INTELLIGENZA ARTIFICIALE

`http://cialdea.dia.uniroma3.it/teaching/ai/`

$\boxed{\text{Argomenti}}$

- Elementi di Logica Formale
  - Rappresentazione della conoscenza
  - Formalizzazione del ragionamento
  - Dimostrazione automatica
- Soluzione automatica di problemi (“*search*”)
- Pianificazione automatica

$\boxed{\text{Materiale didattico}}$

- S. Russell, P. Norvig.
  Intelligenza Artificiale. Un approccio moderno.
  Volume 1.
  Pearson – Prentice Hall 2005
- M. Cialdea Mayer.
  Logica – Linguaggio, Ragionamento, Calcolo.
  Esculapio Editore 2002.

Che cos'è l'intelligenza artificiale?

Definizione: ....
Scopi: ....

Definizione prudente: l'IA studia come far compiere ai computer cose che, se eseguite da un essere umano, sarebbero considerate compiti che richiedono intelligenza.

Che cos'è l'intelligenza? Cosa significa "pensare"?

L'IA è una disciplina costituita da un'**enorme varietà di sottoaree**: ricerca automatica, rappresentazione della conoscenza, ragionamento automatico, sistemi esperti, elaborazione del linguaggio naturale, pianificazione automatica, apprendimento automatico, ...

> "I computer vengono considerati intrinsecamente stupidi, anche se sanno già fare cose un tempo appannaggio esclusivo di esseri intelligenti. Ogni obiettivo, una volta raggiunto dalle macchine (calcolare, giocare a scacchi, dimostrare teoremi), viene eliminato da quelli considerati 'intelligenti' e passa nella serie B delle cose 'automatiche'."

Una piccola storia dell'IA

Yurij Castelfranchi e Oliviero Stock: *Macchine come noi. La scommessa dell'intelligenza artificiale.* Laterza

1950: Alan Turing, *"Computing Machinery and Intelligence"*.

> I propose to consider the question: "Can machines think?".

Il "gioco dell'imitazione": una macchina capace di interagire con un essere umano senza che questi si accorga di avere a che fare con un non-umano si può considerare "intelligente".

(Il "test di Turing")

> Credo che tra circa cinquant'anni sarà possibile programmare computer con una memoria di un miliardo di bytes (1Gb) in maniera che essi giochino il gioco dell'imitazione tanto bene che una persona comune non avrà più del 70% delle probabilità di identificarli dopo 5 minuti di interrogatorio.

La sfida è aperta

Data di nascita ufficiale dell'IA: 1956

Seminario al Darmouth College (Hanover, New Hampshire): "Summer research project on Artificial Intelligence".
John McCarthy, Arthur Samuel, Claude Shannon, Marvin Minsky, ...

> ... studiare come costruire macchine che usino il linguaggio, formino astrazioni e concetti, risolvano classi di problemi ora riservate agli esseri umani, migliorino se stesse ...
> Le macchine potranno un giorno svolgere funzioni umane tradizionalmente ritenute intelligenti.

Tra i temi indicati nel documento di presentazione del seminario:

1. Migliorare la potenza dei calcolatori e raffinare i metodi di programmazione
2. Programmare un computer in modo che manipoli simboli e concetti in maniera simile a una mente umana
3. Costruire sistemi di neuroni artificiali
4. Metodi per stimare la durata di un calcolo o per trovare strategie euristiche per abbreviarle
5. Far sì che i calcolatori possano "imparare", modificando il proprio programma

Logic Theorist

A Darmouth, Cliff Shaw, Allen Newell e Herbert Simon dimostrarono il "primo programma della storia dell'intelligenza artificiale", il Logic Theorist, in grado di dimostrare i teoremi dei *Principia Mathematica* di Whitehead e Russell.

