1 INTRODUZIONE

1.1 Definizione di Intelligenza Artificiale

L'intelligenza artificiale (AI = "*Artificial Intelligence*") è una disciplina ingegneristica il cui obiettivo è comprendere e costruire entità intelligenti. Non esiste una definizione assoluta di AI, ma è possibile dare, generalmente, 4 definizioni:

2)Sistemi che pensano come umani	3)Sistemi che pensano razionalmente
1)Sistemi che agiscono come umani	4)Sistemi che agiscono razionalmente

La prima riga rivolge l'attenzione al *ragionamento*, mentre la seconda al *comportamento*. La prima colonna invece misura il successo in base alla *somiglianza all'essere umano*, mentre la seconda ha come metro di paragone il concetto di *razionalità*. Un sistema è "razionale" se, date le sue conoscenze, "fa la cosa giusta".

1.2 Il test di Turing

Il test di Turing, proposto da Alan Turing nel 1950, è un test che è stato concepito per fornire una soddisfacente definizione operativa dell'intelligenza. Esso prevede un insieme di domande poste da un esaminatore umano a un computer, e se le risposte ottenute dal sistema sono indistinguibili da quelle umane, allora il test è considerato superato e il computer può essere ritenuto **intelligente**. Chiaramente, il computer, per poter superare il test, dovrebbe possedere le seguenti capacità:

- **interpretazione del linguaggio umano**, per poter comunicare con l'esaminatore;
- **rappresentazione della conoscenza**, per poter memorizzare quello che sa o sente;
- **ragionamento automatico**, per utilizzare la conoscenza memorizzata in modo da rispondere alle domande e trarre nuove conclusioni;
- **apprendimento**, per adattarsi a nuove circostanze, individuare ed estrapolare pattern (un pattern è un modo per risolvere un problema specifico).

Da notare che il test di Turing evita deliberatamente l'interazione diretta tra l'esaminatore e il computer, dato che la simulazione *fisica* di una persona non è richiesta per la determinazione di intelligenza.

1.3 Pensare come essere umani: approccio della modellazione cognitiva

Se vogliamo dire che un determinato programma ragiona come un essere umano, dobbiamo prima determinare come pensiamo, entrando *dentro* i meccanismi interni del cervello umano. Ci sono due modi per farlo:

- **l'introspezione (bottom-up)**, che è l'identificazione diretta dei dati neurologici, ovvero il tentativo di catturare "al volo" i nostri pensieri mentre scorrono;
- **la sperimentazione psicologica (top-down)**, che va proprio il comportamento umano;

Una volta che abbiamo formulato una teoria della mente sufficientemente precisa, diventa possibile esprimerla sottoforma di un programma per computer. Andando poi a confrontare la sequenza dei

passi del ragionamento della macchina con un'analoga sequenza prodotta da soggetti umani, si può ottenere una prova che alcuni dei meccanismi del programma operano anche negli esseri umani. Questo campo è quello delle cosiddette **scienze cognitive**; esso unisce i modelli per computer sviluppati dall'intelligenza artificiale e le tecniche di sperimentazione psicologica, nel tentativo di costruire teorie precise e verificabili sul funzionamento della mente umana.

1.4 Pensare razionalmente: approccio delle leggi del pensiero

Per costruire sistemi intelligenti, si parte da programmi che possono in linea di principio, risolvere qualsiasi problema descritto in linguaggio logico e dotato di una soluzione. Tali programmi sono fondati sulle *leggi del pensiero* formulate sulla base dei **sillogismi aristotelici**, che forniscono dei pattern di deduzione che portano sempre a conclusioni corrette quando sono corrette le premesse: ad esempio "Socrate è un uomo; tutti gli uomini sono mortali; quindi Socrate è mortale".
Quest'approccio ha due punti deboli:

1. Non è facile esprimere una conoscenza non formalizzata nei termini strettamente formali richiesti dalla notazione logica, in particolare quando tale conoscenza non è sicura al 100%.
2. C'è una grande differenza tra essere in grado di risolvere un problema "in linea di principio" e farlo nella pratica.

