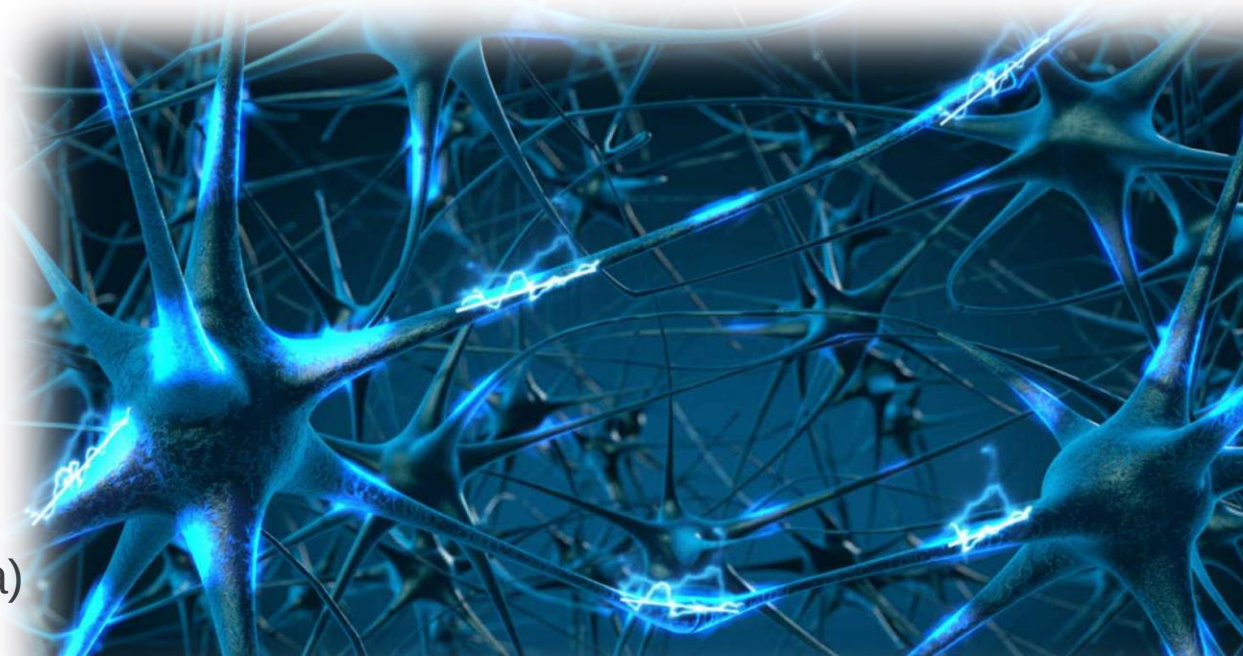
INTELLIGENZA ARTIFICIALE

AGIRE RAZIONALE - AGENTI (*)

Corsi di Laurea in Informatica, Ing. Gestionale, Ing. Informatica, Ing. di Internet

(a.a. 2024-2025)

Roberto Basili



(*) alcune *slides* sono di
Maria Simi (Univ. Pisa)

Intelligenza Artificiale

- L’intelligenza artificiale si occupa della
- 1. comprensione
- 2. riproduzione del comportamento intelligente.

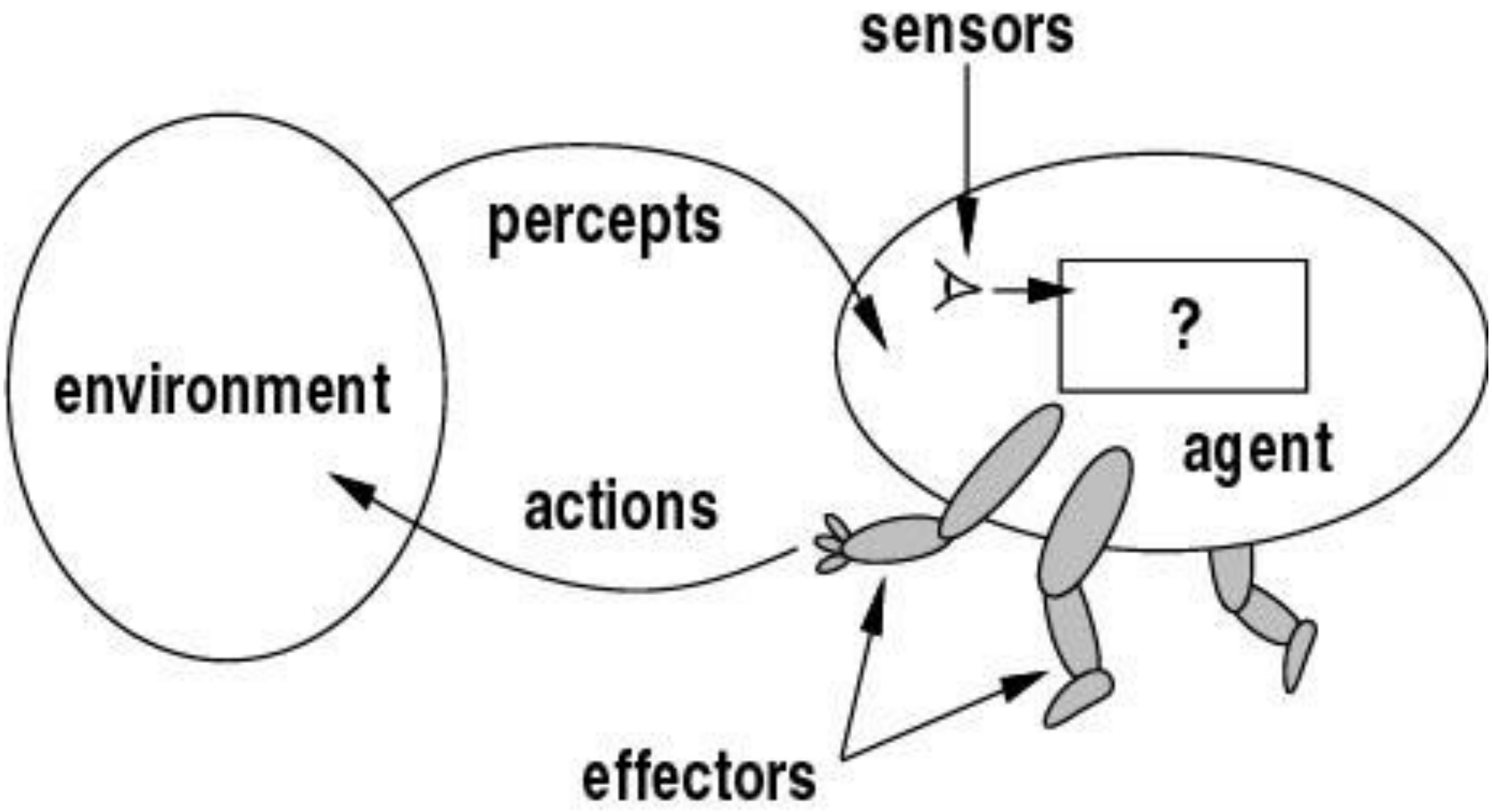
Overview (AIMA cap. 2)

- Agenti Intelligenti
- Razionalità
  - Modello PEAS
- L’ambiente di un agente:
  - Definizione
  - Proprietà e Tipi di Ambiente
- La struttura degli agenti:
  - Agenti reattivi semplici
  - Agenti basati su un modello
  - Agenti con obiettivo
  - Agenti opportunistici, cioè con funzione di utilità
  - Agenti che apprendono

Agenti intelligenti

- Intelligenza come capacità diverse
- L’approccio “moderno” all’IA: costruzione di agenti intelligenti
- La visione ad agenti ci offre un quadro di riferimento e una prospettiva diversa all’analisi dei sistemi

