Introduzione all'intelligenza artificiale

spitfire

A.A. 2024-2025

Contents

1 Introduzione

Che cos'è l'intelligenza artificiale? Prima di tutto, dovremmo definire che cos'è **l'intelligenza**. Negli anni sono state date molte definizioni:

- "L'intelligenza è una capacità mentale molto **generale** che, tra le altre cose, coinvolge la capacità di **ragionare, pianificare, risolvere problemi, pensare in maniera astratta, comprende idee complesse, apprendere velocemente e imparare dall'esperienza**". da "Mainstream science on intelligence: An editorial with 52 signatories, history, and bibliography, Intelligence 24(1):13–23, 1997"
- "L'intelligenza è la **capacità di adattarsi efficacemente all'ambiente**, o cambiando se stessi o cambiando l'ambiente oppure trovandone uno nuovo... l'intelligenza non è un singolo processo mentale, ma piuttosto una combinazioni molti processi mentali indirizzati verso un adattamento efficace all'ambiente" da Encyclopedia Britannica, 2006.
- ...

Quindi il problema di fondo è quello di **definire l'intelligenza**. Le relazioni tra l'informatica e le scienze cognitive non sono quindi sporadici e sono particolarmente significativi. Uno schema proposto da **Russel e Norvig e il seguente**:

essi propongono uno schema che "astrae" i tipi di intelligenza, ponendoli su due dimensioni:

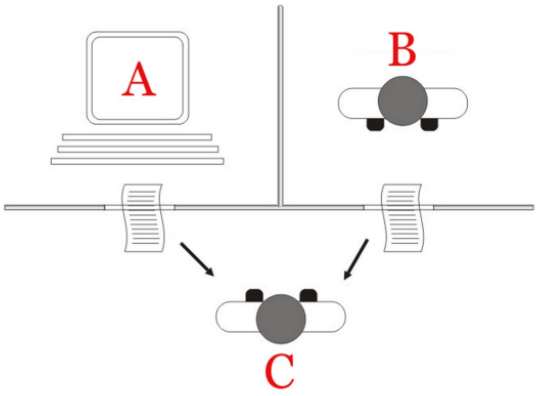
- Sull'asse delle ordinate troviamo il contrasto tra **l'imitare l'essere umano** (cioè imitare il suo modo di agire) e il **pensare come un umano** (quindi il "risolvere problemi")
- Sull'asse delle ascisse troviamo il **contrasto tra il pensare e l'agire**

1.1 Storia dell'intelligenza artificiale

Il termine "intelligenza artificiale" fu coniato nell'agosto del 1955 da **John McCarty, Marvin Minsky, Allan Newell e Herbert Simon**, i quali proposero al Dartmouth College di Hanover (New Hampshire) di organizzare il "Dartmouth College Summer Research Project on Artificial Intelligence"; cioè uno spazio dove accogliere persone che volessero discutere del tema dell'intelligenza artificiale. Essi descrissero l'iniziativa nel seguente modo:

> "Lo studio dovrà procedere sulla base della congettura che ogni aspetto dell'apprendimento o ogni altra caratteristica dell'intelligenza può essere sia, in linea di principio, descrivibile in maniera talmente precisa che una macchina può essere costruita per simularla"

In realtà, la storia dell'intelligenza artificiale risale persino ad Alan Turing (1912-1954), il quale definì il cosiddetto **Test di Turing**: l'idea di questo test è che ci sia un essere umano C separato fisicamente da due interlocutori, uno anch'esso umano, chiamato B, e l'altro una macchina, chiamata A, programmata in qualche modo. C non può interloquire direttamente con A e B, tuttavia può scambiare messaggi con essi tramite un qualche sistema di comunicazione (foglietti di carta, una chat ecc...). Quando C, interagendo in maniera dialogica con i due interlocutori, non riesce più di una certa percentuale di volte a capire chi dei due è la macchina allora A mostra un comportamento intelligente (cioè "agisce come un essere umano", "agisce razionalmente").



Tuttavia, nel 1966, fu prodotto un programma chiamato **ELIZA**, il quale imitava uno **psicoterapeuta**, che **superò il test di Turing nonostante fosse basato su regole di pattern matching con espressioni regolari**. Il programma portava l'utente ad avere una **conversazione plausibile**, cioè davano all'utente l'illusione di star parlando con un **essere umano**. Per questo, il test di Turing ha solo un **interesse storico** e risulta poco interessante. Non è quindi un caso che, nel gruppo di studio citato sopra, una delle cose di cui ci si è occupati non fosse dedicata subito all'apprendimento ma al **ragionamento**.

