



UNIVERSITÀ DEGLI STUDI DI SALERNO  
**DIPARTIMENTO DI INFORMATICA**

Laurea triennale in Informatica

# Fondamenti di Intelligenza Artificiale

Lezione 2 - Agenti Intelligenti



# Agenti Intelligenti

## Definizione di agente

Un **agente** è qualsiasi cosa possa essere vista come un sistema che *percepisce* il suo **ambiente** attraverso dei **sensori** ed *agisce* su di esso tramite degli **attuatori**.



# Agenti Intelligenti

## Definizione di agente

Un **agente** è qualsiasi cosa possa essere vista come un sistema che *percepisce* il suo **ambiente** attraverso dei **sensori** ed *agisce* su di esso tramite degli **attuatori**.



Agente umano

### Sensori

Occhi  
Orecchie  
Naso  
...

### Attuatori

Mani  
Gambe  
Tratto vocale  
...



Agente robotico

### Sensori

Telecamere  
Telemetri  
...

### Attuatori

Motori



Agente software

### Sensori

Battute di tasti  
Contenuto di file  
Pacchetti di rete  
...

### Attuatori

Diversa visualizzazione  
Scrittura file  
Invio pacchetti di rete  
...

# Agenti Intelligenti

## Definizione di agente

Un **agente** è qualsiasi cosa possa essere vista come un sistema che *percepisce* il suo **ambiente** attraverso dei **sensori** ed *agisce* su di esso tramite degli **attuatori**.

**Percezione.** Insieme di input percettivi dell'agente in un dato istante.

**Sequenza percettiva.** La storia completa di tutto ciò che l'agente ha percepito nella sua esistenza.

Più in generale, è bene dire che la scelta di un'azione da parte dell'agente, in un qualsiasi istante, può dipendere dall'intera sequenza percettiva osservata fino a quel momento, ma **non** da qualcosa che non abbia percepito.

Specificando l'azione prescelta dall'agente per ogni possibile sequenza percettiva abbiamo descritto l'agente in modo completo.

Se volessimo dirla in termini matematici, allora il comportamento di un agente è descritto da una **funzione agente**, che descrive la corrispondenza tra una qualsiasi sequenza percettiva ed una specifica azione.

Da un punto di vista pratico, la funzione agente è implementata tramite la specifica di un **programma agente**.

# Agenti Intelligenti

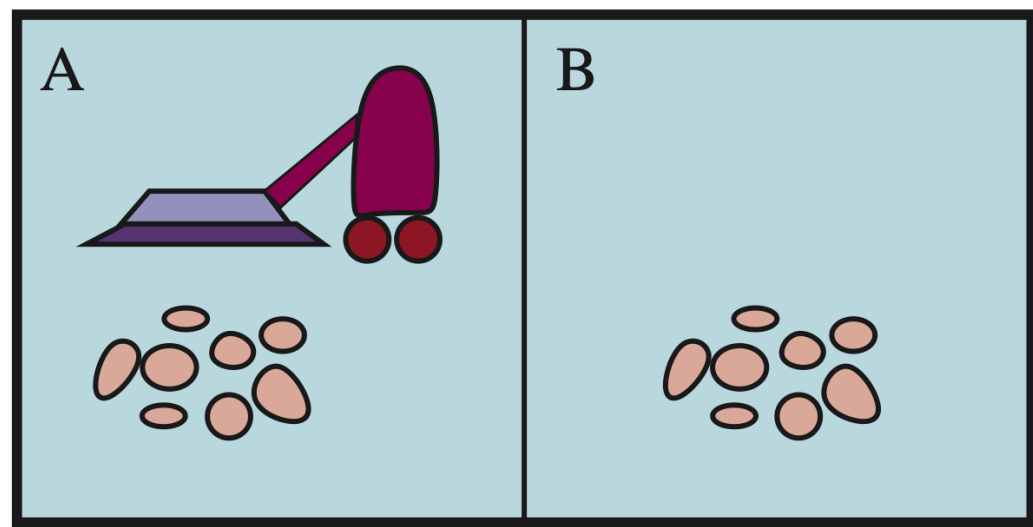
## Definizione di agente

Un **agente** è qualsiasi cosa possa essere vista come un sistema che *percepisce* il suo **ambiente** attraverso dei **sensori** ed *agisce* su di esso tramite degli **attuatori**.