Dimostrazione automatica di teoremi, a partire da un insieme di postulati, mediante un insieme di *regole di inferenza.*

Problema: come evitare la generazione esaustiva di tutti i teoremi (il computer su cui girava Logic Theorist avrebbe lavorato per qualche centinaio di migliaia di anni prima di trovare tutte le 86 dimostrazioni dei *Principia*).

La macchina deve usare un'**euristica** (intuito?)

Newell: "un processo che *può* risolvere un certo problema, ma che non offre nessuna garanzia di riuscirci, viene detto un'euristica per quel dato problema".

Programmi "euristici": a volte sbagliando o non trovando tutte le soluzioni possibili, permettono spesso di arrivare rapidamente a una soluzione soddisfacente.

Logic Theorist arrivò a dare una dimostrazione diversa, molto più elegante, di una delle proposizioni dei Principia.

Il General Problem Solver di Newell e Simon

Scopo: risolvere una grande varietà di problemi logico-matematici, simulando i processi mentali umani

Metodo: analisi mezzo-fine.
Conoscendo lo stato iniziale e l'obiettivo, si cerca di ridurne la distanza applicando opportuni operatori.
L'obiettivo finale viene spezzato in sotto-obiettivi.
La conoscenza è rappresentata in una tabella che pone in relazione "differenze" con "operatori" (*Difference-Operator table*).

	aereo	treno	auto	bus	piedi
più di 1000 Km	X				
tra 100 e 1000 Km		X	X		
tra 1 e 100 Km			X	X	
meno di 1 Km					X

Ad ogni operatore sono associate le *precondizioni* di applicabilità, che determinano i sotto-obiettivi da raggiungere

Il GPS riuscì a dimostrare teoremi, a calcolare funzioni matematiche, a risolvere problemi logici come quello delle torri di Hanoi, dei boccali da 3 e 4 litri, quello di lupo, capra e cavolo, ecc.

Giochi

Newell e Simon nel 1957 scrivono:

> Tra dieci anni, se saranno ancora ammesse le competizioni, le macchine saranno dichiarate campioni del mondo di scacchi; entro 10 anni gli elaboratori scriveranno partite di musica dotate di innegabile valore estetico; entro 10 anni le macchine scopriranno importanti teoremi di matematica, ...

1950: Claude Shannon, *Una macchina giocatrice di scacchi*, primo lavoro sulla teoria del software scacchistico.

Giochi a due avversari: problema ben definito, sia nelle operazioni consentite che nello scopo finale, tuttavia spesso non banale.
Scacchi: impossibile la ricerca esaustiva. Uso di un'*euristica.*

1960: Arthur Samuel scrive un programma per il gioco della dama. Il programma "ricorda" la valutazione delle posizioni incontrate, ed è in grado di modificarla, "imparando" dalle conseguenze delle proprie scelte. Nel 1962 arrivò a sconfiggere un giocatore abbastanza famoso.

Con l'aumento della potenza di calcolo, i programmi per i giochi sfruttavano sempre di più la ricerca (la "forza bruta").

1997: Deep Blue sconfigge il campione del mondo di scacchi, Garry Kasparov.

Le prime delusioni

I programmi per i giochi sfruttano la potenza di calcolo più che tecniche di intelligenza artificiale. Le macchine giocano bene giochi in cui il numero di scelte possibili è finito, grande ma non immenso, e quelli “a carte scoperte”. Ma sono mediocri a bridge e poker, e sono pessimi giocatori di Go dove nessuna strategia di forza bruta basta a sostituire l’intuito.

I programmi che si occupavano di algebra, geometria, calcolo infinitesimale erano rigidi, incapaci di far fronte a variazioni minime nel problema posto.

Il GPS aveva ricevuto critiche durissime: richiede una descrizione dettagliata di ogni problema, che già contiene, in sostanza, la soluzione.

Rinuncia all'obiettivo del risolutore generale di problemi:

I sistemi esperti

Programmi stupidi in generale, ma estremamente "competenti" su determinati settori, ristretti e difficili, riservati a professionisti (medicina, chimica, fisica, biologia).

