

# Ontologia per il Problema della Attesa al Ristorante

## Descrizione del Problema

Un cliente deve decidere se aspettare o meno in un ristorante basandosi su una serie di attributi, tra cui:

- **Affollamento:** il ristorante è pieno o vuoto?
- **Alternativa:** esistono ristoranti vicini?
- **Cibo:** la qualità del cibo è buona o mediocre?
- **Tempo d'attesa:** il tempo stimato è corto o lungo?

L'obiettivo è rappresentare questa situazione tramite un'ontologia, che include:

- Classi principali: Cliente, Ristorante, Attributi.
- Relazioni: `frequenta`, `haCaratteristica`, `decide`.
- Regole di inferenza per determinare se vale la pena aspettare.

## Rappresentazione in Prolog

### Definizione delle Classi

```

% Classi principali
cliente(mario).
cliente(lucia).

ristorante(ristorante1).
ristorante(ristorante2).

% Attributi del ristorante
affollamento(pieno).
affollamento(vuoto).

alternativa(presente).
alternativa(assente).

cibo(buono).
cibo(mediocre).

tempo_attesa(corto).
tempo_attesa(lungo).

```

## Proprietà

```

% Relazioni tra cliente e ristorante
frequenta(mario, ristorante1).
frequenta(lucia, ristorante2).

% Caratteristiche dei ristoranti
haCaratteristica(ristorante1, affollamento(pieno)).
haCaratteristica(ristorante1, alternativa(assente)).
haCaratteristica(ristorante1, cibo(buono)).
haCaratteristica(ristorante1, tempo_attesa(lungo)).

haCaratteristica(ristorante2, affollamento(vuoto)).
haCaratteristica(ristorante2, alternativa(presente)).
haCaratteristica(ristorante2, cibo(mediocre)).
haCaratteristica(ristorante2, tempo_attesa(corto)).

```

## Regole di Inferenza

```

% Regola per decidere se aspettare
decide(Cliente, aspettare) :-

```

```
frequenta(Cliente, Ristorante),  
haCaratteristica(Ristorante, cibo(buono)),  
haCaratteristica(Ristorante, affollamento(vuoto)).  
  
decide(Cliente, non_aspettare) :-  
    frequenta(Cliente, Ristorante),  
    (haCaratteristica(Ristorante, cibo(mediocre));  
     haCaratteristica(Ristorante, tempo_attesa(lungo))),  
    haCaratteristica(Ristorante, affollamento(pieno)).
```

## Esempi di Inferenza

**Mario frequenta il ristorante1. Deve aspettare?**

```
?- decide(mario, Decisione).  
% Risultato:  
% Decisione = non_aspettare.
```

**Lucia frequenta il ristorante2. Deve aspettare?**

```
?- decide(lucia, Decisione).  
% Risultato:  
% Decisione = aspettare.
```

## Conclusione

Questa ontologia modella il problema della attesa al ristorante e fornisce un sistema di regole per decidere se aspettare o meno, basandosi sugli attributi del ristorante. Il modello è estensibile per aggiungere ulteriori attributi o regole.