La sequenza percettiva e la corrispondente azione attuata dall'agente può essere espressa attraverso una tabella. Consideriamo un esempio:

### Il mondo dell'aspirapolvere

Questo mondo è così semplice che potremo esprimere tutto ciò che succede



Ci sono due posizioni, ovvero i riquadri A e B. L'agente aspirapolvere percepisce in che riquadro si trova e se c'è dello sporco in tale locazione

Può decidere se spostarsi verso sinistra o destra, aspirare o non fare nulla. Di conseguenza, la funzione agente sarà molto semplice:

### Funzione agente.

Se il riquadro corrente è sporco, aspira.  
Altrimenti, muoviti nell'altro riquadro.

# Agenti Intelligenti

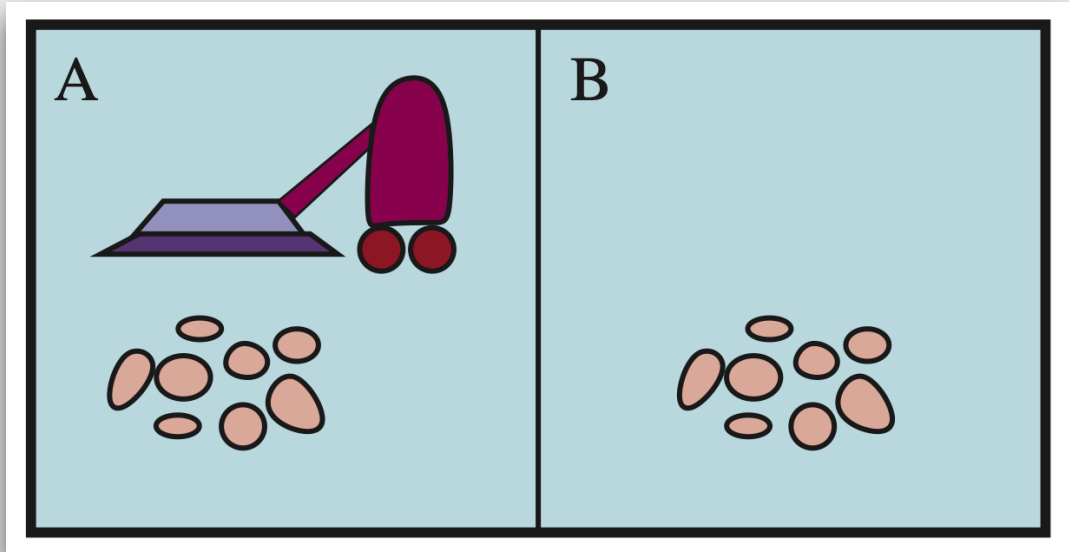
## Definizione di agente

Un **agente** è qualsiasi cosa possa essere vista come un sistema che *percepisce* il suo **ambiente** attraverso dei **sensori** ed *agisce* su di esso tramite degli **attuatori**.

La sequenza percettiva e la corrispondente azione attuata dall'agente può essere espressa attraverso una tabella. Consideriamo un esempio:

Il mondo dell'aspirapolvere

Questo mondo è così semplice che potremo esprimere tutto ciò che succede



Percept sequence	Action
[A, Clean]	Right
[A, Dirty]	Suck
[B, Clean]	Left
[B, Dirty]	Suck
[A, Clean], [A, Clean]	Right
[A, Clean], [A, Dirty]	Suck
⋮	⋮
[A, Clean], [A, Clean], [A, Clean]	Right
[A, Clean], [A, Clean], [A, Dirty]	Suck
⋮	⋮

### Funzione agente.

Se il riquadro corrente è sporco, aspira. Altrimenti, muoviti nell'altro riquadro.