Agenti intelligenti: la prospettiva di AIMA



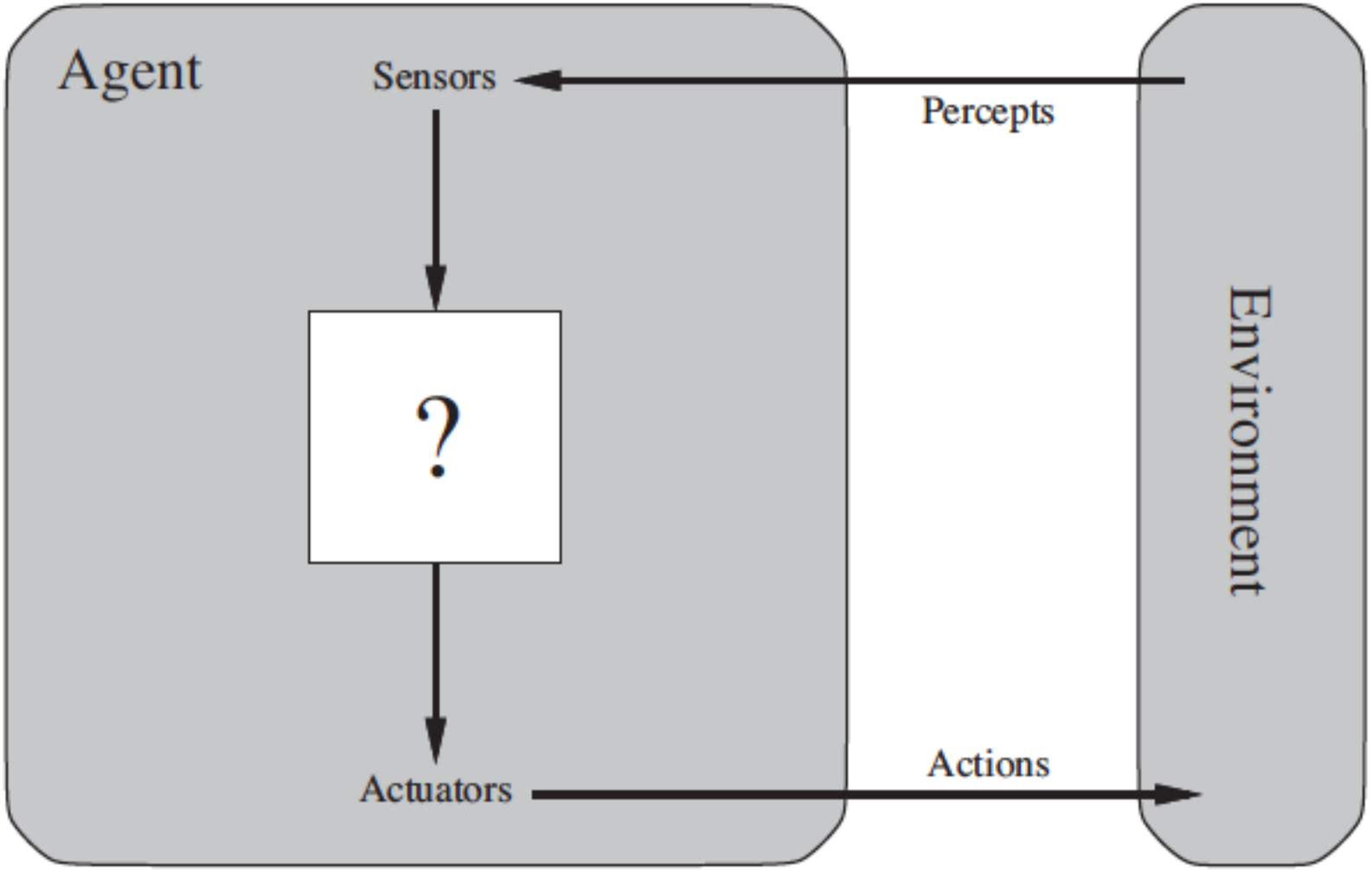
Agenti Intelligenti: la visione “moderna” (dal 1995)

- Gli agenti sono situati
  - ricevono percezioni da un ambiente
  - agiscono sull’ambiente mediante azioni
- Gli agenti hanno capacità di interazione sociale
  - sono capaci di comunicare
  - sono capaci di collaborare
  - sono capaci di difendersi da altri agenti
- Gli agenti hanno credenze, obiettivi, intenzioni ...
- Gli agenti hanno un corpo e provano emozioni

Percezioni e azioni

- Percezione: input da sensori
- Sequenza percettiva: storia completa delle percezioni
- La scelta dell’azione è funzione unicamente della sequenza percettiva
- Funzione agente: definisce l’azione da compiere per ogni seguenza percettiva.
- Implementata da un programma agente (o agente software)

Agente e ambiente



Agenti razionali

- Un agente razionale interagisce con il suo ambiente in maniera “efficace”
  - fa la cosa giusta;
  - la sequenza di stati ha un qualche aspetto auspicabile
  - Mostra una preferenza selettiva verso certi comportamenti
- Serve un criterio di valutazione oggettivo dell’effetto delle azioni dell’agente (della sequenza di stati dell’ambiente)

Valutazione della prestazione

- Misura di prestazione
  - Esterna (come vogliamo che il mondo evolva?)
  - Scelta dal progettista a seconda del problema considerando una evoluzione desiderabile del mondo
  - Valutazione su ambienti diversi

- ESEMPI: classificazione del sentiment in TWITTER
  - Qualità generale:
    - Accuracy (percentuale di twitter class ettamente)
    - Error Rate (complementare all'Accura ntuale di errori)
  - Misure dedicate allo studio della Accuracy attraverso le diverse classi (Positive/Negative/Neutral):
    - Recall/Precision/F-Measure

Agente razionale: definizione

- La razionalità è relativa a:
  - la (buona) misura di prestazioni
  - le conoscenze pregressa dell'ambiente
  - le percezioni presenti e passate
  - le capacità dell'agente
- Agente razionale: per ogni sequenza di percezioni compie l'azione che ***massimizza il valore atteso della misura delle prestazioni***, considerando le sue *percezioni passate* e la sua *conoscenza pregressa*.

Razionalità non onniscienza

- Non si pretendono perfezione e capacità predittive, basta massimizzare il risultato atteso
- Ma potrebbe essere necessarie azioni di acquisizione di informazioni o esplorative
- Esempio di 
  - Quanti tweet sbagliamo in genere noi “Twitter” umani?

Razionalità non onnipotenza

- Le capacità dell'agente possono essere limitate
- Esempio di 
  - Quante *parole* possiamo conoscere? E con quali relazioni ai *Topic*?

Razionalità e apprendimento

- Raramente tutta la conoscenza sull'ambiente può essere fornita "a priori".
- L'agente razionale deve essere in grado di modificare il proprio comportamento con l'esperienza (le percezioni passate).

-  Come apprendere la relazione tra parole e Topic (cioè la tematica legata ad un hashtag, e.g. *#totti*)?

Agenti autonomi

- *Agente autonomo*: un agente è autonomo nella misura in cui il suo comportamento dipende dalla sua esperienza.
- Un agente il cui comportamento fosse determinato solo dalla sua conoscenza *built-in*, sarebbe non autonomo e poco flessibile

AMBIENTI E CODIFICA PEAS

Ambienti

- Definire un problema P per un agente significa caratterizzare l'ambiente in cui l'agente opera (ambiente operativo).
- La progettazione dell'agente corrisponde ad una soluzioni per P, cioè

Agente razionale=soluzione

- La descrizione **PEAS** dei problem è sistematica e caratterizza:
  - Performance (prestazione richiesta all'agente)
  - Environment (l'ambiente dove si muove l'agente)
  - Actuators (gli attuatori, che determinano le azioni possibili)
  - Sensors (i sensori che consentonola poercezione)

Analisi: chatGPT come agente PEAS

Prestazione	Ambiente	Attuatori	Sensori
Rispondere in modo naturale ed efficace (preciso ma anche veloce) alle domande dell'interlocutore	La descrizione della domanda in input,	Emissione di parole in sequenza, in corrispondenza di uno stimolo detto prompt	Il *prompt* testuale osservabile in input

Analisi: chatGPT come agente PEAS

Prestazione	Ambiente	Attuatori	Sensori
Rispondere in modo naturale ed efficace (preciso ma anche veloce) alle domande dell'interlocutore	La descrizione della domanda in input, **La lingua in cui tali interazioni avvengono**	Emissione di parole in sequenza, in corrispondenza di uno stimolo detto prompt	Il *prompt* testuale osservabile in input **Tutta la produzione linguistica che l'agente può osservare prima di iniziare a dialogare**

