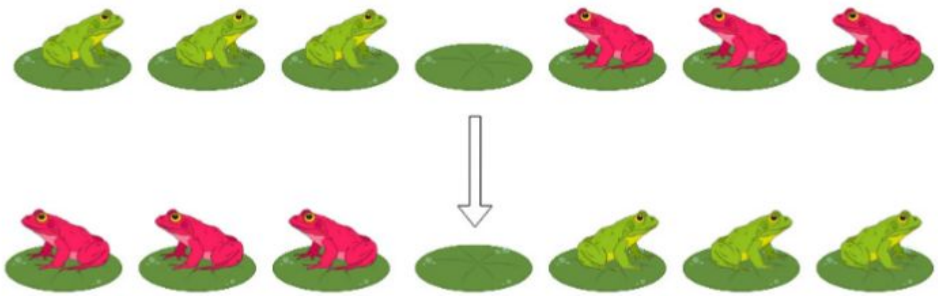

Figure 1: Initial and final configuration of exercise 1.

a) Model the state space of the problem. What is a state in this problem and what are the domains of the variables of the state? What is the initial state? What is the goal state?

b) Model the actions of the problem.

c) Find a heuristic for this problem; comment on whether it is admissible and consistent or not and why.

d) Draw the tree generated by applying A* up to the second level. Give a sequence of actions that solve the problem (not necessarily the one obtained by A*) and show the state after each move and the value of the f function.

Exercise 2

'Oppenheimer' is a new release. 'Barbie' is a new release. Every new release is also a movie. For every new release, there is a movie critic who watched it. A movie critic watched 'Oppenheimer' if and only if (s)he watched 'Barbie'. Mark is a movie critic. Mark watched 'Barbie'.

a) Translate these sentences in FOL. Define a vocabulary for the constants, predicates, and functions that you need. Translate them in CNF (Conjunctive Normal Form).

b) Prove by resolution that "There is a movie critic who watched 'Oppenheimer'".

Given the following two formulas:

$$f(X, g(Y, a), g(b, Z)) \quad and \quad f(h(Z), g(a, a), g(b, b))$$

c) Can they be unified? Show the unification process and, if they can be unified, write the Most General Unifier. Note: capital letters are variables, and lowercase letters are constants.

Let $T = \{P, I, G, A\}$ be an STRIPS planning task describing the job of a modern pigeon, that has to go to the place where the message is and take it. Formally, the task is defined as follows:

- $P : \{at\text{-}pigeon(x), x \in \{L_1, L_2, L_3\}; mes\text{-}found;\ adjacent(x, y)\ with\ \{x, y\}$ $\in \{\{L_1, L_2\}, \{L_2, L_3\}, \{L_2, L_1\}, \{L_3, L_2\}\}$
- $I :\ \{at\text{-}pigeon(L_1)\}$
- $G :\ \{at\text{-}pigeon(L_1), mes\text{-}found\}$
- A:

  fly(x,y):

  - pre: *adjacent*(x,y), *at-pigeon*(x)
  - add: *at-pigeon*(y)
  - del: *at-pigeon*(x)

  take-message:

  - pre: *at-pigeon*(L2)
  - add: *mes-found*
  - del:

1. Compute the value ($h^1(I)$) of the critical path heuristic (with $m = 1$) for the initial state using the dynamic programming algorithm in Ch 9, slide 17.

   Write down the table of intermediate values for each iteration of the algorithm, until convergence. Use the variable names in table 1. See Ch 9, Slide 18 for an example.

I	p(L_1)	p(L_2)	p(L_3)	MF
0	0	∞	∞	∞

Table 1: h^1 Dynamic Programming Table.

$$h^1(s,g) = \begin{cases} 0 & g \subseteq s \\ min_{a \in A, regr(g,a) \neq \perp} c(a) + h^1(s, regr(g,a)) & |g| = 1 \\ \max_{g' \in g} h^1(s, \{g'\}) & |g| > 1 \end{cases}$$

Definition (STRIPS Regression). *Let (P, A, c, I, G) be a STRIPS planning task, $g \subseteq P$, and $a \in A$. We say that g is regressable over a if*

(i) $add_a \cap g \neq \emptyset$; *and*

(ii) $del_a \cap g = \emptyset$.