1.5 Agire razionalmente: l'approccio degli agenti razionali

Cosa contraddistingue un semplice *agente*, da un *agente razionale*?
Un **agente** è semplicemente qualcosa che agisce, che fa qualcosa. Un **agente razionale** agisce in modo da ottenere il miglior risultato o, in condizioni di incertezza, il miglior risultato atteso. Essere in grado di formulare deduzioni corrette è talvolta parte di un agente razionale, perché un modo di agire razionalmente è ragionare in termini logici, arrivare alla conclusione che una data azione porterà al soddisfacimento dei propri obiettivi, e agire quindi in tal senso.

2 AGENTI INTELLIGENTI

Nel capitolo 1 abbiamo introdotto il concetto di **agente razionale**. In questo capitolo lo definiremo più concretamente per arrivare al concetto di **agente intelligente**. Cominceremo con l'esaminare gli agenti, gli ambienti e le relazioni tra essi.

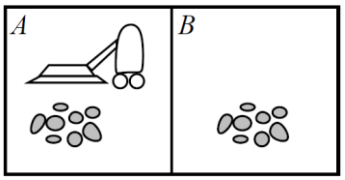
2.1 Agenti e ambienti

Un agente è qualsiasi cosa possa essere vista come un sistema che percepisce il suo ambiente attraverso dei **sensori** e agisce su di esso mediante degli **attuatori**. Useremo il termine percezione per indicare gli input percettivi dell'agente in un dato istante. La sequenza percettiva è la storia completa di tutto ciò che l'agente ha percepito nella sua esistenza. In generale, *la scelta dell'azione di un agente in un qualsiasi istante può dipendere dall'intera sequenza percettiva osservata fino a quel momento*. Se possiamo specificare l'azione prescelta dall'agente per ogni possibile sequenza percettiva, allora abbiamo descritto l'agente in modo completo. In termini matematici diciamo quindi che il comportamento di un agente è descritto dalla **funzione agente**, che descrive la corrispondenza tra una qualsiasi sequenza percettiva e una specifica azione:

$$f: P \Longrightarrow A$$

È importante notare che la funzione di un agente artificiale sarà implementata da un programma agente. In particolare, mentre una funzione è una descrizione matematica astratta; il programma è una sua implementazione completa, in esecuzione sull'architettura agente.

Esempio dell'aspirapolvere



Consideriamo il semplice mondo dell'aspirapolvere. Ci sono solo due posizioni: i riquadri A e B. L'agente aspirapolvere percepisce in quale riquadro si trova e se c'è dello sporco in tale locazione. Può scegliere di muoversi a sinistra, a destra, aspirare lo sporco o non fare nulla. Una funzione agente è molto semplice: "*se il riquadro corrente è sporco, aspira, altrimenti muoviti nell'altro riquadro*". La figura seguente ne mostra una tabulazione parziale:

Sequenza percettiva	Azione
[*A, Pulito*]	*Destra*
[*A, Sporco*]	*Aspira*
[*B, Pulito*]	*Sinistra*
[*B, Sporco*]	*Aspira*
[*A, Pulito*], [*A, Pulito*]	*Destra*
[*A, Pulito*], [*A, Sporco*]	*Aspira*
⋮	⋮
[*A, Pulito*], [*A, Pulito*], [*A, Pulito*]	*Destra*
[*A, Pulito*], [*A, Pulito*], [*A, Sporco*]	*Aspira*
⋮	⋮

vediamo che si possono definire vari agenti del mondo dell'aspirapolvere semplicemente riempiendo la colonna di destra in modi diversi. La domanda che sorge, naturalmente, allora, è questa: *qual è il modo più corretto di progettare la tabella*? In altre parole, che cosa rende un agente migliore di un altro?