1.2 Approcci simbolici

La disciplina che si occupa delle forme corrette di ragionamento è la **logica**. La logica è quindi lo studio sistematico delle **forme di inferenza valida**, cioè forme di elaborazione e rappresentazione della conoscenza che garantiscono che da informazioni vere

si derivino informazioni vere. La **logica computazionale** è l'uso della logica per effettuare ragionamenti riguardo alla computazione (es. dimostrazione della correttezza di un programma). Alcuni sforzi iniziali nel settore dell'intelligenza artificiale erano legati alla realizzazione di **dimostratori automatici di teoremi**.
Un'**inferenza** è un ragionamento che stabilisce delle relazioni tra **premesse** e delle **conclusioni**. I **modelli computazionali** di inferenza si interessano quindi di:

- Quali informazioni possono trarre dato un insieme di premesse?
- Perché la conclusione è corretta?

L'inferenza riguarda quindi il trarre delle conclusioni quando **le premesse sono vere**. Dire che un'inferenza è corretta, tuttavia, **non dice nulla sul valore di verità delle conclusioni**, quindi un ragionamento può essere corretto anche se **le premesse sono false**. Le varie forme di inferenza sono:

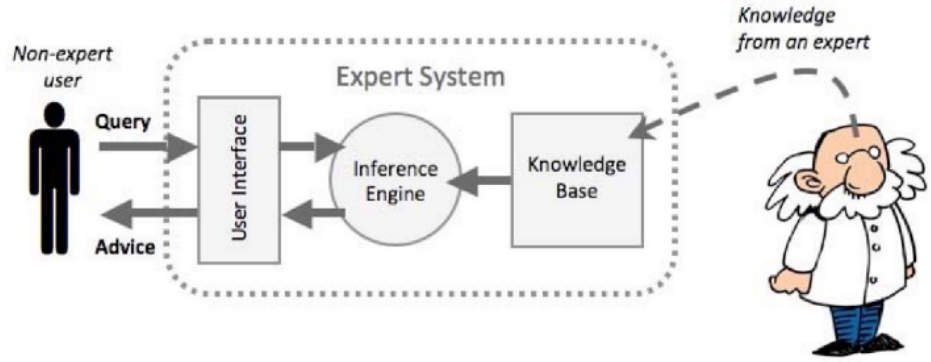
- **Deduzione**:"Se le premesse sono vere, allora le conseguenze devono essere anch'esse vere". Questa forma di ragionamento parte da delle premesse **generali** e trae delle conclusioni **particolari**
- **Inferenze di "senso comune"**: Esse non sono sempre "valide", ma sono **utili nella pratica**; sono modelli per spiegare **quando un inferenza è giustificata** e calcolarla di conseguenza. Esempi di questo tipo di inferenze sono:
  - **Induzione**: Questa forma di ragionamento parte da delle **osservazioni particolari** e arriva a delle **conclusioni generali**.
  - **Abduzione**: Sillogismo in cui la premessa maggiore è certa e la premessa minore è probabile, per cui anche la conclusione risulta solo probabile.

Dove si applica quindi l'IA di tipo simbolico?

- Problemi che possono essere espressi in termini di **vincoli** che devono essere soddisfatti
- Situazioni in cui si devono studiare delle **sequenze di azioni** per portare un certo stato dell'ambiente circostante ad un determinato obbiettivo

In passato, si svilupparono applicazioni il cui obbiettivo era **risolvere problemi specifici e delimitati**; questa applicazioni presero il nome di **sistemi esperti**. Degli esempi notevoli sono:

- **Dendral**(Anni '60): Automatizzò il processo di decisione e l'approccio alla risoluzione dei problemi dei chimici organici
- **Mycin**(Anni '70): Supportò l'identificazione dei batteri che causavano infezioni gravi e la raccomandazione di antibiotici



Tuttavia, presto divennero evidenti i **limiti** dei sistemi esperti:

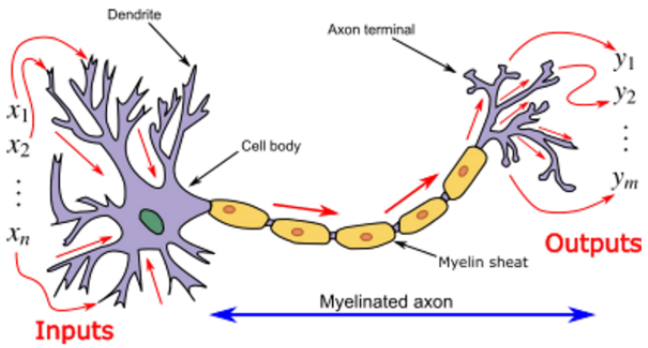
- È necessario trovare un **esperto disposto a codificare la sua conoscenza all'interno della base di conoscenza**
- I sistemi esperti **non scalano bene con grandi quantità di informazioni**: l'aggiornamento del sistema esperto rispetto ai **nuovi approcci alla risoluzione del problema per cui è stato costruito** è particolarmente complesso, soprattutto se il problema **non è ben delimitato**. La base di conoscenza del sistema rischierebbe quindi di **contenere troppi assiomi e regole**.
- Il costo di sviluppo di questi sistemi è **elevato**