Feigenbaum: Knowledge is power

Sistemi esperti: **sistemi basati sulla conoscenza**, costituiti da una **base di conoscenza** e un **motore di inferenza**.

1967: Heuristic Dendral, il primo sistema esperto (chimica organica).
1974: Mycin, il piè celebre sistema esperto di medicina, con prestazioni comparabili con quelle di uno specialista (diagnosi nel campo delle infezioni del sangue, prescrizione di terapie, spiegazione del proprio ragionamento).
1978: Prospector (geologia)

Anni 80: grande successo pratico e commerciale.

Limiti: poco robusti e flessibili, conoscenza limitata, difficile verifica di correttezza, non imparano dall'esperienza.

Apprendimento

Il “collo di bottiglia” dell’IA.

Estrarre dagli esperti umani la conoscenza necessaria è un lavoro arduo (*Ingegnere della conoscenza*).

Apprendimento automatico della conoscenza, osservando il comportamento dell’esperto.
1986: ID3, di Quinlan: apprendimento di un “albero di decisione”, basato sugli attributi di un insieme di “dati di addestramento”.

Programmi capaci di inventare

Anni 80: Am (Automatic Mathematician), di Douglas Lenat, a partire da alcuni semplici concetti basilari in matematica (insiemi, regole logiche, ...), "scoprì" i numeri interi, l'addizione e la moltiplicazione, i numeri primi, i numeri "massimamente divisibili", ...

Ma poi si fermò.

Eurisko, sempre di Lenat, giocava a un gioco simile a Risiko, "imparò" da una serie di partite giocate contro se stesso e vinse un torneo nazionale.

Ma entrambi i programmi spesso si bloccavano. Eurisko arrivò a progettare di cancellare dalla sua memoria tutti i piani ...

Lenat: serve anche una conoscenza approfondita del mondo
Nel **1984** inizia il progetto **Cyc**, un archivio di tutta la conoscenza sul mondo, inclusa molta conoscenza di senso comune.

Comprensione del linguaggio naturale

1964-66: Eliza, scritto da Joseph Weizenbaum, è uno dei più famosi programmi di conversazione: assolutamente stupido.

Importanza dell'elaborazione automatica del linguaggio naturale: progetti di ricerca generosamente finanziati anche dai militari (tra gli anni 50 e 60, gli USA spesero 20 milioni di dollari)

Primi progetti di traduzione automatica deludenti (inglese-russo-inglese):
"lo spirito è forte ma la carne è debole" $\Longrightarrow$ ....
$\Longrightarrow$ "la vodka è alcolica ma la carne è andata a male"

Grammatica + significato delle parole + conoscenza del mondo + pragmatica

1988: Alfresco (IRST, Trento), combina linguaggio naturale e navigazione ipermediale, capisce domande anche complesse, ricorda l'andamento della conversazione e contiene un modello dell'utente.

L'approccio sub-simbolico

Dai problemi della robotica:
Rodney Brooks: "Intelligenza senza ragione", "Gli elefanti non giocano a scacchi"

Un robot intelligente non ha bisogno di rappresentazione del mondo, di euristiche, regole, sistemi logici. Non si può costruire una mente artificiale con metodo top-down, ma si deve utilizzare l'approccio bottom-up: macchine dotate di molti riflessi nervosi semplici, interconnessi in modo tanto complesso da far emergere l'intelligenza

L'intelligenza è una "proprietà emergente dei sistemi complessi"

Reti neurali, algoritmi genetici, vita artificiale

I primi esperimenti per simulare al computer reti di neuroni artificiali risalgono al 1956 (McCulloch & Pitts).

