

# Ontologia per il Wumpus World in Prolog

## Descrizione del Problema

Il Wumpus World è un problema classico in Intelligenza Artificiale che rappresenta un ambiente di esplorazione dove un agente deve:

- Raccogliere l'oro evitando pericoli come il Wumpus (un mostro) e i pozzi.
- Utilizzare percezioni come:
  - **Stench (puzza)**: indica la presenza del Wumpus in una stanza adiacente.
  - **Breeze (corrente d'aria)**: indica la presenza di un pozzo in una stanza adiacente.
- Inferire informazioni sul mondo basandosi sulle percezioni e sulle regole dell'ambiente.

## Struttura del Dominio

Il Wumpus World è rappresentato da:

- **Entità principali:**
  - **Stanza**: rappresenta una cella della griglia.
  - **Elementi**: Wumpus, oro, pozzi, agente.
  - **Sensazioni**: stench (puzza), breeze (corrente d'aria).
- **Relazioni:**

- `adiacente(Stanza1, Stanza2)`: due stanze sono adiacenti.
- `contiene(Stanza, Elemento)`: una stanza contiene un certo elemento.
- `percepito(Stanza, Sensazione)`: una sensazione è percepita in una stanza.

## Rappresentazione in Prolog

### Adiacenza delle Stanze

Le stanze sono disposte su una griglia 2x2:

```
% Adiacenze delle stanze in una griglia 2x2
adiacente((1,1), (1,2)).
adiacente((1,1), (2,1)).
adiacente((1,2), (1,1)).
adiacente((1,2), (2,2)).
adiacente((2,1), (1,1)).
adiacente((2,1), (2,2)).
adiacente((2,2), (1,2)).
adiacente((2,2), (2,1)).
```

### Contenuto delle Stanze

Definiamo il contenuto delle stanze:

```
% Contenuto delle stanze
contiene((1,1), agente).      % L'agente inizia nella stanza
    (1,1)
contiene((2,2), wumpus).      % Il Wumpus nella stanza
    (2,2)
contiene((1,2), pozzo).       % Un pozzo nella stanza
    (1,2)
contiene((2,1), oro).         % L'oro nella stanza (2,1)
```

### Sensazioni Percepite

Le sensazioni nelle stanze dipendono dal contenuto delle stanze adiacenti:

```
% Sensazioni percepite nelle stanze
percepito(Stanza, stench) :-
    contiene(WumpusStanza, wumpus),
    adiacente(Stanza, WumpusStanza).

percepito(Stanza, breeze) :-
    contiene(PozzoStanza, pozzo),
    adiacente(Stanza, PozzoStanza).
```

## Regole per Inferire Stanze Sicure

Una stanza è considerata sicura se non contiene né un Wumpus né un pozzo:

```
sicura(Stanza) :-
    \+ contiene(Stanza, wumpus),
    \+ contiene(Stanza, pozzo).
```

## Esempi di Inferenza

Ecco alcune query che possiamo eseguire con il sistema logico:

### Quali sensazioni sono percepite in una stanza?

```
?- percepito((1,1), Sensazione).
% Risultato:
% Sensazione = stench ;
% Sensazione = breeze.
```

### Quali stanze sono sicure per l'agente?

```
?- sicura((1,1)).
% Risultato: true.

?- sicura((2,2)).
% Risultato: false.
```

## Dove si trova l'oro?

```
?- contiene(Stanza, oro).  
% Risultato:  
% Stanza = (2,1).
```

## Conclusione

Questa rappresentazione del Wumpus World in Prolog permette di descrivere formalmente il dominio, inferire informazioni basate sulle regole definite e supportare un agente nella navigazione dell'ambiente. Il modello è espandibile per includere ulteriori regole o stanze.