1.3 Approcci sub-simbolici

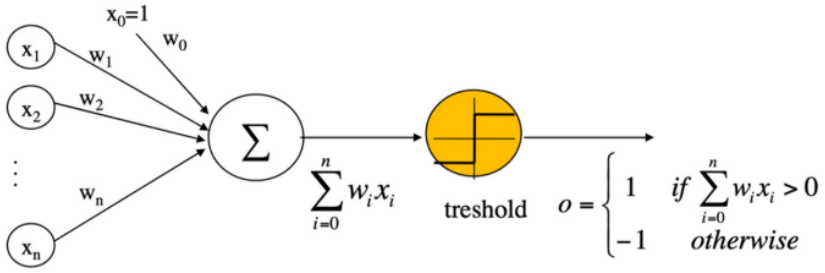
I sistemi di IA **sub-simbolici non manipolano una rappresentazione simbolica** per trovare soluzioni a problemi, ma effettuano **calcoli secondo alcuni principi** che hanno dimostrato di essere in grado di portare alla risoluzione del problema. Esempi notevoli sono:

- Algoritmi genetici
- Reti Neurali
- "Intelligenza dello sciame" (Swarm Intelligence)

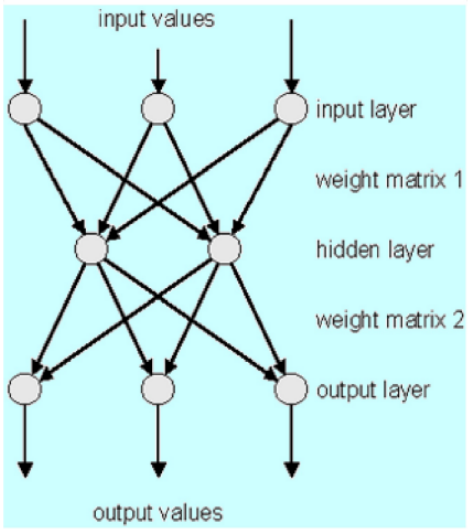
Tuttavia, l'argomento più importante correntemente è quello del **Machine Learning**. Le **reti neurali artificiali** (Artificial Neural Networks (ANN)) sono una simulazione astratta del nostro sistema nervoso, il quale contiene una collezione di **neuroni** che comunicano tra loro tramite delle connessioni dette **assoni**.



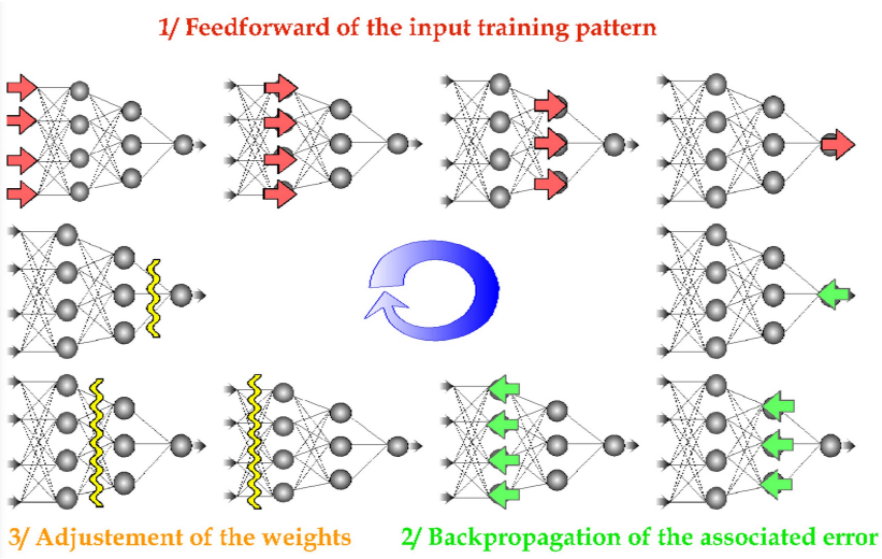
Il modello delle ANN ha delle certe somiglianze con gli assoni e i dendriti nel nostro sistema nervoso. Il primo modello di rete neurale artificiale fu proposto nel 1943 da **McCulloch** e **Pits** nei termini di un **modello computazione dell'attività neurale**. Questo modello fu poi seguito da altri modelli proposti da **John Von Neumann, Marvin Minsky, Frank Rosenblatt** e molti altri. Rosenblatt definì un modello "algebrico" del neurone, chiamato **percettrone**; esso è una pietra miliare della ricerca sulle reti neurali. Il percettrone cerca di **simulare le operazioni svolte da un singolo neurone**; l'apprendimento quindi diventa un problema di **scegliere i pesi e le soglie corrette**.



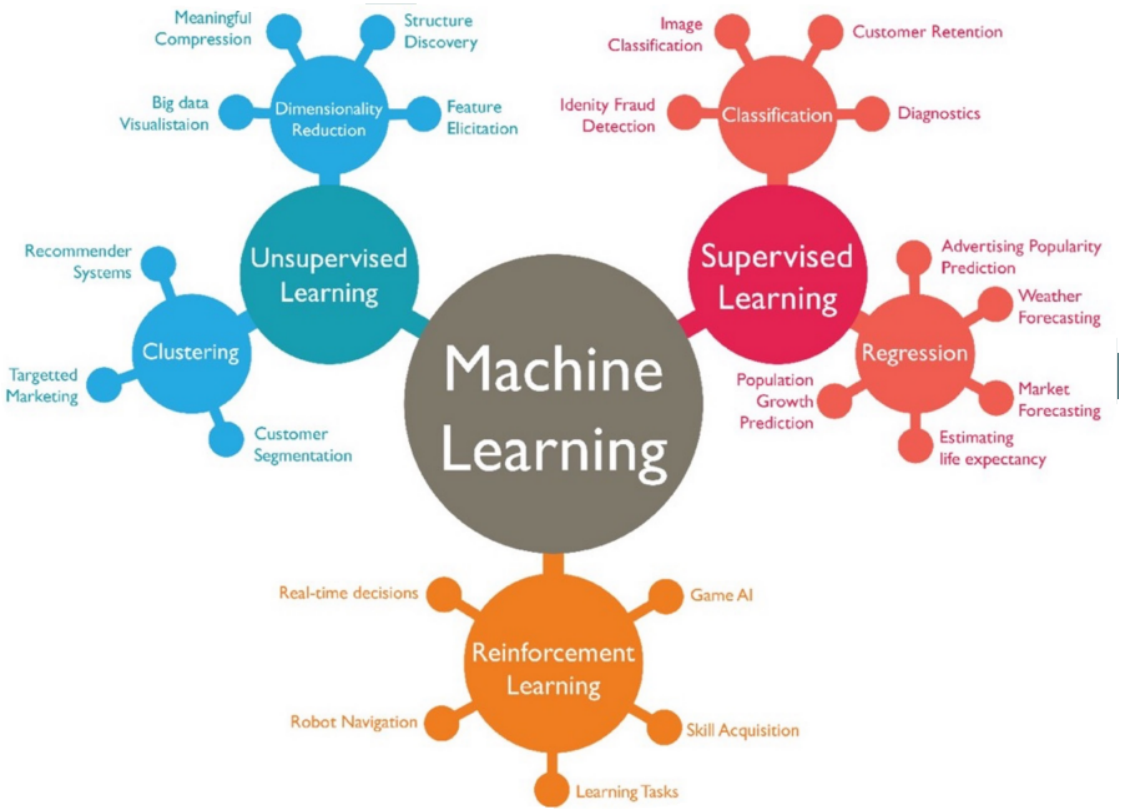
tendenzialmente, si arriva ad avere dei **percettroni multistrato**, dove ogni nodo è un singolo percettrone (introdotti per la prima volta da Minksy e S.Papert nel 1969)



come faccio però a determinare i pesi di tutta la rete? Si utilizza il meccanismo della **back-propagation**: l'idea è che l'input è una **rappresentazione del problema** e che vi sia un **output desiderato** che la rete deve offrire; inizialmente la rete neurale ha **dei pesi causali**, che verranno corretti **retropropagando** l'output desiderato sulla rete. Questo è un approccio all'apprendimento che si dice "**supervisionato**".



Oltre all'apprendimento supervisionato, esistono molte altre tecniche di addestramento:

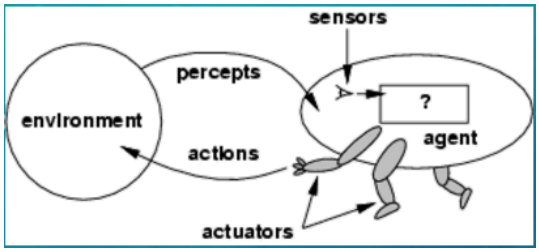


1.4 Agenti intelligenti, architetture e ambienti

Un **agente** è tutto ciò che può essere vista come "percepente il suo ambiente" attraverso dei **sensori** e che può **agire su tale ambiente** attraverso degli **attuatori**. Come agenti possono essere quindi classificati

- Gli **esseri umani**, definendo come "sensori" gli occhi, le orecchie ecc... e come attuatori la bocca, le gambe, le braccia ecc...
- Gli **agenti robotici**, i quali hanno telecamere e sensori ad infrarossi come sensori e vari motori come attuatori
- Nulla vieta che **un agente possa essere anche solamente un software**

Un agente è quindi rappresentabile nel seguente modo:



la forma più generale di agente è una **funzione** che mappa **l'insieme potenza di tutte le percezioni istantanee** $\mathcal{P}^*$ ad una azione dell'insieme **di tutte le azioni eseguibili dall'agente** $\mathcal{A}$, cioè:

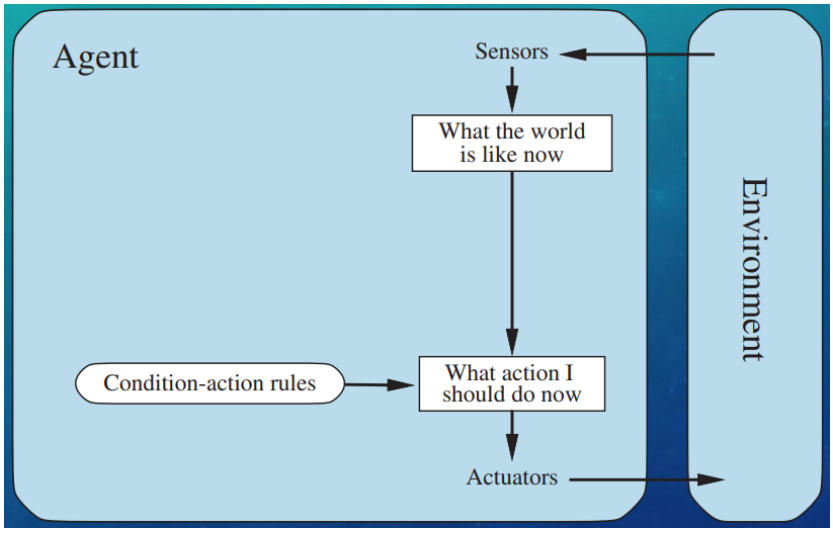
$$f : \mathcal{P}^* \to \mathcal{A}$$

quindi, l'agente può mappare una **sequenza arbitrariamente lunga di percezioni istantanee** (insieme potenza di $\mathcal{P}$) **ad una singola azione contenuta nell'insieme** $\mathcal{A}$. Un agente è quindi **la sua architettura (la sua struttura profonda) più il suo programma (specifico dell'agente)**. Gli agenti possono essere classificati, in base alla loro architettura interna, nelle seguenti classi:

- Agenti con riflessi semplici
- Agenti con riflessi basati su un modello
- Agenti basati su un obbiettivo e su un modello
- Agenti basati su un'utilità e un modello.

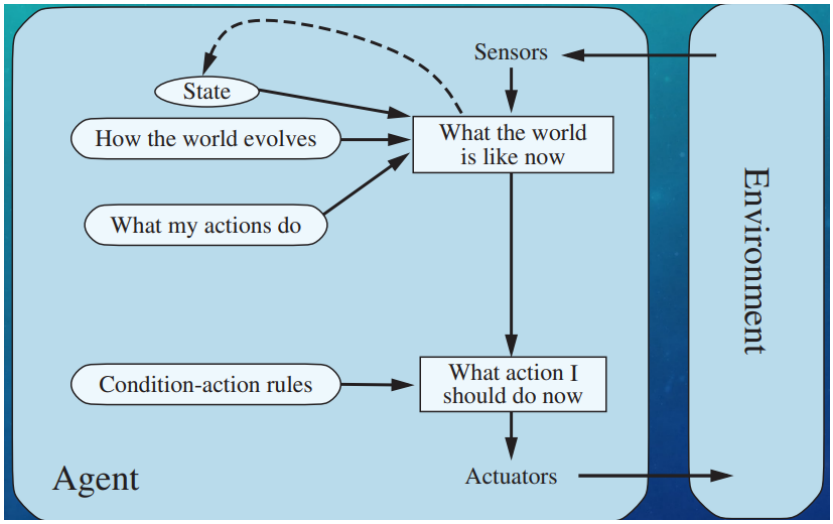
1.4.1 Agenti con riflessi semplici

Un agente con riflessi semplici è un agente in cui vi è una comunicazione con l'ambiente (tramite i sensori e gli attuatori dell'agente). Al suo interno, i sensori vanno a realizzare una "vista" che rappresenta **lo stato attuale dell'ambiente circostante**. L'agente deve quindi **scegliere quale azione intraprendere** e, in questo caso, per farlo ha solamente a disposizione delle regole del tipo **condizione-azione**.