Reti neurali: strutture basate su una rete complessa di oggetti semplici (neuroni). Le unità ricevono segnali di input (se gli impulsi superano una certa soglia), calcolano una funzione, e emettono un output verso unità connesse.
Nessun simbolo o operazione esplicita.
La conoscenza emerge dall'intero sistema

Sottoaree dell'IA

- **Ricerca automatica**
  Metodi generali e uniformi per la soluzione di problemi
  Tecniche per la risoluzione di problemi che richiedono l'esplorazione di uno "spazio di stati"
  (dal GPS di Newell & Simon)
  - **Giochi** a due avversari
    Dominio limitato e sottoposto a regole precise, "ci danno la massima complessità con le più piccole strutture iniziali" (Minski). Terreno di prova.
- **Rappresentazione della conoscenza**
  Catturare in un linguaggio formale la conoscenza necessaria per il comportamento intelligente
- **Ragionamento automatico**
  - Linguaggi formali per la rappresentazione di conoscenza
  - Dimostrazione automatica di teoremi
  - Applicazioni: Progettazione e verifica di circuiti, correttezza di programmi, controllo di sistemi complessi
- **Sistemi Esperti**

- **Elaborazione del linguaggio naturale**
- **Pianificazione e robotica**
- **Apprendimento automatico**
  - approccio simbolico
  - approccio sub-simbolico (reti neurali)

Risultati importanti per l'informatica

Tecniche e metodi dell'IA si sono integrati con il software tradizionale

- time sharing (vs. programmazione batch)
- sviluppo di linguaggi di programmazione di alto livello e interattivi (LISP, Prolog)
- interfacce grafiche, a finestre
- personal workstation
- importanza della separazione tra conoscenza e meccanismo di ragionamento (tra dati e algoritmi, tra oggetti e metodi ...)
- strutturazione gerarchica della conoscenza, *reti semantiche* e *frames*
  $\Rightarrow$ paradigma di programmazione a oggetti

RAPPRESENTAZIONE DELLA CONOSCENZA
(Cap. 6 RN)

Conoscenza su:

- lo stato del mondo,
- come cambia il mondo,
- il proprio obiettivo,
- l'effetto delle proprie azioni

Capacità di ragionamento per dedurre proprietà del mondo che non si possono osservare

Per progettare un agente basato sulla conoscenza:

- un linguaggio formale per rappresentare la conoscenza
- un metodo per ragionare in tale linguaggio

COMPONENTI DI UN AGENTE BASATO SULLA CONOSCENZA:

Knowledge Base (KB) (Base di conoscenze)

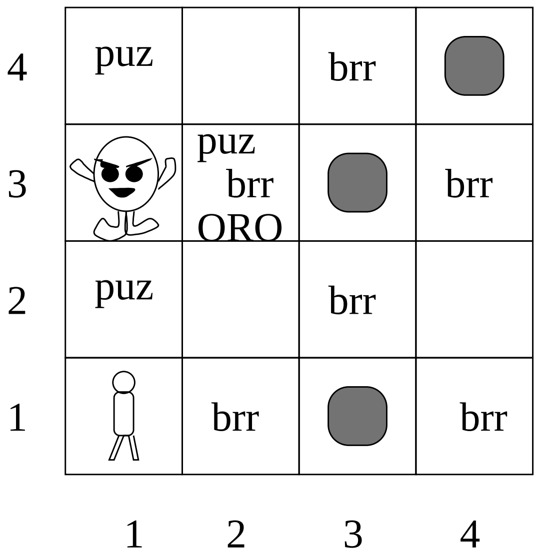
insieme di *enunciati* che rappresentano fatti del mondo, espressi in un

LINGUAGGIO PER LA RAPPRESENTAZIONE DELLA CONOSCENZA

Meccanismo di inferenza

per rispondere a domande su che cosa l'agente conosce (che cosa è derivabile dalla KB)

Un esempio: il mondo del wumpus



PERCEPTS : `stench`, `breeze`, `glitter`
L'agente non percepisce la propria posizione,
Non ha una mappa della caverna.
Se non conosce la dimensione della caverna: `bump`