Agente guidatore di taxi

Prestazione	Ambiente	Attuatori	Sensori
Arrivare alla destinazione, …	Strada, altri veicoli, pedoni, clienti	Sterzo, acceleratore, freni, frecce, clacson, schermo di interfaccia o sintesi vocale	Telecamere, sensori a infrarossi e sonar, tachimetro, GPS, contachilometri, acelerometro, sensori sullo stato del motore, tastiera o microfono

Agente guidatore di taxi

Prestazione	Ambiente	Attuatori	Sensori
Arrivare alla destinazione, sicuro, veloce, ligio alla legge, viaggio confortevole, minimo consumo di benzina, profitti massimi	Strada, altri veicoli, pedoni, clienti	Sterzo, acceleratore, freni, frecce, clacson, schermo di interfaccia o sintesi vocale	Telecamere, sensori a infrarossi e sonar, tachimetro, GPS, contachilometri, acelerometro, sensori sullo stato del motore, tastiera o microfono

Formulazione PEAS dei problemi

Problema	P	E	A	S
Diagnosi medica	Diagnosi corretta, cura del paziente	Pazienti, ospedale	Domande, suggerimenti test, diagnosi	Sintomi, Test clinici, risposte paziente
Robot "selezionatore"	% delle parti correttamente classificate	Nastro trasportatore	Raccogliere le parti e metterle nei cestini	Immagini (pixel di varia intensità)
Giocatore di calcio	Fare più goal dell'avversario	Altri giocatori, campo di calcio, porte	Dare calci al pallone, correre	Locazione pallone altri giocatori, porte
Bibliotecario				
Information broker (Web search engine)	Suggerimenti, utilità, rilevanza, tempo di risposta, completamento interr.	Web ed i suoi documenti, utente, (ambiente circ.)	Accedere a rete, quindi ai documenti, alle query, comunicare risposta	Accedere alla rete, "lettura" dei documenti, "lettura" della query, localizz. dell'utente
Google Assistant / Alexa				

Proprietà dell’ambiente-problema

- Completamente/parzialmente osservabile
- Agente singolo/multi-agente
- Deterministico/stocastico/non deterministico
- Episodico/sequenziale
- Statico/dinamico
- Discreto/continuo

Osservabilità

- Ambiente completamente osservabile
  - L'apparato percettivo è in grado di dare una conoscenza completa dell'ambiente o almeno tutto quello che serve a decidere l'azione
  - Non c'è bisogno di mantenere uno stato del mondo
- Ambiente parzialmente osservabile
  - Sono presenti limiti o inaccuratezze dell'apparato sensoriale.

Osservabilità dell'Ambiente: esempi

- Completamente osservabile
  - Scacchi
  - Go
- Parzialmente osservabile
  - Interpretazione linguistica
  - Guida Autonoma
  - Web Search

Ambiente singolo/multiagente

- Distinzione agente/non agente
  - Il mondo può anche cambiare per eventi, non necessariamente per azioni di agenti.
- Ambiente multi-agente competitivo
  - Comportamento randomizzato
- Ambiente multi-agente cooperativo
  - Comunicazione

Predicibilità

- Deterministico
  - Se lo stato successivo è completamente determinato dallo stato corrente e dall'azione.
    Esempio: scacchi
- Stocastico
  - Esistono elementi di incertezza con associata probabilità.
    Esempi: guida, tiro in porta
- Non deterministico
  - Se gli stati possibili non corrispondono ad una specifica distribuzione di probabiltà, ad es. sono equiprobabili
    Esempio:

Determinismo

- Agenti Deterministici
  - I cambiamenti dell'ambiente dipendono solo dallo stato dell'ambiente in un certo istante e dalla azione dell'agente
- Agenti non deterministici
  - Tali cambiamenti non seguono questa legge in modo rigido, ma sono diversi gli stati che costituiscono l'esito di una azione
- Il ruolo della probabilità:
  - Agenti stocastici in genere sono legati a forme di non determinismo descritte da distribuzioni di probabiltà in insiemi di stati possibili

Episodico/sequenziale

- Episodico
  - L’esperienza dell’agente è divisa in episodi atomici indipendenti.
  - In ambienti episodici non c’è bisogno di pianificare.
- Sequenziale
  - Ogni decisione influenza le successive

Statico/dinamico

- Statico
  - il mondo non cambia mentre l' agente decide l'azione
- Dinamico
  - I cambiamenti del mondo possono avvenire durante la decisone dell'agente
  - *Ritardare una decisione* equivale a *non agire*
- Semi-dinamico
  - L'ambiente non cambia ma la valutazione dell'agente sì.
  - Esempio: Scacchi con timer.

Discreto/continuo

- Possono assumere valori discreti o *continui*
  - lo stato: solo un numero finito di stati
  - *il tempo*
  - le percezioni, ad es. *gradi di illuminazione*
  - le azioni, ad es. le azioni possibili con *parametri*

- La guida del taxi è un problema i cui stati e tempi variano nel continuo

Noto/ignoto

- Distinzione riferita allo stato di conoscenza dell'agente
- L'agente conosce l'ambiente oppure deve compiere azioni esplorative?
- OSS. *Noto* è diverso da *osservabile*

- Gli ambienti reali son in genere: parzialmente osservabili, stocastici, sequenziali, dinamici, continui, multi-agente, ignoti

Tipi di Agenti e Ambienti

Task Environment	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete
Chess with a clock	Fully	Multi	Deterministic	Sequential	Semi	Discrete
Poker	Partially	Multi	Stochastic	Sequential	Static	Discrete
Backgammon	Fully	Multi	Stochastic	Sequential	Static	Discrete
Taxi driving	Partially	Multi	Stochastic	Sequential	Dynamic	Continuous
Medical diagnosis	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Image analysis	Fully	Single	Deterministic	Episodic	Semi	Continuous
Part-picking robot	Partially	Single	Stochastic	Episodic	Dynamic	Continuous
Refinery controller	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Interactive English tutor	Partially	Multi	Stochastic	Sequential	Dynamic	Discrete

Figure 2.6 Examples of task environments and their characteristics.

Tipologie di ambiente

	Osservabile ?	Deterministico/ stocastico	Episodico/ sequenziale	Statico/ dinamico	Discreto/ continuo	Mono/ multiagente?
Gioco 15	SI	Det	Seq	Stat	Disc	Mono
Briscola						
Scacchi						
Scacchi con timer						
Sudoku						
Guida Autonoma						
Information Broker (SE)						
Diagnostica per immagini						
Alexa						

