

# Progetto Introduzione Intelligenza Artificiale

## A.A. 2021/2022

**Docente: Prof. Valentina Poggioni**  
**Assistente alla didattica: Dr. Alina Elena Baia**

Il progetto consiste nella realizzazione di una applicazione di Intelligenza Artificiale completa degli aspetti di gestione di: *sensori* per l'acquisizione dei dati dall'esterno relativi a *stati* e *obiettivi*, *ragionamento/ricerca* della soluzione per i goal acquisiti, *esecutori* per la realizzazione delle azioni che conducono alla soluzione.

### Progetto Uniform Coloring

Uniform Coloring è un dominio in cui si hanno a disposizione alcune celle da colorare, e vari colori a disposizione. Per semplicità immaginiamo una griglia rettangolare in cui è possibile spostare una testina colorante fra le celle attigue secondo le 4 direzioni cardinali (N,S,E,W), senza uscire dalla griglia. Tutte le celle hanno un colore di partenza (B=blu, Y=yellow, G=green) ad eccezione di quella in cui si trova la testina indicata con T. La testina può colorare la cella in cui si trova con uno qualsiasi dei colori disponibili a differenti costi ( $\text{cost}(B)=1$ ,  $\text{cost}(Y)=2$ ,  $\text{cost}(G)=3$ ), mentre gli spostamenti hanno tutti costo uniforme pari a 1. L'obiettivo è colorare tutte le celle dello stesso colore (non importa quale) e riportare la testina nella sua posizione di partenza.

La codifica di tutto il dominio (topologia della griglia, definizione delle azioni etc.) è parte dell'esercizio. Partendo dalla posizione iniziale della testina e combinando azioni di spostamento e colorazione, si chiede di trovare la sequenza di azioni dell'agente per raggiungere l'obiettivo.

La posizione iniziale della testina, la struttura della griglia e la colorazione iniziale delle celle sono passati al sistema tramite un'immagine.

Il progetto consiste nel produrre:

1) **Descrizione formale del dominio e dei vincoli** che le azioni eseguibili dall'agente (la testina) nella griglia devono rispettare (esempio vincoli:  $v1$ ="l'agente può compiere un solo passo alla volta",  $v2$ ="l'agente si può muovere solo fra celle adiacenti", etc.). La descrizione sarà un testo che descrive le regole da rispettare e le assunzioni desunte dall'analisi del dominio.

2) **Implementazione delle classi per la ricerca nello spazio degli stati di Smart Vacuum.**

Utilizzando le classi di AIMA-python si implementi quindi un dominio **UniformColoring** come classe derivata da **Problem**, scegliendo e definendo una **rappresentazione per gli stati** e ridefinendo opportuni metodi **actions** e **result** e tutti gli eventuali metodi aggiuntivi, es. **goal\_test**, che si rendessero necessari.

Si descriva e si implementi almeno una **euristica** definendone le caratteristiche di consistenza e ammissibilità rispetto al dominio dato. L'euristica definita mantiene le stesse proprietà nel caso in cui le azioni di spostamento costassero 0 o le colorazioni avessero tutte lo stesso costo?

3) **Acquisizione e classificazione degli input, stato iniziale e goal.** Si realizzi un programma che, passata in input un'immagine contenente la configurazione della griglia e la posizione iniziale dell'agente:

a) interpreti l'immagine individuando la configurazione della griglia e, **attraverso un metodo di classificazione**, la posizione iniziale dell'agente e la colorazione iniziale delle celle. Non ci sono vincoli sul metodo/modello utilizzato per la classificazione. Si consiglia di utilizzare il dataset

eMNIST/MNIST per la classificazione di lettere e cifre scritte a mano, visto a lezione e facilmente reperibile su Web. Assumiamo che le lettere siano solo maiuscole;

b) traduca i dati risultanti dall'analisi delle immagini negli stati **stato\_iniziale** e **stato\_goal** del problema, secondo la rappresentazione definita per il punto 2;

c) **invochi il solutore (vedi punto 2)**, tramite una tecnica di ricerca informata e almeno una non informata, della classe *UniformColoring*, e **produca**, se esiste, la **soluzione del problema**, ovvero la sequenza azioni da eseguire per raggiungere lo stato goal. E' interessante mostrare come algoritmi diversi possano portare a soluzioni diverse nel caso in cui ottimizzino rispetto al costo della soluzione oppure rispetto la sua lunghezza.

4) **Esecuzione.** Si implementi un simulatore dell'esecuzione del piano di azioni calcolato, anche semplicemente attraverso una sequenza di immagini.

**Dovrà inoltre essere prodotta una relazione** che, oltre alla descrizione del dominio richiesta nel punto 1, dovrà contenere statistiche e valutazione dei risultati ottenuti relativi a:

- classificazione lettere e cifre ed estrazione stati dalle immagini
- prestazioni della ricerca nello spazio degli stati, problemi risolti/irrisolti dimensione dei problemi risolti.

Relazione e codice prodotto dovranno essere consegnati attraverso la procedura messa a disposizione nella pagina di Unistudium del corso.

Di seguito sono forniti alcuni esempi di immagini. Griglia e lettere possono essere prodotte digitalmente oppure possono essere disegnate/scritte a mano.

**Al momento della presentazione del progetto verrà richiesta la soluzione di problemi descritti con immagini nuove (mai viste dal sistema).**

**Note:**

- Per l'elaborazione delle immagini in input, e per la generazione della sequenza di immagini che simula il processo di esecuzione si consiglia l'utilizzo della libreria OpenCV.

Per qualunque incertezza sul testo, sullo sviluppo o per la discussione di versioni parziali del progetto scrivere una email a docente e assistenti alla didattica.

ESEMPI di problemi, con possibili soluzioni e relativi costi

**Stato iniziale**

G	T	G	B
G	Y	G	B

**Possibile stato goal**

G	T	G	G
G	G	G	G

**Possibile soluzione**

Sud, col-G, East, East, col-G, Nord, col-G, West, West

Costo = 15    Lunghezza = 9

**Stato iniziale**

G	G
Y	Y
Y	T

**Possibile stato goal**

Y	Y
Y	Y
Y	T

**Possibile soluzione**

Nord, Nord, col-Y, West, col-Y, East, Sud, Sud

Costo = 10    Lunghezza = 8