`(Stench, Breeze, Glitter)`

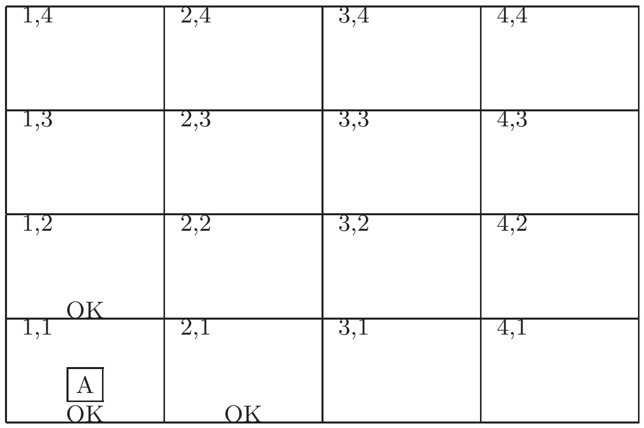
ACTIONS : `forward`, `right`, `left`, `grab`, `shoot`

GOAL : essere in (1,1) con ORO

ENVIRONMENT: caverna

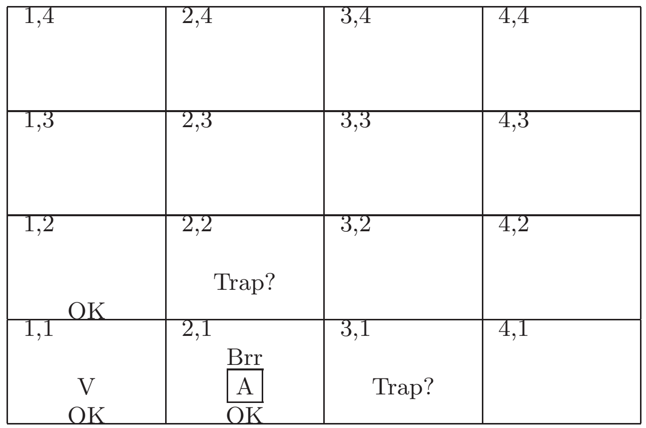
La conoscenza dell'agente nel mondo del wumpus

Tempo 0 : conoscenza iniziale (assumendo che conosca la dimensione della caverna):

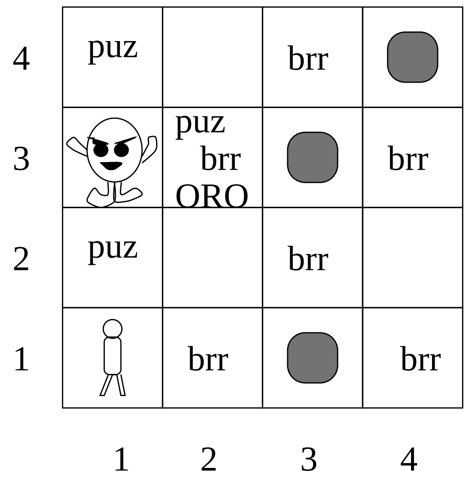


Percept: `(None, None, None)`

Tempo 2 : dopo l'esecuzione di `Right, Forward`



Percept: `(None, Breeze, None)`



Tempo 7 : dopo l'esecuzione di `Left, Left, Forward, Right, Forward`

1,4	2,4	3,4	4,4
1,3 Wumpus	2,3	3,3	4,3
1,2 Puzz $\boxed{A}$ OK	2,2 OK	3,2	4,2
1,1 V OK	2,1 Brr V OK	3,1 Trap	4,1

Percept: `(Stench,None, None)`

Tempo 11 : dopo l'esecuzione di `Right, Forward, Left, Forward`

1,4	2,4	3,4	4,4
1,3 Wumpus	2,3 Puzz,Brr Oro $\boxed{A}$	3,3	4,3
1,2 Puzz V OK	2,2 V OK	3,2	4,2
1,1 V OK	2,1 Brr V OK	3,1 Trap	4,1

Percept: `(Stench,Breeze,Glitter)`