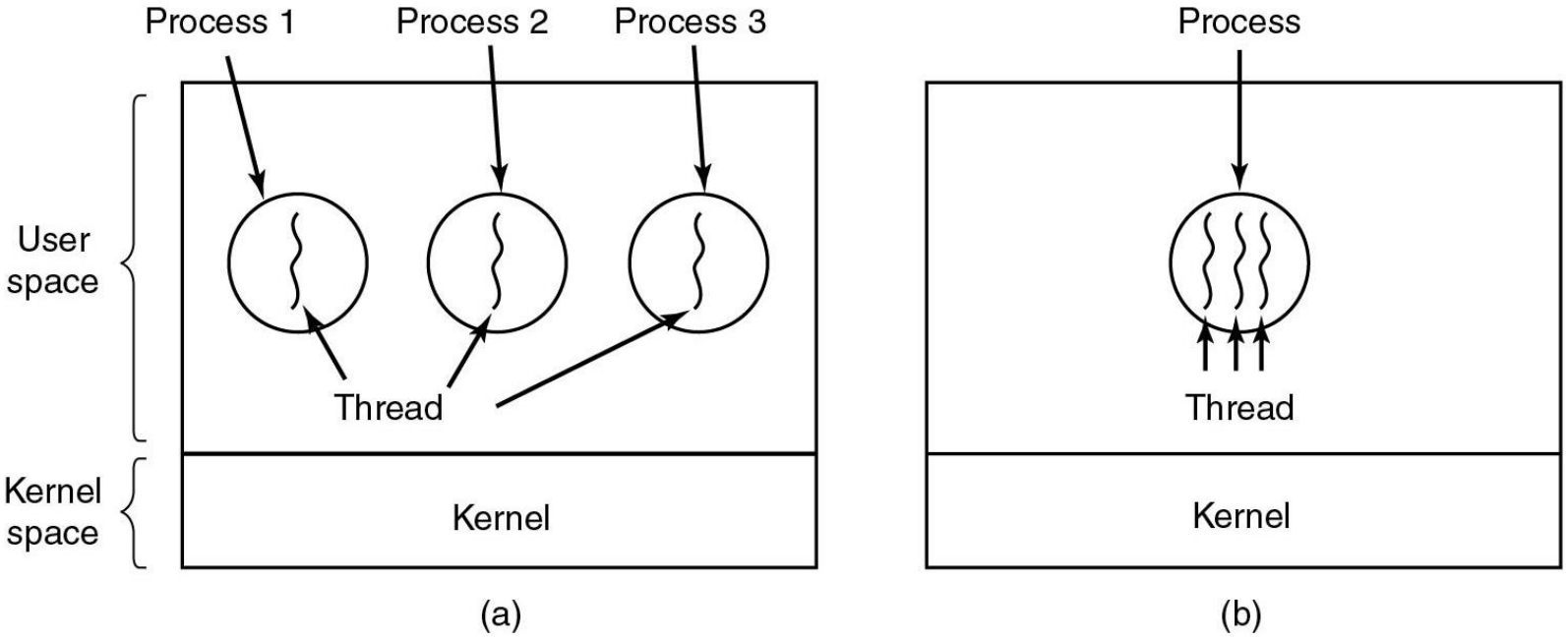
Ambiente: automazione

L’ambiente richiede la simulazione attraverso uno strumento software che si occupa di:

- generare gli stimoli per gli agenti
- raccogliere le azioni in risposta
- aggiornare il proprio stato
- [attivare altri processi implicati da tale cambiamento che influenzano a loro volta l’ambiente]
- valutare le prestazioni degli agenti

Ambiente, agente e processi

- L'ambiente ed i singoli (potenzialmente multipli) agenti sono indipendenti, analogamente a processi e threads del SO



Simulatore

```
function Run-Eval-Environment(state,
                              Update-Fn, agents,
                                Performance-Fn)
returns scores
local variables: scores   %(vector of size = #agents, all 0)
 repeat
    for each agent in agents do
       Percept[agent] ← Get-Percept(agent, state)
    end
    for each agent in agents do
       Action[agent] ← Program[agent](Percept[agent])
    end
    state ← Update-Fn(actions, agents, state)
    scores ← Performance-Fn(scores, agents, state)
until termination(state)
return scores
```

Simulatore - Prolog

```
start :-      init(InitState),    %inizializzazione unico agente e ambiente, …
              run_env(InitState, 0, Score).

run_env(State, Score, FinScore) :-
  get_percept(Perception), %acquisizione percezioni ag.

  selectAct(State, Perception, Action),  %selezione azione, …
  update_env(State, Action, NewState), %attuazione

  evaluatePerf(NewState, Score, CurrScore),

  check_term(NewState, CurrScore, FinScore).
                                          %check GOAL, …


check_term(State, CS, CS) :- satgoal(State).
check_term(State, CS,0) :-  unsecure(State),
                            writeln('LOST!\n Done!!'),
check_term(State, CS, FS) :-
    run_env(State, CS, FS). %next step
```

LA NOZIONE DI AGENTE

Ciclo di vita e Struttura interna

Il programma agente

Tipi di Agenti

Struttura di un agente

Agente = Architettura + Programma

$$Ag: \quad P \rightarrow Az$$

percezioni azioni

Il programma dell'agente implementa la funzione Ag

Programma agente

function SKELETON-AGENT (*percept*) **returns** action

static: *memory* %the agent's memory of the world

memory ← UPDATEMEMORY(*memory, percept*)

action ← CHOOSE-BEST-ACTION(*memory*)

memory ← UPDATEMEMORY(*memory, action*)

return *action*

Un esempio

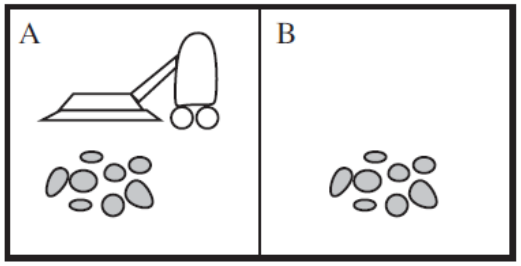


Figure 2.2 A vacuum-cleaner world with just two locations.

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

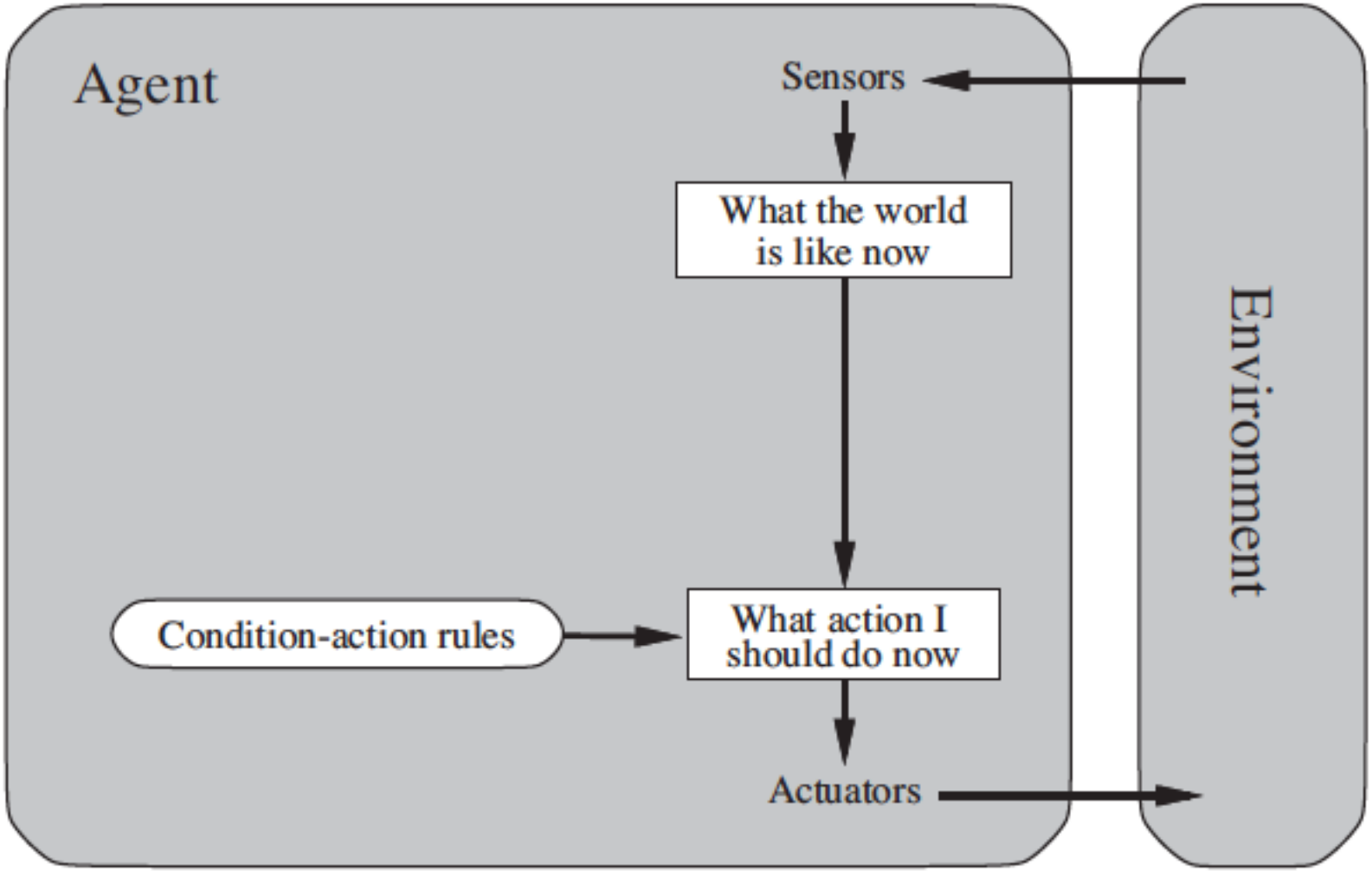
Figure 2.3 Partial tabulation of a simple agent function for the vacuum-cleaner world shown in Figure 2.2.