Come possiamo facilmente notare, possiamo esprimere diversi agenti semplicemente modificando la colonna di destra!



# Agenti Intelligenti

## Definizione di agente

Un **agente** è qualsiasi cosa possa essere vista come un sistema che *percepisce* il suo **ambiente** attraverso dei **sensori** ed *agisce* su di esso tramite degli **attuatori**.

Il ch  ci porta ad una domanda: **Qual   il modo corretto di progettare la tabella?**

O, in altri termini: **Cosa rende un agente buono o cattivo, intelligente o stupido?**

Prestazione

Conoscenza pregressa

Azioni che l'agente pu  compiere

Sequenza percettiva fino all'istante corrente

**Razionalit .** La razionalit  di un agente dipende da questi quattro fattori —> Come accade per un agente umano, questo   razionale in base alle sue conoscenze, al suo obiettivo, oltre che al contesto che lo circonda.

**Agente razionale.** Per ogni possibile sequenza di percezioni, un agente razionale dovrebbe scegliere un'azione che massimizzi il valore atteso della sua *misura di prestazione*, date le informazioni fornite dalla sequenza percettiva e da ogni ulteriore conoscenza dell'agente.

**Misura di prestazione.** Come regola generale, la misura dovrebbe essere progettata sulla base dell'*effetto che si desidera osservare sull'ambiente e non su come dovrebbe comportarsi l'agente*. Altrimenti, un agente otterrebbe una razionalit  perfetta semplicemente "illudendosi" di fare la cosa giusta.

# Agenti Intelligenti

## Definizione di agente

Un **agente** è qualsiasi cosa possa essere vista come un sistema che *percepisce* il suo **ambiente** attraverso dei **sensori** ed *agisce* su di esso tramite degli **attuatori**.

Ma attenzione: la *razionalità* non implica l'*onniscienza*.

Un **agente onnisciente** conosce il risultato effettivo delle sue azioni e può agire di conseguenza. Sfortunatamente, nel mondo reale l'onniscienza è impossibile.

Un **agente razionale** fa la cosa “*giusta*” sulla base del contesto in cui opera. Razionalità, pertanto, non significa perfezione ma piuttosto **massimizzazione del risultato atteso**.

**Information gathering.** Per massimizzare il risultato atteso, è possibile (e a volte opportuno) intraprendere azioni di information gathering, ovvero azioni che hanno la sola finalità di raccogliere informazioni sull'ambiente circostante e che consentono di modificare le percezioni future.

Ad esempio, l'azione di guardare a destra e sinistra prima di attraversare la strada è un'operazione di information gathering che consente di modificare le nostre percezioni e di essere più razionali rispetto all'obiettivo di raggiungere l'altro lato della strada sani e salvi.

Un secondo esempio di information gathering è detto **esplorazione**, ovvero l'azione necessaria per far “conoscenza” di un ambiente sconosciuto —> L'esplorazione è particolarmente rilevante per gli algoritmi di ricerca che vedremo presto.



# Agenti Intelligenti

## Definizione di agente

Un **agente** è qualsiasi cosa possa essere vista come un sistema che *percepisce* il suo **ambiente** attraverso dei **sensori** ed *agisce* su di esso tramite degli **attuatori**.

Ma attenzione: la *razionalità* non implica l'*onniscienza*.

Un **agente onnisciente** conosce il risultato effettivo delle sue azioni e può agire di conseguenza. Sfortunatamente, nel mondo reale l'onniscienza è impossibile.

Un **agente razionale** fa la cosa “*giusta*” sulla base del contesto in cui opera. Razionalità, pertanto, non significa perfezione ma piuttosto **massimizzazione del risultato atteso**.