2.2 Comportarsi correttamente: il concetto di razionalità

Un **agente razionale** è un agente che fa la *cosa giusta*: "fare la cosa giusta" è quel comportamento che fa sì che l'agente ottenga il massimo grado di successo. Di conseguenza, avremo bisogno di qualche tecnica per misurare il successo: quest'ultima, insieme alle descrizioni dell'ambiente, dei sensori e degli attuatori dell'agente, fornirà la specifica completa dell'attività che l'agente è chiamato a svolgere.

Misure di prestazione

Una **misura di prestazione** (*performance measure*) rappresenta il criterio in base al quale valutare il successo del comportamento di un agente. Possiamo ricorrere tipicamente a una misura di prestazione:

- **soggettiva**, quindi chiedere all'agente stesso quanto sia contento della propria prestazione; ma alcuni agenti non sarebbero in grado di risponde e altri potrebbero illudersi;
- **oggettiva**, tipicamente stabilita dal progettista dell'agente stesso; noi ricorreremo tipicamente a quest'ultimo tipo di misura di prestazione.

Esempio dell'aspirapolvere (parte II)

Consideriamo l'esempio dell'agente aspirapolvere, visto nel paragrafo precedente.
Una misura di prestazione adeguata farebbe riferimento al grado di pulizia del pavimento, dopo che l'agente ha "pulito" per un certo lasso di tempo T: ad esempio si potrebbe assegnare *un punto* per ogni riquadro di pavimento pulito nel tempo T; e magari *una penalità* per ogni riquadro che invece è rimasto sporco.

Come regola generale, è meglio progettare le misure di prestazione in base all'effetto che si desidera ottenere sull'ambiente piuttosto che su come si pensa che debba comportarsi l'agente.

Razionalità
Per ogni possibile sequenza di percezioni, un agente razionale dovrebbe scegliere un'azione che massimizzi la sua misura di prestazione attesa, date le informazioni fornite dalla sequenza percettiva e da ogni ulteriore conoscenza dell'agente.

Dobbiamo distinguere accuratamente il concetto di razionalità da quello di onniscienza.
Un agente razionale **non è onnisciente**. Un agente onnisciente, infatti, conosce il risultato effettivo delle sue azioni e può agire di conseguenza; in pratica è capace di prevedere il futuro, e questo nella realtà è impossibile. Potrà anche comportarsi nella maniera più razionale possibile, ma potranno comunque accadere degli imprevisti che rovineranno la sua performance, e non è colpa sua perché chiaramente non poteva conoscerli a priori. Razionalità allora *non significa "perfezione*". Infatti la razionalità massimizza il risultato atteso, mentre la perfezione quello reale.

L'agente razionale non deve limitarsi a raccogliere informazioni, ma dev'essere anche in grado di **imparare** il più possibile sulla base delle proprie percezioni. Generalmente il programmatore fornisce una conoscenza iniziale all'agente; la quale può essere modificata e può aumentare man mano che l'agente accumula esperienza. Esistono ovviamente anche dei *casi limite* nei quali l'ambiente è completamente conosciuto a priori, e quindi in tal caso l'agente non ha bisogno di percepire o imparare; può semplicemente agire nel modo corretto.

Quando un agente si appoggia totalmente alla conoscenza iniziale inserita dal programmatore invece che sulle proprie percezioni per cercare di arricchirla, diciamo che l'agente manca di **autonomia**. Un agente razionale dovrebbe essere autonomo, e imparare il più possibile per compensare e arricchire la conoscenza iniziale di cui è dotato (anche perché è solo iniziale… cioè, è molto scarsa e potrebbe essere anche *erronea*).

È chiaro che è dunque ragionevole fornire a un agente intelligente artificiale (oltre all'abilità di imparare) un po' di conoscenza inziale con la speranza che, dopo aver accumulato una sufficiente esperienza dell'ambiente, il comportamento dell'agente razionale possa diventare a tutti gli effetti *indipendente* dalla conoscenza pregressa.