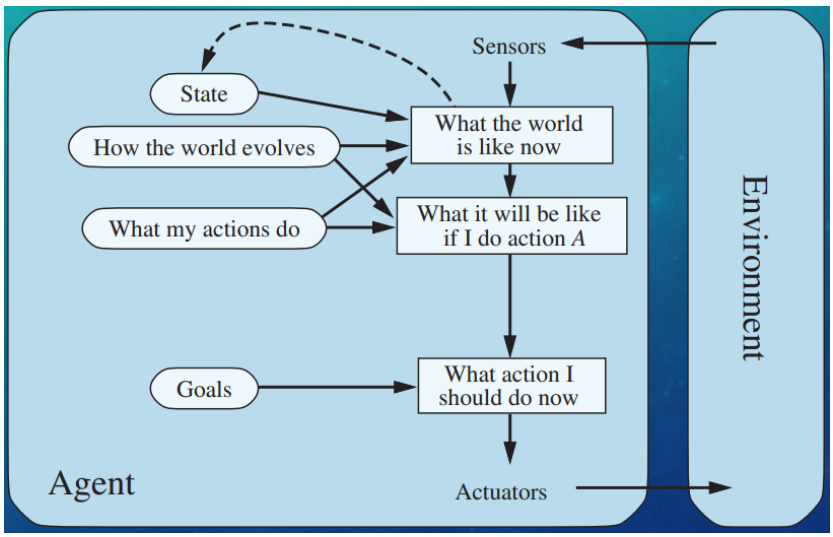
questo tipo di agente è **privo di stato**. Con architetture estremamente semplici, magari con più agenti, si riescono quindi ad ottenere dei comportamenti, se non intelligenti, perlomeno utili.

1.4.2 Agenti con riflessi basati su modello



La differenza più significativa di questi agenti rispetto agli agenti con riflessi semplici è la **conoscenza del proprio stato interno**. Questo tipo di agente può quindi **riflettere sul proprio stato** e quindi effettuare azioni basate su di esso. Anche ignorando che l'ambiente ha una struttura, questo tipo di agente deve quindi avere un **modello matematico dell'evoluzione del mondo rispetto alle azioni fatte** che gli permetta di decidere quale azioni intraprendere. Lo stato **non è una memoria che mantiene lo stato del mondo**, ma è solo un'informazione sullo stato dell'agente. L'agente, inoltre, deve **poter percepire il proprio stato come parte dell'ambiente** (o come una "percezione esterna" o proprio come uno stato interno all'agente e che esso aggiorna e conosce). Questo tipo di agente quindi può eseguire **comportamenti più adattivi rispetto all'ambiente**.

1.4.3 Agenti basati su un obbiettivo e su un modello



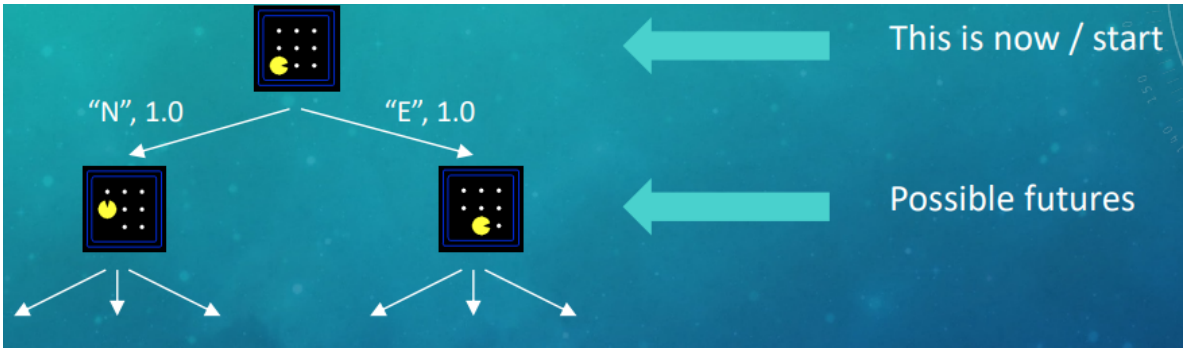
Questo tipo di agente, tramite i sensori ed eventualmente l'informazione di stato, si fa un'idea dello stato in cui si trova. Questo agente presenta una **funzione che, partendo dallo stato dell'agente, ritorna tutte le azioni ammissibili che l'agente può intraprendere**. Lo stato attuale viene quindi messo come **radice di un albero** e generiamo, a partire da esso, un numero di figli pari al numero di azioni ammissibili. Possiamo costruire questo albero perché sappiamo **in quale stato si andrà a finire eseguendo una determinata azione**. Per ogni nodo di stato che viene generato in questo modo, potremmo generare **un ulteriore livello di approfondimento dell'albero**, cioè ogni nodo stato potrebbe diventare la radice di un ulteriore sotto-albero. In linea di principio, potrei quindi costruire l'albero di **tutti gli stati raggiungibili possibili tramite ogni combinazione possibile di azioni**. Ovviamente, il fattore di ramificazione di questo albero è **estremamente elevato**. L'agente deve quindi esplorare questo albero per capire se esistono delle configurazioni nelle quali **l'obbiettivo sia verificato**. Vi è quindi una **costruzione dello spazio degli stati** e una **ricerca nello spazio degli stati**.
Per capire quali azioni intraprende, l'agente deve quindi risolvere un **problema di ricerca**, il quale consiste in:

- Uno **spazio degli stati**
- Una **funzione di successione**, che dato un certo stato e l'azione che si vuole intraprendere (la quale potrebbe avere un certo **costo**), in quale stato si vada a finire. Questa funzione mi dice anche **quali sono le azioni ammissibili in un determinato stato**
- Uno **stato iniziale**
- Una **funzione "goal test"** che ci dica **se l'obbiettivo è stato raggiunto o meno**

Una **soluzione** ad un problema di ricerca è una **sequenza di azioni** (un piano) che trasformano lo stato iniziale nello stato obbiettivo. Uno **spazio di ricerca** astrae l'ambiente per selezionare solo le informazioni utili per risolvere il problema.

lo spazio degli stati, per problemi di ragionevole complessità, **tendono ad esplodere**, quindi non si possono applicare generalmente algoritmi **forza bruta** per trovare una configurazione che risolva il problema. La costruzione dello spazio di ricerca avviene tramite **un albero di ricerca**, dove:

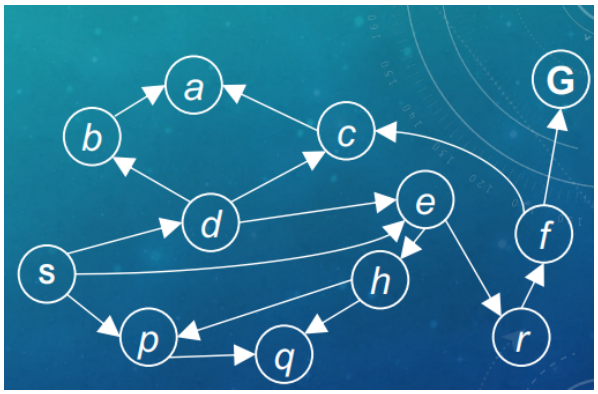
- Lo stato iniziale è il nodo radice
- I figli corrispondono agli stati successivi data un'azione
- I nodi mostrano gli stati, ma **corrispondono ai piani per raggiungere quelli stati**
- Per la maggior parte dei problemi, non arriviamo mai a costruire l'intero albero (troppo grande)



Invece di usare un albero per rappresentare il problema di ricerca, potremmo invece usare un **grafo dello spazio degli stati**: esso da una rappresentazione matematica del problema di ricerca nel seguente modo:

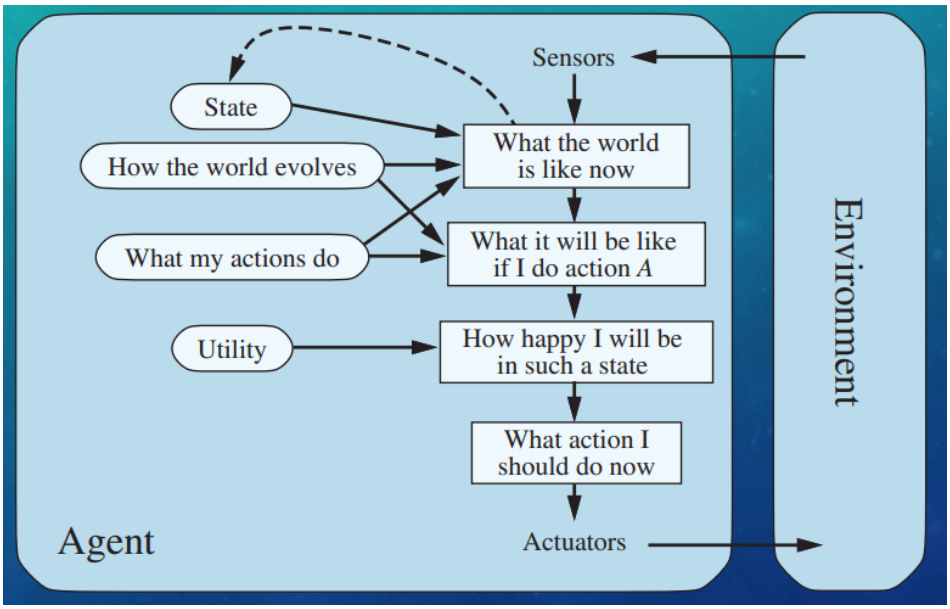
- I nodi sono **le possibili configurazioni del mondo**
- Gli archi rappresentano **i risultati delle azioni**

- La funzione **"goal test"** viene rappresentata da un **insieme di nodi**



In un grafo dello spazio degli stati, **ogni stato occorre solamente una volta!** Tuttavia, raramente possiamo costruire l'intero grafo in memoria, poiché **esso cresce troppo velocemente**; tuttavia è un'idea utile. Qual'è quindi la differenza sostanziale tra un grafo dello spazio degli stati e un albero di ricerca? Ogni **nodo** in un albero di ricerca è l'equivalente di un **intero cammino sul grafo dello spazio degli stati**. Entrambi devono essere costruiti **"on demand"** (cioè li espandiamo solamente quando occorre) e devono esser espansi il meno possibile.

1.4.4 Agenti basati su un'utilità e un modello



La differenza di questo tipo di agente con il precedente è la **scomparsa della funzione "goal test"** e l'introduzione di una **funzione "Utility"**, la quale è una vera e propria funzione che attribuisce ad un certo stato del mondo un idea di **quanto quello stato sia desiderabile dall'agente**. Perché? Ci sono diversi motivi:

- Potrei non essere in grado di definire formalmente una funzione che descriva l'obbiettivo

- Dato lo stato dell'ambiente, posso definire una funzione che valuti i vari elementi di esso e mi permetta di effettuare valutazione sulla prossima aziona da eseguire (es. scacchi)

- Avere una funzione d'utilità mi permette di considerare obbiettivi contrastanti fra loro; quindi poter definire una funzione che valuti tutti i fattori in gioco e che possa portare alla **discriminazione di certi stati** (es. problema di logistica)

1.4.5 Caratteristiche dell'ambiente

Russel e Norvig definiscono una serie di caratteristiche per l'ambiente:

- **Ambienti accessibili vs Inaccessibili**: Un **ambiente accessibile** è un ambiente dove l'agente può ottenere un informazione **completa, accurata e aggiornata** riguardo lo stato dell'ambiente. Molti ambiente moderatamente complessi (includendo, per esempio, il mondo fisico e internet) sono **inaccessibili**. Più un ambiente è accessibile, più è semplice costruire un agente che possa operare in esso. I problemi di accessibilità del mondo si presentano ogni qualvolta che ci accingiamo a risolvere problemi nel **mondo fisico**

- **Ambienti deterministici vs non-deterministici**: Un **ambiente deterministico** è un ambiente in cui ogni azione ha un singolo effetto garantito: non c'è **incertezza** riguardo allo stato in cui l'ambiente si troverà in seguito al compiersi di un azione. Il mondo fisico dovrebbe essere sempre considerato come sostanzialmente **non-deterministico**, salvo certi casi in cui si può pensare come deterministico. Gli ambienti non-deterministici presentano problemi maggiore per il progettista dell'agente.

- **Ambienti episodici vs sequenziali**: In un ambiente **episodico**, l'esperienza dell'agente può essere divisa in **passi atomici** dove l'agente percepisce uno stimolo ed effettua una singola azione. La scelta dell'azione da intraprendere **dipende solamente dall'episodio stesso**. Gli ambienti episodici sono più semplici dal punto di vista del progettista dell'agente, poiché l'agente può decidere quale azione intraprendere basandosi solo sull'episodio corrente, **non deve quindi ragionare riguardo all'interazione tra questo episodio e quelli futuri**.

- **Ambienti statici vs dinamici**: Per **ambiente statico** si intende un ambiente che **non cambia nel mentre che l'agente sta decidendo l'azione da compiere**; un cambiamento in un ambiente statico avviene quindi **solamente a causa di un'azione da parte dell'agente**. Per **ambiente dinamico** si intende un ambiente che **cambia mentre l'agente sta decidendo quale azione intraprendere** e che quindi ha al suo interno altri processi, oltre all'agente, che ne modificano lo stato e le cui azioni possono interferire con le azioni dell'agente (come nella teoria dei sistemi concorrenti); le trasformazioni di un ambiente dinamico quindi avvengono **al di fuori del controllo dell'agente**. Il mondo fisico è un ambiente altamente dinamico.

- **Ambienti discreti vs continui**: Un **ambiente discreto** è un ambiente che presenta un numero **fissato e finito** di azioni e di percezioni in esso. Gli **ambienti**

continui hanno un certo livello di **discrepanza** con i sistemi computerizzati. Gli ambienti discreti potrebbero essere gestiti, in linea di principio, da una specie di **tabella di ricerca** (lookup table). Per semplificare la gestione di un ambiente continuo, si può **sovrapporre ad esso una sua discretizzazione** in modo da rendere più semplice la progettazione e l'implementazione degli agenti che devono operare al suo interno.

Environm.	Accessib.	Determ.	Episodic	Static	Discrete
Chess	Yes	Yes	No	Yes	Yes
Backgammon	Yes	No	No	Yes	Yes
Poker	No	No	No	Yes	Yes
Medical Diagnosis	No	No	No	No	No
Taxi driver	No	No	No	No	No