**Information gathering.** Per massimizzare il risultato atteso, è possibile (e a volte opportuno) intraprendere azioni di information gathering, ovvero azioni che hanno la sola finalità di raccogliere informazioni sull'ambiente circostante e che consentono di modificare le percezioni future.

è  
zioni e di  
e salvi.  
Oltre che l'esplorazione, la definizione di razionalità prevede il concetto di **apprendimento**, ovvero la capacità di imparare coppie di percezioni-azioni sulla base delle azioni e dei rispettivi risultati ottenuti in precedenza.

ne  
Un agente che intraprende azioni sulla base della sola conoscenza inserita in fase di progettazione e non delle sue percezioni manca di *autonomia*.

—> Il concetto di apprendimento è alla base degli algoritmi di machine learning che vedremo nella seconda parte del corso.

# Agenti Intelligenti

## Definizione di ambiente

Un **ambiente** è essenzialmente un'istanza di un problema di cui gli agenti razionali rappresentano le soluzioni.

Un ambiente viene generalmente descritto tramite la formulazione **PEAS**, ovvero **P**erformance, **E**nvironment, **A**ctuators, **S**ensors.

**P.** Misura di prestazione adottata per valutare l'operato di un agente.

**E.** Descrizione degli elementi che formano l'ambiente.

**A.** Gli attuatori disponibili all'agente per intraprendere le azioni.

**S.** I sensori attraverso i quali riceve gli input percettivi.

Tipo di ambiente	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, tachimetro, GPS, contachilometri, accelerometro, sensori sullo stato del motore, tastiera

# Agenti Intelligenti

## Definizione di ambiente

Un **ambiente** è essenzialmente un'istanza di un problema di cui gli agenti razionali rappresentano le soluzioni.

Tipo di ambiente	Misura di prestazione	Ambiente	Attuatori	Sensori
Sistema di diagnosi medica	Paziente sano, minimizzare i costi e le denunce	Paziente, ospedale, staff medico	Schermo per visualizzare domande, test, diagnosi, trattamenti	Tastiera per l'inserimento dei sintomi, dei risultati, delle risposte del paziente
Sistema di analisi di immagini satellitari	Categorizzazione corretta delle immagini	Collegamento verso terra di un satellite orbitante	Visualizzazione della categorizzazione della scena	Array di pixel sensibili al colore
Robot selezionatore di parti meccaniche	Percentuale di pezzi inseriti nei contenitori giusti	Nastro trasportatore con parti meccaniche, contenitori	Braccio meccanico con manipolatore	Telecamera, sensori di posizionamento del braccio meccanico
Tutor interattivo per lo studio dell'inglese	Risultati del test dello studente	Insieme di studenti, scuola erogatrice di test	Visore per proporre esercizi, suggerimenti, correzioni	Tastiera

# Agenti Intelligenti

## Proprietà degli ambienti

Se la varietà di ambienti operativi nell'intelligenza artificiale è molto vasto, possiamo identificare un numero relativamente piccolo di dimensioni per caratterizzarli.

- > **Completamente osservabili (o parzialmente osservabili).** I sensori di un ambiente danno accesso allo stato completo dell'ambiente in ogni momento.
- > **Deterministico (o stocastico).** Lo stato successivo dell'ambiente è completamente determinato dallo stato corrente e dall'azione eseguita dall'agente.
- > **Episodico (o sequenziale).** L'esperienza dell'agente è divisa in “episodi” atomici, dove ciascun episodio consiste nell'eseguire una singola azione. La scelta dell'azione in ciascun episodio dipende dall'episodio stesso.
- > **Statico (o dinamico).** L'ambiente è invariato mentre un agente sta deliberando. L'ambiente si dice semi-dinamico se esso non cambia con il passare del tempo ma il punteggio delle prestazioni dell'agente lo fa.
- > **Discreto (o continuo).** L'ambiente fornisce un numero limitato di percezioni e azioni distinte, chiaramente definite. La dama è un esempio di ambiente discreto, mentre l'auto a guida autonoma è un esempio di ambiente continuo.
- > **Singolo (o multi-agente).** L'ambiente consente la presenza di un unico agente. Se ci sono più agenti, gli scenari cambiano, come vedremo quando, negli algoritmi di ricerca, parleremo di teoria dei giochi.

# Agenti Intelligenti

## Proprietà degli ambienti

Se la varietà di ambienti operativi nell'intelligenza artificiale è molto vasto, possiamo identificare un numero relativamente piccolo di dimensioni per caratterizzarli.

—> **Singolo (o multi-agente).** L'ambiente consente la presenza di un unico agente. Se ci sono più agenti, gli scenari cambiano, come vedremo quando, negli algoritmi di ricerca, parleremo di teoria dei giochi.

A questo proposito, vale la pena fornire una seconda classificazione:

—> **Competitivo.** L'ambiente è competitivo se, in un ambiente multi-agente, il comportamento di un agente A è tale da massimizzare/minimizzare la misura di prestazione del comportamento di B (e viceversa).

—> **Cooperativo.** L'ambiente è cooperativo se, in un ambiente multi-agente, due agenti A e B puntano a massimizzare/minimizzare la stessa misura di prestazione.

Detto questo, facciamo qualche esempio.

# Agenti Intelligenti

## Proprietà degli ambienti

Se la varietà di ambienti operativi nell'intelligenza artificiale è molto vasto, possiamo identificare un numero relativamente piccolo di dimensioni per caratterizzarli.

Tipo di ambiente	Osservabile	Agenti	Deterministico	Episoldico	Statico	Discreto
Cruciverba Scacchi con orologio	Completamente Completamente	Singolo Multi	Deterministico Deterministico	Sequenziale Sequenziale	Statico Semi	Discreto Discreto
Poker Backgammon	Parzialmente Completamente	Multi Multi	Stocastico Stocastico	Sequenziale Sequenziale	Statico Statico	Discreto Discreto
Autista di taxi Diagnosi medica	Parzialmente Parzialmente	Multi Singolo	Stocastico Stocastico	Sequenziale Sequenziale	Dinamico Dinamico	Discreto Discreto



# Agenti Intelligenti

## Struttura degli agenti

Il compito dell'intelligenza artificiale è progettare il programma agente, il quale consiste nell'implementazione della funziona agente.

Fin qui, parlando di agenti, ci siamo limitati a descrivere il loro *comportamento*, ovvero l'azione eseguita in corrispondenza di una data sequenza di percezioni.

Volendo fare un passo verso la progettazione del programma agente e dando per scontato che tale programma sarà eseguito da un computer dotato di sensori e attuatori, i quali formano la cosiddetta architettura, diremo che:

*agente = architettura + programma.*

I programmi agente che progetteremo in questo corso hanno la stessa struttura: prendono come input la percezione corrente dei sensori e restituiscono un'azione agli attuatori.

E' importante notare la differenza tra **programma** e **funzione** agente: il primo prende in input solo la percezione corrente, la seconda l'intera storia percettiva.

Più in generale, la sfida principale dell'intelligenza artificiale è trovare il modo di scrivere programmi che, nella massima misura possibile, producano comportamento razionale con una **piccola quantità di codice** invece che con la **rappresentazione di tutti gli input** percettivi possibili.

# Agenti Intelligenti

## Struttura degli agenti

Il compito dell'intelligenza artificiale è progettare il programma agente, il quale consiste nell'implementazione della funziona agente.

Proprio alla luce di voler raggiungere l'obiettivo di scrivere quanto meno codice possibile, un approccio basato su tabelle è giocoforza fallimentare.

Sia  $P$  l'insieme delle possibili percezioni e  $T$  la durata di vita dell'agente (ovvero, il numero totale di percezioni che riceverà). La tabella risultante avrà un numero di righe pari a  $\sum_{t=1}^T |P|^t$ .

Considerando il caso del gioco degli scacchi, la tabella avrà  $10^{150}$  righe... come riferimento, basti pensare che il **numero di atomi nell'universo è di poco inferiore  $10^{80}$** .

```
function AGENTE-CON-TABELLA(percezione) returns un'azione
  persistent: percezioni, una sequenza inizialmente vuota;
               tabella, una tabella di azioni, indicizzata per sequenze percettive, completamente
               specificata dall'inizio;

  aggiungi percezione alla fine di percezioni
  azione ← LOOKUP(percezioni, tabella)

  return azione
```

# Agenti Intelligenti

## Struttura degli agenti

Il compito dell'intelligenza artificiale è progettare il programma agente, il quale consiste nell'implementazione della funziona agente.

Proprio alla luce di voler raggiungere l'obiettivo di scrivere quanto meno codice possibile, un approccio basato su tabelle è giocoforza fallimentare.

Sia  $P$  l'insieme delle possibili percezioni e  $T$  la durata di vita dell'agente (ovvero, il numero totale di percezioni che riceverà). La tabella risultante avrà un numero di righe pari a  $\sum_{t=1}^T |P|^t$ .

Considerando il caso del gioco degli scacchi, la tabella avrà  $10^{150}$  righe... come riferimento, basti pensare che il **numero di atomi nell'universo è di poco inferiore  $10^{80}$** .

```
function AGENTE-CON-TABELLA(percezione) returns un'azione
  persistent: percezioni, una sequenza inizialmente vuota;
              tabella, una tabella di azioni, indicizzata per sequenze percettive, completamente
              specificata dall'inizio;

  aggiungi percezione alla fine di percezioni
  azione ← LOOKUP(percezioni, tabella)

  return azione
```

← /\* ogni percezione ricevuta viene inserita nella sequenza percettiva \*/

# Agenti Intelligenti

## Struttura degli agenti

Il compito dell'intelligenza artificiale è progettare il programma agente, il quale consiste nell'implementazione della funziona agente.

Proprio alla luce di voler raggiungere l'obiettivo di scrivere quanto meno codice possibile, un approccio basato su tabelle è giocoforza fallimentare.

Sia  $P$  l'insieme delle possibili percezioni e  $T$  la durata di vita dell'agente (ovvero, il numero totale di percezioni che riceverà). La tabella risultante avrà un numero di righe pari a  $\sum_{t=1}^T |P|^t$ .

Considerando il caso del gioco degli scacchi, la tabella avrà  $10^{150}$  righe... come riferimento, basti pensare che il **numero di atomi nell'universo è di poco inferiore  $10^{80}$** .

```
function AGENTE-CON-TABELLA(percezione) returns un'azione
```

```
  persistent: percezioni, una sequenza inizialmente vuota;
```

```
             tabella, una tabella di azioni, indicizzata per sequenze percettive, completamente  
             specificata dall'inizio;
```

```
  aggiungi percezione alla fine di percezioni
```

```
  azione ← LOOKUP(percezioni, tabella)
```

```
  return azione
```

/\* la funzione LOOKUP andrà semplicemente  
a consultare la tabella e ricercherà la prossima  
azione da compiere in relazione alla sequenza  
percettiva attuale e alla tabella delle azioni \*/

Studiamo quindi alcune alternative più efficienti/efficaci. Tutti gli agenti che vedremo sono classificati come **learning agents**, ovvero agenti capaci di migliorare le loro prestazioni e attuare migliori azioni attraverso l'apprendimento.

# Agenti Intelligenti

## Agenti Reattivi Semplici

Questi agenti intraprendono azioni sulla base della percezione corrente, ignorando la storia percettiva pregressa. Pertanto, non hanno memoria.



# Agenti Intelligenti

## Agenti Reattivi Semplici

Questi agenti intraprendono azioni sulla base della percezione corrente, ignorando la storia percettiva pregressa. Pertanto, non hanno memoria.

Il grosso vantaggio degli agenti reattivi semplici si ottiene in termini di rappresentazione delle percezioni. Nel caso del mondo dell'aspirapolvere,  $4^T = 4$ .

Un agente reattivo semplice fa uso delle regole condizione-azione per decidere quale sarà la prossima azione da compiere. Le regole sono banalmente espresse come degli ***if-then***. Ad esempio:

***if la-macchina-davanti-frena then inizia-a-frenare***

```
function AGENTE-REATTIVO-SEMPLICE(percezione) returns un'azione  
  persistent: regole, un insieme di regole condizione-azione;
```

```
  stato <- INTERPRETA-INPUT(percezione)  
  regola <- REGOLA-CORRISPONDENTE(stato, regole)  
  azione <- regola.AZIONE
```

```
return azione
```

/\* piuttosto che implementare agenti reattivi semplici ad-hoc per specifici problemi, questa implementazione consente di rendere l'agente usabile in più contesti, a patto di specificare le regole specifiche. \*/

Un agente reattivo semplice funziona bene solo se l'ambiente è completamente osservabile (altrimenti, come specificheremmo le regole condizione-azione?)

Ma cosa succede in caso di dati mancanti?

regole c

agen



# Agenti Intelligenti

## Agenti Reattivi Semplici

Questi agenti intraprendono azioni sulla base della percezione corrente, ignorando la storia percettiva pregressa. Pertanto, non hanno memoria.

In alcuni casi, un agente reattivo semplice che opera in ambienti parzialmente osservabili può incappare in *cicli infiniti*.

Una soluzione a questo problema è quella di definire una **componente casuale**, la quale verrà invocata nel caso di dati mancanti per poter compiere un'azione casuale e, quindi evitare cicli infiniti.

**Ma attenzione:** come vedremo spesso nel corso, una componente casuale porta solitamente ad un comportamento razionale. Sfortunatamente, non è questo il caso: la componente casuale serve solo ad un agente reattivo semplice per poter evitare cicli infiniti, ma non “crea” intelligenza.

regole c

agen

# Agenti Intelligenti

## Agenti Reattivi basati su Modello

Un agente con due tipi di conoscenza: (1) come evolve il mondo, indipendentemente dal suo stato; (2) informazioni sull'impatto delle sue azioni sull'ambiente.



# Agenti Intelligenti

## Agenti Reattivi basati su Modello

Un agente con due tipi di conoscenza: (1) come evolve il mondo, indipendentemente dal suo stato; (2) informazioni sull'impatto delle sue azioni sull'ambiente.

La conoscenza del mondo, sviluppata mediante una teoria scientifica completa, viene chiamata “modello” del mondo.

```
function AGENTE-REATTIVO-BASATO-SU-MODELLO(percezione) returns un'azione
  persistent: stato, la concezione corrente dello stato del mondo da parte dell'agente;
               modello, una descrizione della dipendenza dello stato successivo dallo stato
                 corrente e dall'azione;
               regole, un insieme di regole condizione-azione;
               azione, l'azione più recente, inizialmente nessuna;

  stato ← AGGIORNA-STATO(stato, azione, percezione, modello)
  regola ← REGOLA-CORRISPONDENTE(stato, regole)
  azione ← regola.AZIONE

  return azione
```

E' importante notare che la variabile *stato* non necessariamente rappresenta con *esattezza* lo stato del mondo, ma è la migliore ipotesi che l'agente può fare sul suo stato.

come e

cosa

regole co

agent

# Agenti Intelligenti

## Agenti basati su Obiettivi

Un agente che aggiunge al modello informazioni sugli obiettivi specifici che si desiderano raggiungere.



# Agenti Intelligenti

## Agenti basati su Obiettivi

Un agente che aggiunge al modello informazioni sugli obiettivi specifici che si desiderano raggiungere.

In altri termini, conoscere lo stato corrente dell'ambiente non sempre basta a decidere cosa fare. Per questo incorporare degli obiettivi può aiutare l'agente a compiere azioni più intelligenti e razionali che portino ad una soluzione più rapida.

**Ma attenzione:** il tipo di decisioni prese da un agente di questo tipo non ha niente a che vedere con le regole condizione-azione viste per gli agenti basati su modello.

Infatti, in questo caso dobbiamo prendere in considerazione il futuro sotto due aspetti: (1) Cosa accadrà se faccio così? E (2) Se faccio questo verrò soddisfatto?

Ovvero, le decisioni prese dall'agente non sono necessariamente deterministiche e basate su esplicite regole ma dipendono da come cambierà l'ambiente in funzione delle azioni e degli obiettivi da raggiungere!

Nel corso delle prossime lezioni, approfondiremo gli agenti basati su obiettivi in quanto la **ricerca** e la **pianificazione** sono sottocapi dell'intelligenza artificiale dedicati ad identificare sequenze di azioni che permettono ad un agente di raggiungere i proprio obiettivi.

come ev

cosa

agente

# Agenti Intelligenti

## Agenti basati sull'Utilità

Gli obiettivi non bastano in quanto consentono di esprimere in maniera binaria i risultati “buoni” da quelli “cattivi”. In situazioni reali, esistono situazioni “desiderabili” e non.





# Agenti Intelligenti

## Agenti basati sull'Utilità

Gli obiettivi non bastano in quanto consentono di esprimere in maniera binaria i risultati “buoni” da quelli “cattivi”. In situazioni reali, esistono situazioni “desiderabili” e non.

Una funzione di utilità assegna ad uno stato un numero reale che quantifica il grado di “desiderabilità” ad esso associato.

Quando ci sono più obiettivi tra di loro contrastanti, una funzione di utilità consente di determinare quale obiettivo preferire in base al punteggio ad esso assegnato dalla funzione utilità.

Nel contesto degli algoritmi di ricerca, vedremo alcuni casi in cui un agente basato su obiettivi deve necessariamente trasformarsi in un agente basato sull'utilità, ad esempio quando parleremo di algoritmi genetici multi-obiettivo.

come e

cosa

agen

# Agenti Intelligenti

## Agenti capaci di apprendere

L'apprendimento presenta il vantaggio di permettere agli agenti di operare in ambienti inizialmente sconosciuti diventando col tempo più competenti.

Un agente di questo tipo ha quattro componenti principali:

- > **Elemento di apprendimento.** L'elemento responsabile del miglioramento interno.
- > **Elemento esecutivo.** L'elemento responsabile della selezione delle azioni esterne. Questo è l'elemento che abbiamo considerato finora come agente.
- > **Elemento critico.** L'elemento responsabile di fornire feedback sulle prestazioni correnti dell'agente, così che l'elemento di apprendimento possa determinare se e come modificare l'elemento esecutivo affinché si comporti meglio in futuro.
- > **Generatore di problemi.** L'elemento responsabile di suggerire azioni che portino ad esperienze nuove e significative che, chiaramente, portino l'agente ad apprendere nuove conoscenze da sfruttare poi per migliorare le sue azioni.

# Agenti Intelligenti

## Agenti capaci di apprendere

L'apprendimento presenta il vantaggio di permettere agli agenti di operare in ambienti inizialmente sconosciuti diventando col tempo più competenti.



# Agenti Intelligenti

Riassumendo, l'IA dipende dagli agenti che il progettista considera...

## Artificial Intelligence

**Natural  
Language  
Processing**

**Reasoning**

...

**Planning**

## Search algorithms

**Meta-heuristics**

**Game Theory**

**Constraint  
Satisfaction Problems**

## Machine Learning

**Supervised learning**

**Unsupervised learning**

**Reinforcement learning**

**Deep learning**



UNIVERSITÀ DEGLI STUDI DI SALERNO  
**DIPARTIMENTO DI INFORMATICA**

Laurea triennale in Informatica

# Fondamenti di Intelligenza Artificiale

Lezione 2 - Agenti Intelligenti