2.3 La natura degli ambienti

Specificare un ambiente
Gli ambienti sono essenzialmente i problemi di cui gli agenti razionali rappresentano le "soluzioni". Quando abbiamo discusso la razionalità di un agente abbiamo dovuto specificare la misura di prestazione, l'ambiente esterno, gli attuatori e i sensori dell'agente. Tutto ciò può essere raggruppato nel termine specifico **task environment**, che potremmo tradurre "ambiente operativo". In alternativa, possiamo parlare di descrizione PEAS (*Performance*, *Environment*, *Actuators*, *Sensors*). Quando si progetta un agente, il primo passo dovrebbe sempre corrispondere alla specifica del task environment, la più ricca possibile.

Esempio: PEAS dell'ambiente operativo del taxi
Cerchiamo di fornire una descrizione PEAS dell'ambiente operativo del taxi. Dobbiamo procedere punto per punto, rispondendo alle seguenti domande (implicite):

1. A quale **misura di prestazione** dovrebbe aspirare il nostro pilota automatico?
   Gli aspetti desiderabili includono: arrivare alla destinazione in maniera corretta, minimizzare la durata temporale del viaggio e il suo costo o anche minimizzare il consumo di carburante e l'usura. Da notare che se abbiamo obiettivi *in contrasto*, occorrerà trovare una qualche forma di compromesso.

2. Qual è l'**ambiente** nel quale il taxi dovrà operare?
   Ogni tassista dev'essere in grado di muoversi in una varietà di strade, le quali contengono a loro volta altri veicoli, pedoni, animali randagi o lavori in corso.

3. Quali sono gli **attuatori** disponibili al pilota automatico?
   Gli attuatori disponibili al pilota automatico sono più o meno quelli utilizzati da un autista umano, vale a dire: acceleratore, sterzo, freno…

4. Quali sono i **sensori** di cui è dotato il pilota automatico?
   I sensori devono permettere al pilota automatico di sapere dove si trova, cosa c'è sulla strada, e quanto velocemente si sta muovendo, affinché raggiunga i propri obiettivi. Quindi, come sensori, dovrebbe avere: almeno una o più telecamere, tachimetro, contachilometri, accelerometro…

Dopo aver risposto a queste quattro domande, siamo in grado di fornire una descrizione PEAS del task environment dell'esempio in questione, che nel nostro caso è un taxi automatizzato. Una descrizione possibile (un po' più accurata, perché fornisce più risposte) è rappresentata nella figura seguente:

tipo di agente	misura di prestazione	ambiente	attuatori	sensori
guidatore di taxi	sicuro, veloce, ligio alla legge, viaggio confortevole, profitti massimi	strada, altri veicoli nel traffico, pedoni, clienti	sterzo, acceleratore, freni, frecce, clacson, schermo di interfaccia	telecamere, sonar, tachimetro, GPS, contachilo-metri, accelerometro, sensori sullo stato del motore, tastiera

Proprietà degli ambienti
La varietà degli ambienti operativi nell'IA è naturalmente molto vasta. È comunque possibile identificare un numero relativamente piccolo di proprietà base a cui suddividerli in categorie. Possiamo dunque avere le seguenti categorie di ambiente:

