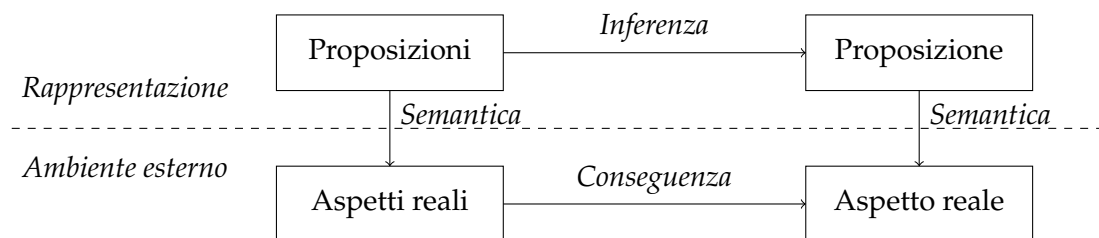


1 Lezione del 30-09-24

1.1 Agenti logici

Vediamo adesso l'implementazione di **agenti logici**, cioè agenti che si basano su una certa **rappresentazione**, detta **base di conoscenza** (KB, *Knowledge Base*) per immagazzinare **proposizioni** su ciò che hanno imparato riguardo all'ambiente esterno, e fare **inferenze**, sulla base di queste proposizioni, rispetto a informazioni non conosciute. Le proposizioni sono legate ad aspetti reali dell'ambiente esterno mediante una determinata **semantica**.

Possiamo schematizzare il funzionamento della KB, e la sua corrispondenza con l'ambiente esterno, come segue:



La conoscenza che l'agente ottiene può essere fornita manualmente, cioè in forma di **assiomi** o conoscenza di *background* (magari in un agente potrebbe essere "precaricata" della conoscenza riguardo allo stato iniziale del problema), estratta dai dati dei sensori, o ricavata dall'esperienza. Un agente logico dovrebbe essere in grado di fare inferenze in quanto spesso le informazioni che ha sull'ambiente esterno è **parziale** o **incompleta**.

Una KB deve poi rappresentare questa conoscenza attraverso un linguaggio, che dovrebbe essere sufficientemente **espressivo** da poter rappresentare la realtà dell'ambiente esterno, ma non troppo complesso da impedire di effettuare inferenze in modo efficiente.

Esistono due approcci all'implementazione di una KB:

- **Dichiarativo:** si concentra su una rappresentazione a **livello di conoscenza** dei fatti, cioè informazioni riguardo a *cosa* è vero. In sistemi di questo tipo, si usano primitive di scrittura (TELL) e query (ASK) sulla KB, partendo da un insieme di informazioni nullo o comunque limitato, fino ad arrivare ad avere una serie di conoscenze comprensive dell'ambiente esterno. I dettagli delle operazioni che poi l'agente andrà a svolgere, il cosiddetto **livello di implementazione**, sono mantenuti separati dalla KB.
- **Procedurale:** si concentra su una rappresentazione di *come* effettuare operazioni. Invece di implementare sistemi che possano immagazzinare proposizioni sull'ambiente esterno, si va a codificare l'informazione direttamente nel codice, attraverso procedure, algoritmi, o come avevamo visto nei modelli a riflesso, regole "if-then".

1.1.1 Basi di conoscenza

Come abbiamo detto, una base di conoscenza è formata da una serie di formule (formule *atomiche*, cioè proposizioni) contenenti informazioni riguardo all'ambiente esterno e codificate in un certo **linguaggio formale**. Si possono definire alcune primitive per l'interazione con la KB:

- **TELL:** aggiungi una nuova proposizione alla KB;

- **ASK**: richiedi informazioni dalla KB;
- **RETRACT**: elimina informazioni dalla KB.

La KB si basa sui fatti che già conosce per ricavare inferenze o **deduzioni logiche** α , della forma:

$$KB \models \alpha$$

L'agente logico si interfaccia con la KB attraverso le primitive sopra definite, e implementa effettivamente un ciclo TELL-ASK-TELL del tipo:

Algoritmo 1 Agente logico

Input: le percezioni correnti
Output: la prossima azione da eseguire
TELL(percezioni correnti)
prossima azione \leftarrow ASK(KB)
TELL(prossima azione)
return prossima azione

Ovvero, l'agente invia le sue percezioni correnti alla KB, e richiede la prossima azione da eseguire. Invia poi l'azione scelta alla KB (così che possa diventare parte delle informazioni note), e la restituisce.

1.1.2 Differenza fra KB e DB

Una KB potrebbe sembrare simile ad un comune database: la differenza è che il database si occupa solo di ricavare fatti specifici, senza possibilità di deduzione di alcun tipo. La KB è invece progettata per mantenere una rappresentazione strutturata dei fatti, specifici o generali, come riferiti a oggetti reali, e permettere quindi inferenze su quei fatti.