In that case, the regression of g over a is $regr(g,a) = (g \setminus add_a) \cup pre_a$; else $regr(g,a)$ is undefined, written $regr(g,a) = \perp$.

Dynamic Programming Algorithm

new table $T_0^m(g)$, for all $g \subseteq P$ with $|g| \leq m$

For all $g \subseteq P$ with $|g| \leq m$: $T_0^m(g) := \begin{cases} 0 & g \subseteq s \\ \infty & \text{otherwise} \end{cases}$

fn $Cost_i(g) := \begin{cases} T_i^m(g) & |g| \leq m \\ \max_{g' \subseteq g, |g'|=m} T_i^m(g') & |g| > m \end{cases}$

fn $Next_i(g) := \min[Cost_i(g), \min_{a \in A, regr(g,a) \neq \perp} c(a) + Cost_i(regr(g,a))]$

$i := 0$

do forever:

  new table $T_{i+1}^m(g)$, for all $g \subseteq P$ with $|g| \leq m$

  For all $g \subseteq P$ with $|g| \leq m$: $T_{i+1}^m(g) := Next_i(g)$

  if $T_{i+1}^m = T_i^m$ **then** stop **endif**

  $i := i + 1$

enddo

2. Compute the value of the critical path heuristic (with $m = 2$) for the initial state, $h^2(I)$. Write down the table of intermediate values for each iteration of the algorithm, until convergence. You can use abbreviations for the variable names.

I	p(L_1)	p(L_2)	p(L_3)	MF	L_1,MF	L_2,MF	L_3,MF	L_1, L_2	L_2, L_3	L_1, L_3
0	0	∞	∞	∞	∞	∞	∞	∞	∞	∞

Try also computing the values of h_max and h_add

Definition (h^{add}). *Let $\Pi = (P, A, c, I, G)$ be a STRIPS planning task. The additive heuristic h^{add} for Π is the function $h^{\mathsf{add}}(s) := h^{\mathsf{add}}(s, G)$ where $h^{\mathsf{add}}(s, g)$ is the function that satisfies*

$$h^{\mathsf{add}}(s,g) = \begin{cases} 0 & g \subseteq s \\ min_{a \in A, g' \in add_a} c(a) + h^{\mathsf{add}}(s, pre_a) & g = \{g'\} \\ \sum_{g' \in g} h^{\mathsf{add}}(s, \{g'\}) & |g| > 1 \end{cases}$$

Definition (h^{max}). *Let $\Pi = (P, A, c, I, G)$ be a STRIPS planning task. The max heuristic h^{max} for Π is the function $h^{\mathsf{max}}(s) := h^{\mathsf{max}}(s, G)$ where $h^{\mathsf{max}}(s, g)$ is the function that satisfies*

$$h^{\mathsf{max}}(s,g) = \begin{cases} 0 & g \subseteq s \\ min_{a \in A, g' \in add_a} c(a) + h^{\mathsf{max}}(s, pre_a) & g = \{g'\} \\ \max_{g' \in g} h^{\mathsf{max}}(s, \{g'\}) & |g| > 1 \end{cases}$$

Reminder: → slide 27

... $h^{\max}(s) := h^{\max}(s, G)$ *where* $h^{\max}(s, g)$ ... *satisfies* $h^{\max}(s, g) =$

$$\begin{cases} 0 & g \subseteq s \\ min_{a \in A, g' \in add_a} c(a) + h^{\max}(s, pre_a) & g = \{g'\} \\ \max_{g' \in g} h^{\max}(s, \{g'\}) & |g| > 1 \end{cases}$$

Reminder: → **Chapter 19**

... $h^1(s) := h^1(s, G)$ *where* $h^1(s, g)$ ... *satisfies* $h^1(s, g) =$

$$\begin{cases} 0 & g \subseteq s \\ min_{a \in A, regr(g,a) \neq \perp} c(a) + h^1(s, regr(g, a)) & |g| = 1 \\ \max_{g' \in g} h^1(s, \{g'\}) & |g| > 1 \end{cases}$$