2025

Indovina Chi?

PROLOG CON PYTHON E MONGODB

ALBERTO ESPOSITO N46006817, CARLO FALCONE N46007079,
GABRIELE TERRACCIANO N46006952

Indice

I nostri personaggi per il gioco.

1. Introduzione e motivazioni

L'elaborato nasce con l'intento di progettare e realizzare un'applicazione intelligente ispirata al gioco "Indovina Chi?", in cui uno dei due giocatori deve dedurre, attraverso domande, quale personaggio sia stato scelto dall'avversario. A partire da questa dinamica, si è scelto di sviluppare un motore deduttivo in linguaggio **Prolog**, in grado di simulare il processo inferenziale basato su attributi caratterizzanti.

Fin dalle fasi iniziali, l'obiettivo non era solo quello di emulare un gioco, ma di integrare concetti centrali dell'intelligenza artificiale simbolica: **ragionamento basato su regole**, **gestione della conoscenza**, **dialogo uomo-macchina** e **persistenza dello stato del mondo**. Per raggiungere questi obiettivi, abbiamo affiancato al motore logico due tecnologie complementari:

- **MongoDB** per la gestione dinamica dei dati.
- **Python** come ponte tra le componenti.

Il progetto nasce dunque con una forte valenza didattica, ma è pensato anche per essere modulare ed estendibile, in modo da consentire futuri sviluppi nel campo dell'analisi dei dati o della costruzione di interfacce utente più avanzate.

2. Cos'è Prolog?

Prolog (PROgramming in LOGic) è un linguaggio di programmazione dichiarativo sviluppato negli anni '70, impiegato principalmente nell'ambito dell'intelligenza artificiale e dei sistemi esperti. A differenza dei linguaggi imperativi, in cui si specificano passo-passo le istruzioni da eseguire, Prolog permette di descrivere la conoscenza e le relazioni logiche tra entità, lasciando al motore di inferenza il compito di determinare i passi esecutivi.

Il paradigma di Prolog si basa sulla **logica del primo ordine (FOL)**: si definiscono dei **fatti**, che rappresentano le conoscenze note, e delle **regole**, che permettono di inferire nuove informazioni. Le query sono formulate come obiettivi logici da soddisfare, e Prolog cerca soluzioni tramite un processo di **unificazione** e **backtracking**.

Tra le caratteristiche principali di Prolog troviamo:

- la capacità di rappresentare in modo naturale la **conoscenza simbolica**;
- l'utilizzo di un **motore inferenziale** che effettua deduzioni a partire da regole e fatti;
- il supporto diretto alla **ricorsione**, molto utile nei processi di ricerca e deduzione.

Grazie a queste peculiarità, Prolog si presta perfettamente alla costruzione di sistemi intelligenti, come giochi logici (come il nostro), configuratori, diagnostici e risponditori automatici.

3. Il motore inferenziale in Prolog

Il cuore logico dell'applicazione è stato implementato in **Prolog**, linguaggio storico dell'intelligenza artificiale simbolica. In questo contesto, Prolog si è rivelato particolarmente adatto grazie al suo modello computazionale basato su fatti e regole.

La conoscenza viene espressa mediante una **base di fatti** del tipo:

- *personaggio(mario, [baffi, occhiali, cappello]).*
- *personaggio(elena, [occhiali, capelli_rossi]).*

Ogni fatto rappresenta un personaggio e la lista degli attributi che lo caratterizzano. Il sistema presenta due modalità di gioco:

- **Modalità 1** (utente indovina): il sistema sceglie casualmente un personaggio e l'utente formula domande o ipotesi. L'interazione è gestita interamente in Prolog e si conclude alla prima ipotesi errata, simulando fedelmente la dinamica del gioco originale.
- **Modalità 2** (Prolog indovina): il giocatore pensa a un personaggio e risponde alle domande poste dal sistema, che filtra i candidati possibili man mano che riceve informazioni.

Nel secondo caso, Prolog applica una semplice ma efficace **regola di inferenza**: sceglie un attributo significativo, lo propone all'utente, e a seconda della risposta aggiorna la lista dei candidati usando include/3 e exclude/3. Il ciclo si ripete fino a che rimane un solo personaggio, o nessuno.