- **Completamente/parzialmente osservabile:** se i sensori di un agente gli danno accesso allo stato completo dell'ambiente in ogni momento l'ambiente è *completamente osservabile*, altrimenti lo è *solo parzialmente* (la parziale osservabilità può essere dovuta alla presenza di rumore o al problema di avere sensori inaccurati).
- **Deterministico/stocastico:** se lo stato successivo dell'ambiente è completamente determinato dallo stato corrente e dall'azione eseguita dall'agente, allora si può dire che l'ambiente è *deterministico*; in caso contrario si dice che è *stocastico*. Inoltre, se l'ambiente è deterministico in tutto, tranne che per le azioni di altri agenti, si dice che è *strategico*.
- **Episodico/sequenziale:** in un ambiente operativo *episodico*, l'esperienza dell'agente è divisa in episodi atomici. Ogni episodio consiste nella percezione dell'agente seguita dall'esecuzione di una singola azione. L'aspetto fondamentale è che l'episodio non dipende dalle azioni intraprese in quelli precedenti: negli ambienti episodici, la scelta dell'azione, dipende solo dall'episodio corrente. Negli ambienti *sequenziali,* al contrario, ogni decisione può influenzare tutte quelle successive.
- **Statico/dinamico:** se l'ambiente può cambiare mentre l'agente sta pensando, allora diciamo che è *dinamico* per quell'agente; in caso contrario, diciamo che è *statico*. Ad esempio, guidare un taxi è chiaramente dinamico: le altre macchine e il taxi stesso continuano a muoversi mentre l'algoritmo di guida pondera la mossa successiva. Un cruciverba, al contrario, è statico. Se l'ambiente stesso non cambia al passare del tempo, ma la valutazione delle prestazioni dell'agente sì, allora diciamo che l'ambiente è *semidinamico*.
- **Discreto/continuo:** la distinzione tra *discreto* e *continuo* può essere applicata allo *stato* dell'ambiente, al modo in cui è gestito il *tempo*, alle *percezioni*, e *azioni* dell'agente. Ad esempio, un ambiente a stati discreti, come una scacchiera, ha un numero finito di stati distinti. La guida di un taxi, invece, è un problema con stato e tempo continui: la velocità e la posizione del taxi e degli altri veicoli cambiano con continuità al passare del tempo.
- **Agente singolo/multiagente:** la distinzione tra ambienti ad *agente singolo* e *multiagente* è abbastanza ovvia: nel primo caso, nell'ambiente, opera un solo agente, nel secondo due o più. Un ambiente multiagente può essere a sua volta **competitivo**, se un agente, cercando di massimizzare la sua performance, minimizza quelle degli altri (ad. es. il gioco degli scacchi → in tal caso gli agenti sono tra di loro *avversari*); oppure **cooperativo**, se invece, un agente, cercando di massimizzare la sua performance, massimizza le performance di tutti gli agenti (ad es. l'ambiente del traffico → un'auto guidando *al meglio* evita incidenti che coinvolgerebbero altre auto).

Come ci si potrebbe aspettare, il caso più difficile è quello: *parzialmente osservabile, stocastico, sequenziale, dinamico, continuo* e *multiagente*; e rappresenta la maggior parte delle situazioni *reali*.

Esempi di task environment e loro caratteristiche

task environment	osservabile	deterministico	episodico	statico	discreto	agenti
cruciverba	completamente	deterministico	sequenziale	statico	discreto	singolo
scacchi con orologio	completamente	strategico	sequenziale	semi	discreto	multi
poker	parzialmente	stocastico	sequenziale	statico	discreto	multi
backgammon	completamente	stocastico	sequenziale	statico	discreto	multi
autista di taxi	parzialmente	stocastico	sequenziale	dinamico	continuo	multi
diagnosi medica	parzialmente	stocastico	sequenziale	dinamico	continuo	singolo
analizzatore di immagini	completamente	deterministico	episodico	semi	continuo	singolo
robot di selezione per parti meccaniche	parzialmente	stocastico	episodico	dinamico	continuo	singolo
controllore per una raffineria	parzialmente	stocastico	sequenziale	dinamico	continuo	singolo
maestro di inglese interattivo	parzialmente	stocastico	sequenziale	dinamico	discreto	multi

2.4 La struttura degli agenti

Il compito dell'IA è progettare il **programma agente** che implementa la funzione agente, mettendo in relazione percezioni e azioni. Diamo per scontato che questo programma sarà eseguito da un computer dotato di sensori fisici e attuatori; questa prende il nome di **architettura**:

$$agente = architettura + programma$$

Programmi agente

Prendono come input la percezione corrente dei sensori e restituiscono un'azione agli attuatori. Si noti la differenza tra il programma agente, che prende come input *solamente la percezione corrente*, e la funzione agente, il cui input è costituito dall'*intera storia delle percezioni*. Il programma agente si basa sulla sola percezione corrente perché l'ambiente non è in grado di fornirgli nulla di più. Al massimo, se le sue azioni devono dipendere dalla sequenza percettiva precedente, dev'essere lui a preoccuparsi di *memorizzarle*.