Agente basato su tabella

- La scelta di una azione (per volta) è un accesso a una tabella che associa un'azione ad ogni possibile sequenza di percezioni.

- Problemi:
- 1. Per giocare a scacchi tabella con righe per tutte le configurazioni individuali e tutte le sequenze!
- 2. Difficile da costruire
- 3. Nessuna autonomia
- 4. Di difficile aggiornamento, apprendimento complesso.

Agenti reattivi semplici



Agenti reattivi - programma

function AGENTE-REATTIVO-SEMPLICE (*percezione*)

returns *azione*

persistent: *regole* %un insieme di regole condizione-azione

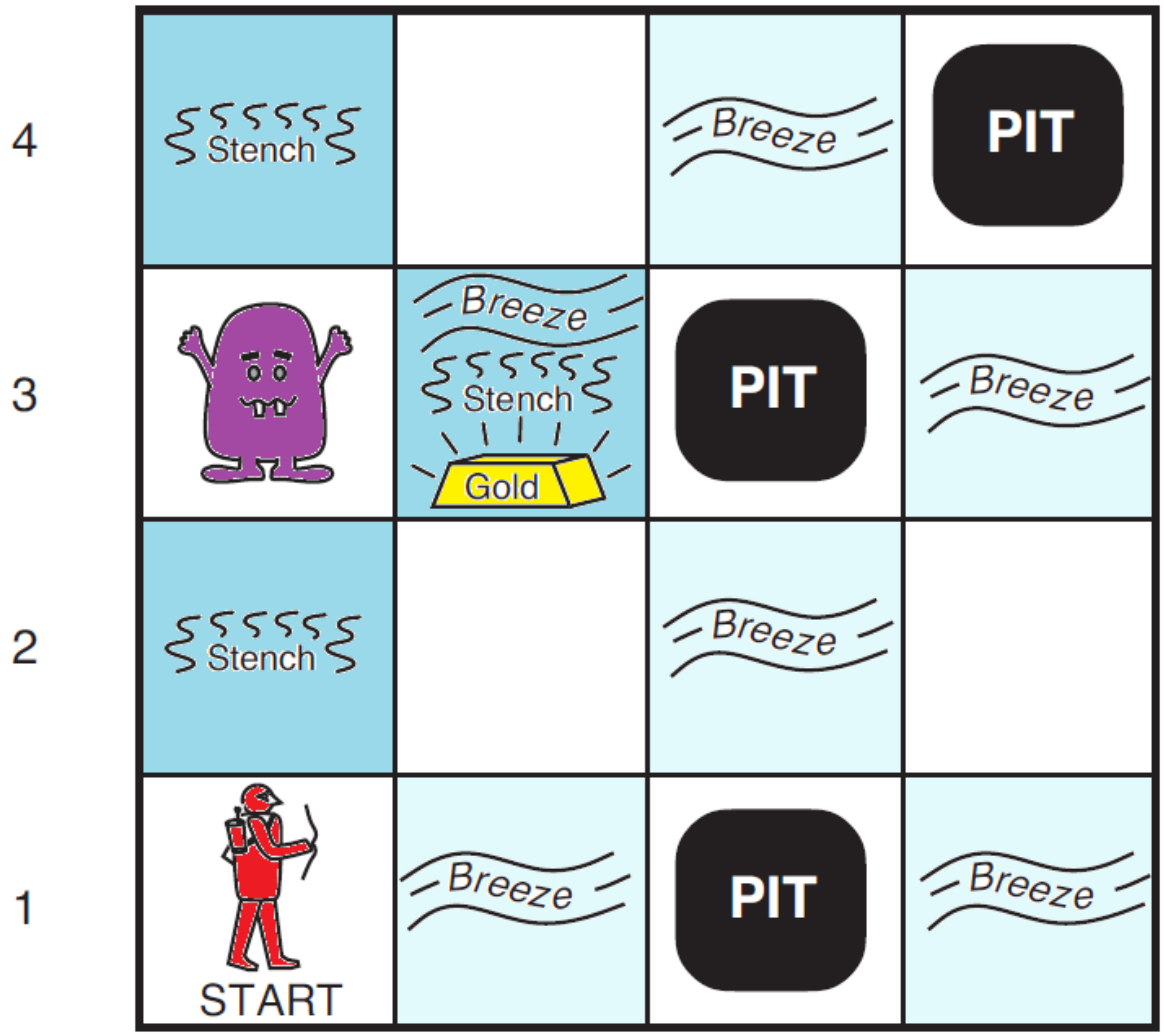
stato ← INTERPRETA-INPUT(*percezione*)

regola ← REGOLA-CORRISPONDENTE(*stato, regole*)

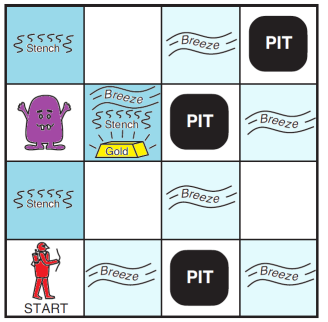
azione ← *regola*.AZIONE()