L'interazione avviene attraverso i predicati writeln/1 e read_line_to_string/2, rendendo il sistema più robusto rispetto a input errati.

4. La gestione della conoscenza: KB dinamica

Un punto chiave del progetto è la **separazione tra la logica e la conoscenza**. Le basi di conoscenza (KB) non sono incluse nel file .pl, ma viene generato un altro file dinamicamente a partire da un database non relazione “MongoDB” attraverso una funzione avviata dal main.py. La lista aggiornata dei personaggi e degli attributi viene trasformata in una serie di fatti Prolog, scritti nel file kb_importata.pl, che viene poi caricato nel motore mediante: - *['kb_importata']*.

Questo approccio rende il sistema flessibile: per aggiungere nuovi personaggi o modificare attributi non è necessario alterare il codice Prolog, ma basta modificare i dati nel database (per semplicità e sicurezza si è aggiunta una funzione *popola_personaggi* perché altrimenti il primo avvio avrebbe un database senza personaggi).

5. Perché MongoDB?

La scelta di **MongoDB** come database per la conoscenza è stata guidata da diversi fattori:

- La natura **NoSQL** lo rende perfetto per rappresentare entità con liste variabili di attributi (consapevoli che in ottica di una gestione di una mole elevata di dati sarebbe la scelta opportuna).
- È un sistema **familiare**, già incontrato nel corso di "Advanced Computer Programming", quindi il suo utilizzo è stato per noi immediato e semplice.
- Offre grande flessibilità nell’estensione futura del progetto, permettendo di aggiungere e modificare collections con facilità.

Inoltre, MongoDB è adatto ad essere interrogato e analizzato anche da strumenti esterni, rendendolo ideale per progetti che potrebbero includere **data mining** o **analisi comportamentale** degli utenti.

6. Perché Python?

Python agisce da **componente di coordinamento** tra il motore logico e il database. In particolare, le sue responsabilità sono:

- Connessione e manipolazione del database tramite il pacchetto pymongo
- Generazione del file kb_importata.pl con i fatti Prolog
- Avvio del motore logico con subprocess.run(...)
- Gestione del salvataggio dei risultati delle partite in MongoDB
- Scelta della modalità di gioco (potrebbe essere scelta anche da Prolog)

L’uso di Python permette di mantenere l’interazione utente più moderna, senza rinunciare alla potenza deduttiva di Prolog. La divisione dei compiti tra Python (gestione dati) e Prolog (logica) è ben modulata e conforme a principi di buona progettazione per la divisione dei compiti.

7. Analisi dati e KNIME

Un possibile sviluppo del progetto riguarda l'**analisi delle partite giocate**, già salvate in MongoDB. Questi dati possono essere esportati e analizzati con strumenti di visualizzazione e data mining.

Un esempio è **KNIME**, che permette di:

- caricare collections MongoDB,
- trasformare i dati,
- visualizzare statistiche sulla quantità di domande fatte,
- confrontare le performance tra modalità.

Questo renderebbe l'applicazione non solo un gioco, ma anche un possibile esempio di riconoscimento di un soggetto legato alle proprie caratteristiche.

8. Considerazioni conclusive

Questo progetto ha rappresentato un'occasione concreta per esplorare le basi dell'intelligenza artificiale, mettendo in pratica nozioni di KB, regole deduttive e inferenza. L'integrazione con Python e MongoDB dimostra come un sistema simbolico possa convivere con tecnologie moderne, mantenendo chiarezza e modularità.

Ogni componente è stato scelto e progettato per essere didattico, estensibile e facilmente adattabile. Il risultato è un gioco logico completamente funzionante, che è anche un esercizio completo di progettazione di un sistema intelligente.

Nota sulla documentazione del codice

Il codice sorgente Prolog e Python è stato incluso nel progetto in forma di **allegato separato** per facilitare la lettura della tesina. Abbiamo scelto di non frammentare il testo con listing troppo lunghi, preferendo invece una spiegazione discorsiva accompagnata da estratti rappresentativi dei blocchi logici fondamentali.

In allegato sono inclusi:

- main.py
- interfaccia_mongo.py
- indovina_chi.pl
- kb_importata.pl (generato automaticamente)