Elenchiamo qui di seguito 5 tipi base di programma agente, che rappresentano i principi alla base di quasi tutti i sistemi intelligenti:

- agenti reattivi semplici
- agenti reattivi basati su modello
- agenti basati su obiettivi
- agenti basati sull'utilità
- agenti basati sull'apprendimento

Agenti reattivi semplici
Questi agenti scelgono le azioni sulla base della percezione *corrente*, ignorando tutta la storia percettiva precedente. Ad esempio, l'aspirapolvere è un agente reattivo semplice, perché la sua decisione è basata unicamente sulla posizione corrente e se questa contiene dello sporco o no. Qui di seguito è riportato il programma agente per l'agente reattivo semplice nell'ambiente dell'aspirapolvere a due stati:

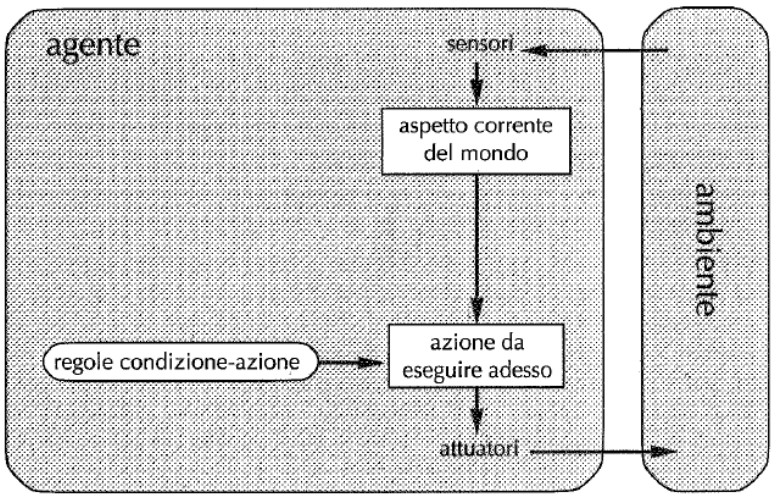
```
function AGENTE-REATTIVO-ASPIRAPOLVERE([posizione, stato]) returns un'azione

    if stato = Sporco then return Aspira
    else if posizione = A then return Destra
    else if posizione = B then return Sinistra
```

Questo programma implementa la funzione agente corrispondente alla tabella che abbiamo trattato prima, anche se è molto più piccolo (e meno complesso) di quest'ultima, perché avendo ignorato (qui) la storia delle percezioni, il numero di possibilità chiaramente si è ridotto. È importante considerare anche la **regola condizione-azione**: l'agente, effettuando alcuni calcoli sulle percezioni che rileva tramite i sensori, stabilisce *se* e *quando* una certa condizione (che descriviamo con una frase) si verifica, e agisce in risposta a questa mediante gli attuatori. Ad esempio:

$$\boldsymbol{if} \quad la-macchina-davanti-frena \quad \boldsymbol{then} \quad inizia-a-frenare$$

