

# Proposta Progetto: Planning & Reasoning

Giacomo Telloli matricola:1792451  
Sapienza Università di Roma

April 1, 2025

## 1 Descrizione del Dominio 1

Il dominio del problema sarà un robot taglia erba che ha l'obiettivo di tagliare tutta l'erba nella mappa, conosce lo stato di carica della sua batteria e conosce dove si trova la stazione di ricarica all'interno della mappa, lo stato della batteria sarà gestito da 3 predicati high, middle e low, si scaricherà di uno stato solo quando il robot deciderà di tagliare l'erba, inoltre deve evitare eventuali ostacoli che sono già segnalati dalla mappa

## 2 Descrizione del Dominio 2

Questo dominio è uguale al primo con la sola differenza che la carica della batteria sarà gestita da un fluente

## 3 Descrizione delle istanze PDDL

Le istanze saranno adattate al dominio cercando di renderle più simili possibili

### 3.1 Istanza 1 (Facile)

Mappa 6x6 con stazione di ricarica al centro e un basso rate di scarica della batteria e un ostacolo

### **3.2 Istanza 2 (Media)**

Mappa 6x6 con stazione di ricarica al punto di partenza e un basso rate di scarica della batteria e 2 ostacoli

### **3.3 Istanza 3 (Difficile)**

Mappa 6x6 con stazione di ricarica al punto di partenza e un basso rate di scarica della batteria durante il movimento ma alto durante il taglio dell'erba e 3 ostacoli

## **4 Planner Usati**

Usare come algoritmi di ricerca A\* e GreedyBFS della libreria fastdownward per il dominio 1 , e ENHSP per il dominio 2

## **5 Euristiche da Usare**

Per ogni algoritmo vorrei usare 3 euristiche hmax,hff, hadd quindi per ogni istanza verrà prodotta una matrice riassuntiva per il confronto degli algoritmi con le relative euristiche applicate

## **6 Reasoning Task with IndiGolog**

Formulare una BAT per risolvere il problema di planning e aggiungere degli eventi esogeni quali pioggia o l'apparizione di un ostacolo improvviso

I tre reasoning task sono :

### **6.1 Legality Task**

Dato uno stato controllare la legalità di una sequenza di azioni

### **6.2 Projection Task**

Controllare se un fluente è vero date un paio di situazioni e una serie di azioni da applicare

### **6.3 Controller Task**

Scrivere una funzione di controllo che reagisce ad eventuali eventi esogeni come la pioggia o l'apparizione improvvisa di un ostacolo