return *azione*

Un esempio … di agente con antagonista

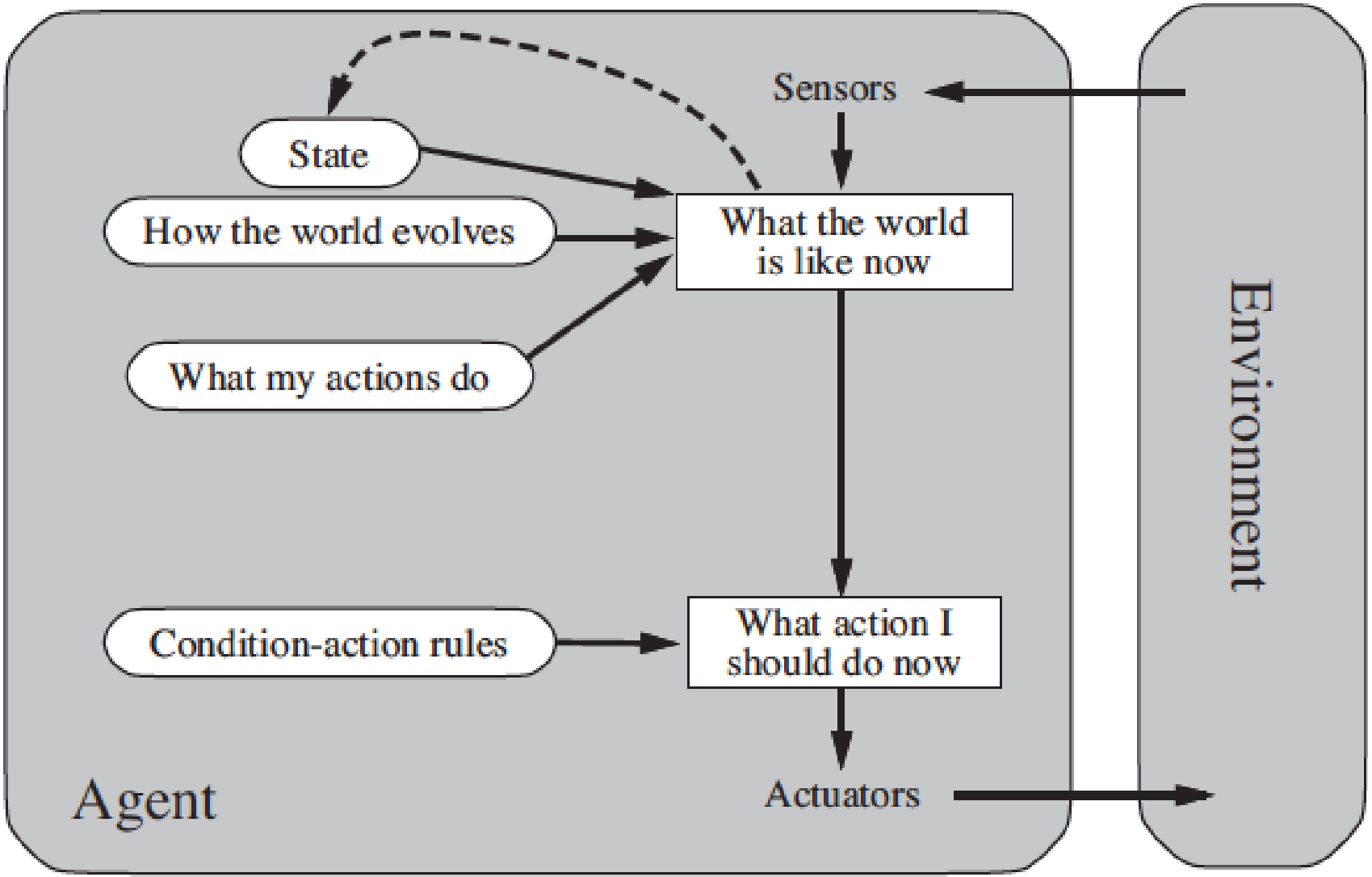


Wumpus: Simulatore Prolog




```
start :-      init,                  %inizializzazione agente, ambiente, ...
              take_steps([[1,1]], AgL, WuLoc, GoLoc).

take_steps(VisitedList, AgL, WuL, GoL) :-
  make_percept_sentence(Perception), %acquisizione percezioni

  update_KB(Perception),             %aggiornamento credenze sul mondo, …
  ask_KB(VisitedList, Action),       %selezione azione, …
  update_env(VisitedList, Action, NewVisitedList), %attuazione
  updatePos(NewAgentLocation, WumpusL, GoldL),

  check_term(NewVisitedList, NewAgentLocation, WumpusL, GoldL).
                                     %check GOAL, …

check_term(VisitedList, A_GLoc, _, A_GLoc) :- writeln('Won!').
check_term(VisitedList, AL, AL, _) :-
                    writeln('The Wumpus eats you!\n Lost!!'),

check_term(VisitedList, AgL, WuL, GoL) :-
    take_steps(VisitedList, AgL, WuL, GoL). %next step
```

Agenti basati su modello



Agenti basati su modello

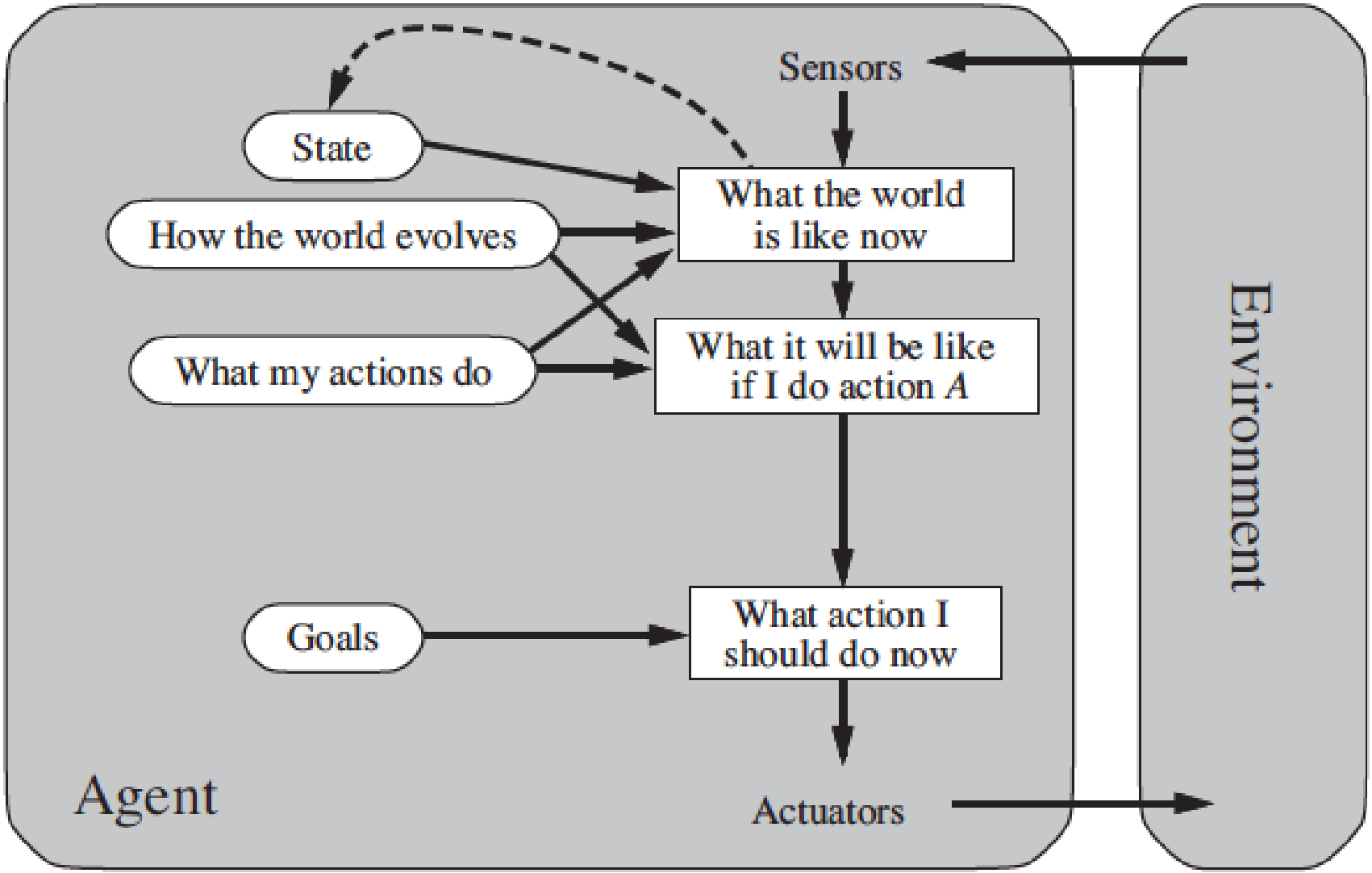
function AGENTE-BASATO-SU-MODELLO (*percezione*)
returns *azione*

persistent: *stato*, %una descrizione dello stato corrente
modello, %conoscenza del mondo
regole, %un insieme di regole condizione-azione
azione, %l'azione più recente

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

return *azione*

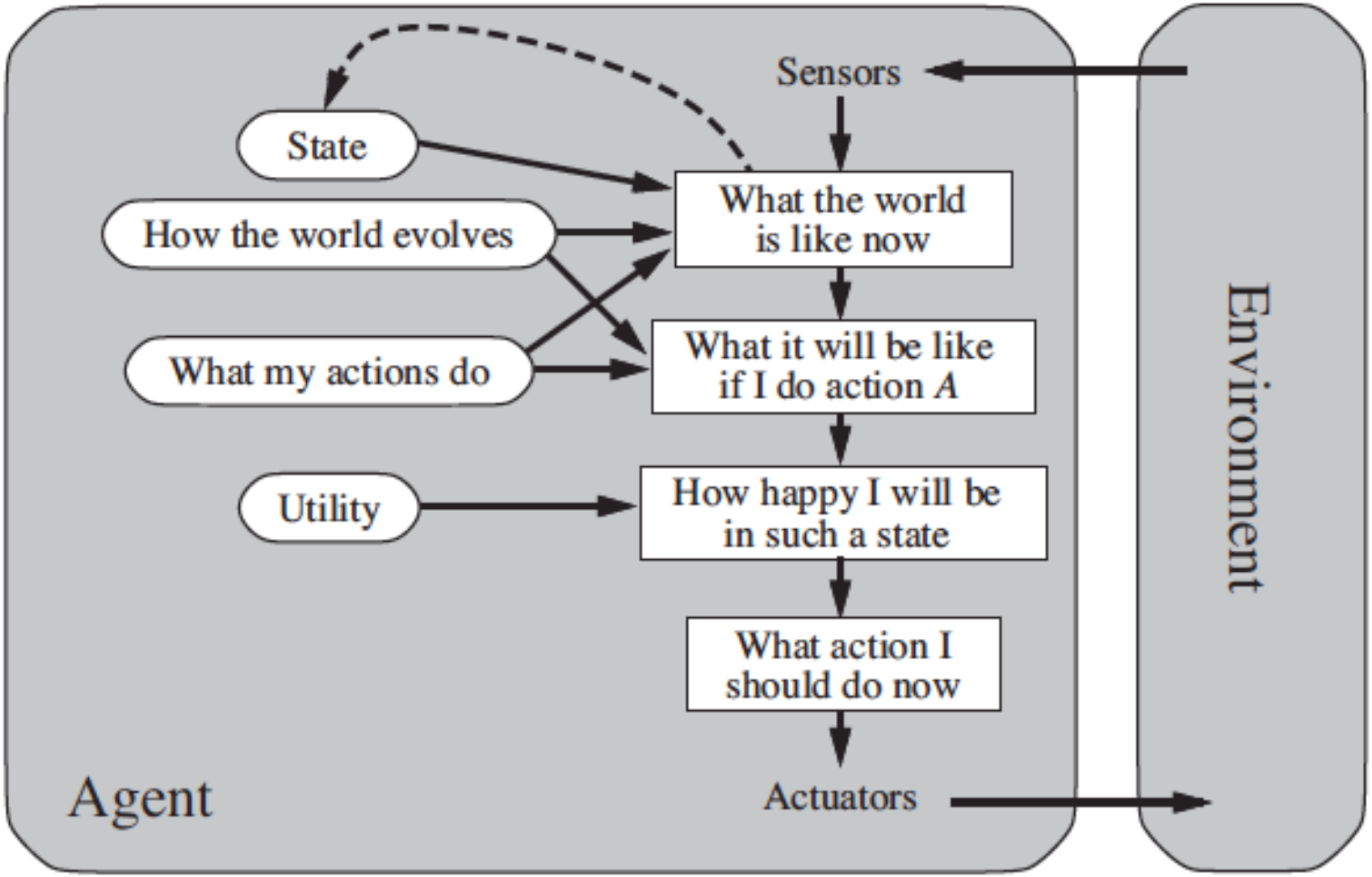
Agenti con obiettivo



Agenti con obiettivo

- Sono guidati da un obiettivo nella scelta dell'azione
  - A volte l'azione migliore dipende da qual è l'obiettivo da raggiungere (es. da che parte devo girare?).
  - Devono pianificare una sequenza di azioni per raggiungere l'obiettivo.
  - Meno efficienti ma più flessibili di un agente reattivo

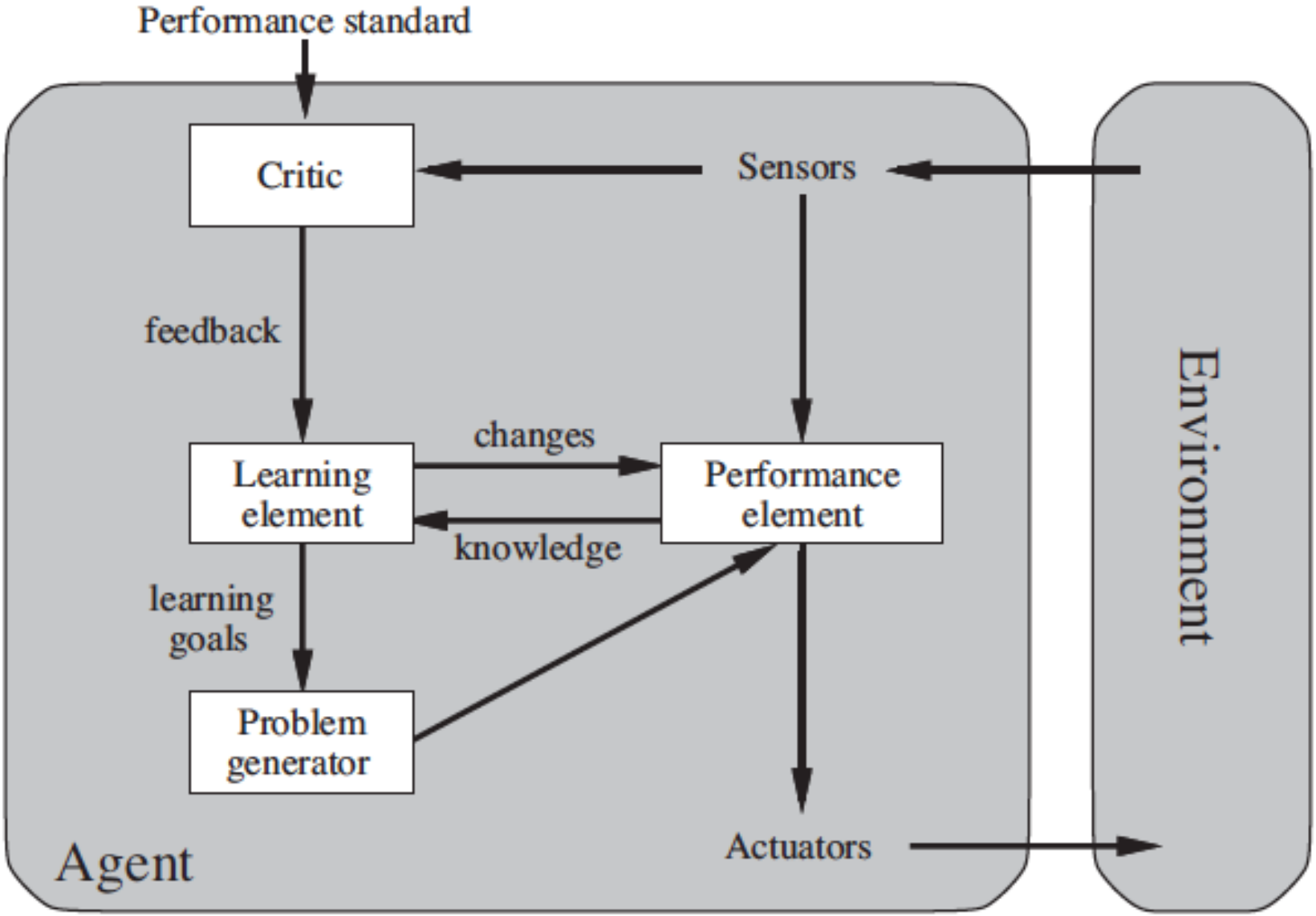
Agenti con valutazione di utilità



Agenti con valutazione di utilità

- Obiettivi alternativi
  - l’agente deve decidere verso quali di questi muoversi.
  - necessaria una funzione di utilità (che associa ad uno stato obiettivo un numero reale).
- Obiettivi più facilmente raggiungibili di altri
  - la funzione di utilità tiene conto anche della probabilità di successo: utilità attesa

Agenti che apprendono



Agenti che apprendono

- Componente di apprendimento
  - Produce cambiamenti al programma agente
- Elemento esecutivo
  - Il programma agente
- Elemento critico
  - Osserva e da feedback sul comportamento
- Generatore di problemi
  - Suggerisce nuove situazioni da esplorare

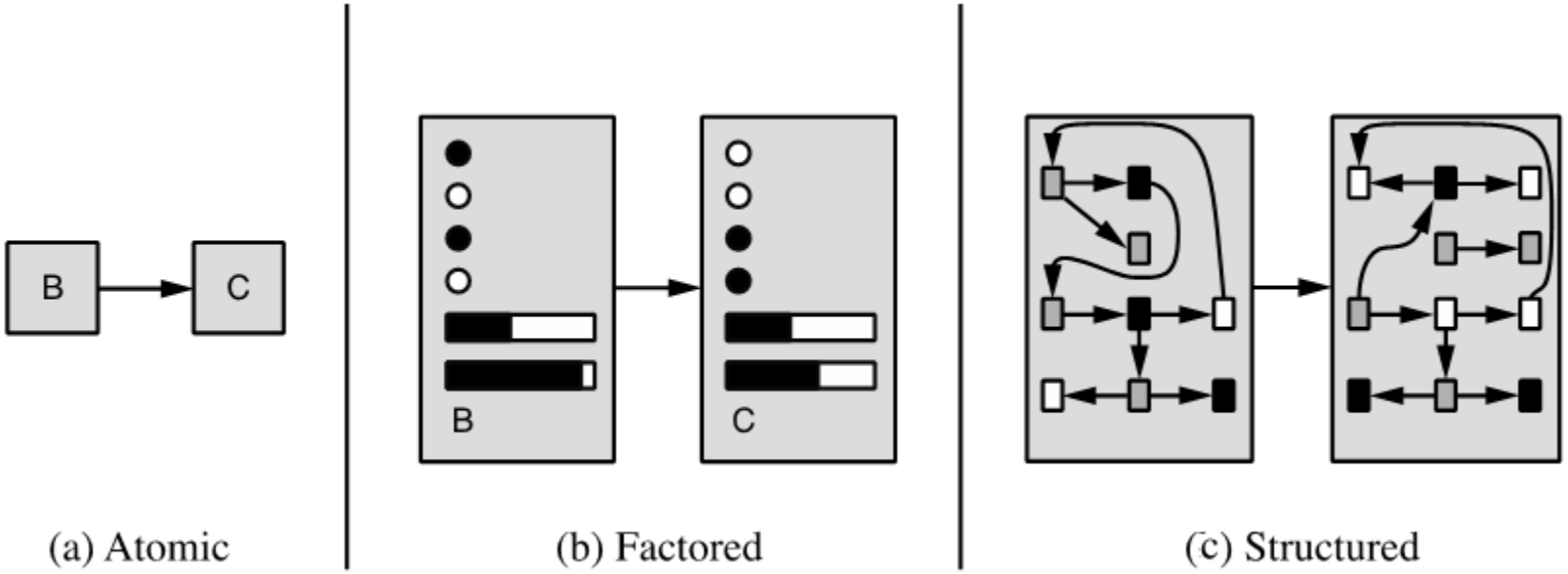
IMPLICAZIONI COMPUTAZIONALI

Rappresentazioni

Funzioni principali dell’agente

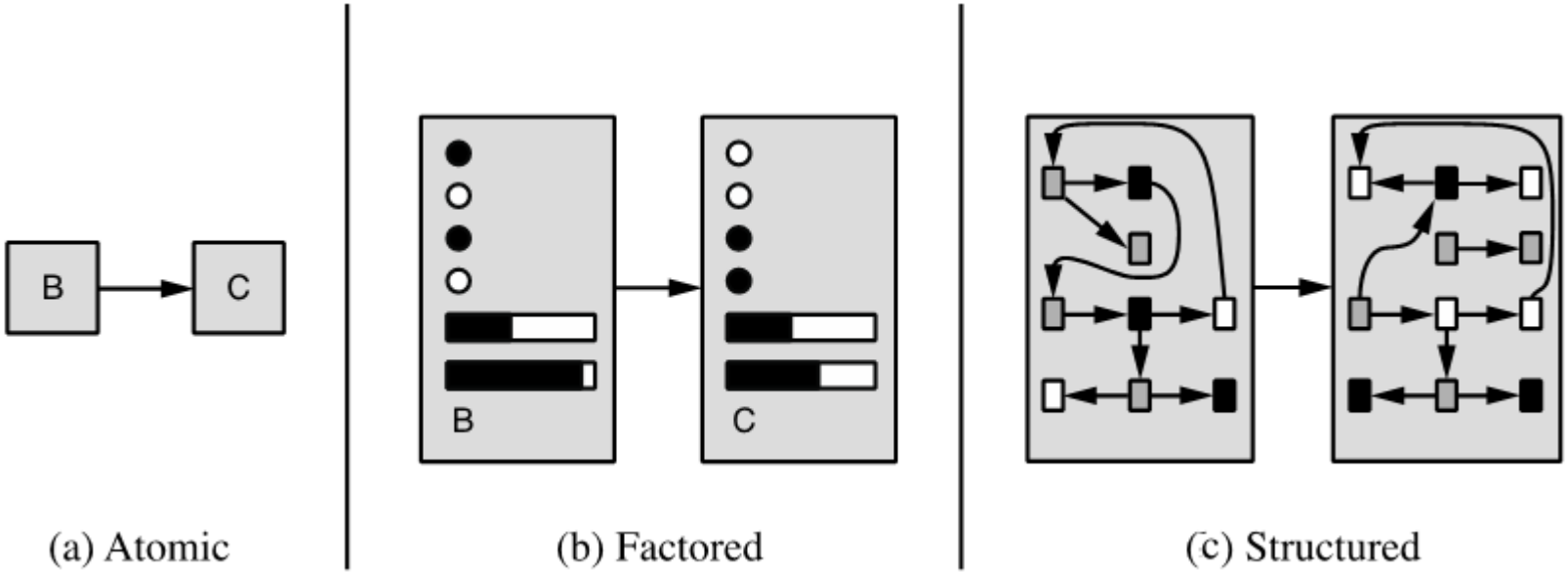
Desiderata di modelli accurati di AI

Tipi di rappresentazione



- Rappresentazione atomica
- Rappresentazione fattorizzata
- Rappresentazione strutturata

Espressività



(a) Stati finiti, Transizioni semplici, possibili estensioni probabilistiche

(b) Fattori come dimensioni in uno spazio vettoriale

(c) Rappresentazioni complesse come le relazioni della RA o le espressioni ricorsive della logica

Conclusioni

- Agenti e programmi agente
- Misure di prestazioni
- Classificazione degli ambienti operativi
- Diverse architetture di complessità crescente per i programmi agente
- Tutti gli agenti possono migliorarsi con l’apprendimento

Capacità di interagire con un ambiente?

- Robot capaci di muoversi in un ambiente, evitare ostacoli, compiere semplici missioni …
- Approccio top-down (agenti deliberativi)
- Approccio bottom-up (agenti reattivi ed evolutivi)

Capacità di emozioni?

- “The question is not whether intelligent machines can have emotions, but whether machines can be intelligent without any emotions”
- [Minsky, The Society of Mind]

Capacità di emozioni?

- Comprendere e dimostrare emozioni
  - Agenti credibili (fiducia)
  - *Affective computing* (Riconoscimento e stato emozionali)
  - Computer indossabili
- Ruolo delle emozioni nel meccanismo di decisione
  - Soggettività/sentiment
  - Emozione come target della inferenza dell'agente (social robots)
    - *Affective computing*
    - Supporto alle decisioni mediche
  - Emozione come meccanismo di *utilità/guadagno*
- Apprendimento da esempi o per rinforzo

SummarAIzing

- Perche il programma basato sulla nozione di agenti è un buon obbiettivo per il concetto generale di IA
- Cos'è un agente
- Come modelliamo l'ambiente
- Cosa sono gli stati, la conoscenza, i sensori, le regole e gli scopi di un agente
- Quali tipi di agente conosciamo:
  - Agenti basati su riflessi semplici
  - Agenti basati su Regole (statiche)
  - Agenti basati su un modello del mondo
  - Agenti basati su un modello del task (opportunistici)
  - Agenti basati su una nozione quantitative del task (utility)
  - Agenti che Apprendono

AI making

- Si spieghi **perché** la formulazione del problema deve seguire quella dell'obbiettivo

- Si determinino lo stato iniziale, il test obbiettivo, gli operatori e la funzione costo di un cammino per i segg. problemi:
  - Si deve colorare una mappa planare complessa usando solo quattro colori senza che nessuna regione confinante abbia lo stesso colore
  - Ci si è persi nella giungla dell'Amazzonia e si deve tornare al mare. C'e' un ruscello nelle vicinanze.

Ai making

- Si consideri il task di raggiungere l'uscita di un labirinto. Un robot parte dal centro del labiritno, e può muoversi a Nord, Est, Sud o Ovest. Puoi coordinare il robot a muoversi in avanti e ad una certa distanza, sebbene il robot si fermi prima di scotrarsi con un muro:
  - **Formulare il problema** e **lo spazio degli stati**.
  - Nella navigazione, i soli punti in cui intervengono delle decisioni sono le **intersezioni** tra due o più corridoi. Riformulare il problema in modo che tali decisioni corrispondano alle azioni. Come cambia lo spazio degli stati?
  - Da ogni punto del labirinto possiamo muovere il robot nelle **4 direzioni** fino a raggiungere un punto di svolta. Riformulare il problema. E' necessario ora tenere traccia dell'orientazione del robot?

- Discutere le diverse **semplificazioni e astrazioni** corrispondenti alle formulazioni ottenute