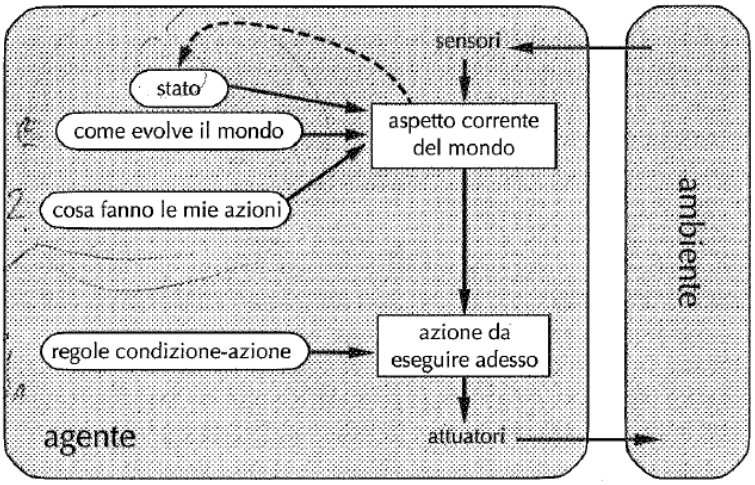
Qui di seguito la struttura di questo programma generale, che mostra come le regole condizione-azione permettono all'agente di stabilire una connessione da percezione ad azione:



Gli agenti reattivi semplici hanno la proprietà di essere, appunto, semplici, ma la loro intelligenza è molto limitata. L'agente funzionerà *solo se si può selezionare la decisione corretta in base alla sola percezione corrente, ovvero solo nel caso in cui l'ambiente sia completamente osservabile*. Anche una minima parte di non-osservabilità può causare grandi problemi.

Agenti reattivi basati su modello

A differenza degli agenti reattivi semplici, permettono di gestire la parziale osservabilità, e lo fanno nel modo più efficace: *tenendo traccia della parte del mondo che non possono vedere nell'istante corrente*. In pratica, l'agente memorizza una sorta di **stato interno** che dipende dalla storia delle percezioni e che quindi riflette almeno una parte degli aspetti non osservabili dello stato corrente. Nel passare del tempo l'agente aggiorna questo stato interno, e lo fa utilizzando due tipi di informazione: informazioni sull'evoluzione del mondo indipendentemente dalle sue azioni, e informazioni sull'effetto che hanno sul mondo le azioni dell'agente stesso. Questa conoscenza sul "funzionamento del mondo" viene chiamata **modello** del mondo, e un agente che si appoggia a un simile modello prende il nome appunto di **agente basato su modello**. La seguente figura illustra la struttura di un agente reattivo dotato di stato interno, mostrando come la descrizione aggiornata dello stato scaturisce dalla combinazione del vecchio stato interno e della percezione corrente.



Agenti basati su obiettivi

Conoscere lo stato corrente dell'ambiente non sempre basta a decidere che cosa fare. In alcune situazioni, oltre che della descrizione dello stato corrente, l'agente ha bisogno di qualche tipo di informazione riguardante il suo **obiettivo** (***goal***), che descriva situazioni *desiderabili*. Il programma agente può unire quest' informazione a quella che riguarda i risultati delle possibili azioni, per scegliere quelle che portano al soddisfacimento dell'obiettivo. Talvolta scegliere un'azione in base a un obiettivo è molto semplice, quando questo può essere raggiunto in un solo passo. Altre volte è più difficile, quando l'agente deve considerare lunghe sequenze di azioni alternative per trovare il "cammino" che porta al risultato desiderato. La **ricerca** e la **pianificazione** sono dei sottocampi dell'IA dedicati proprio ad identificare le sequenze di azioni che permettono a un agente di raggiungere i propri obiettivi. Benché un agente basato su obiettivi sembri meno efficiente, d'altra parte è più flessibile, perché la conoscenza che motiva le sue decisioni è rappresentata esplicitamente e *può essere modificata*. Il comportamento dell'agente basato su obiettivi può essere facilmente alterato per farlo andare verso una *destinazione diversa*. Al contrario, le regole di un agente reattivo, funzioneranno solo per una singola destinazione finale; per andare da qualche altra parte dovranno essere tutte riscritte. Qui di seguito un agente dotato di modello del mondo e basato su obiettivi. Oltre a tener traccia dello stato dell'ambiente, l'agente memorizza un insieme di obiettivi e sceglie l'azione che lo porterà (a un certo punto) a soddisfarli: