Lecture 1 – Maggio 10

- **Perché sistemi cognitivi**

vogliamo dai sistemi intelligenti di operare in ambienti non controllati ma ambienti complessi cioè ambienti in cui un sistema non ha un'informazione completa e c'è dell'incertezza sul tipo di informazione che gestisce. tutti gli ambienti che si chiamano *real life*. quindi un sistema cognitivo è un sistema che deve essere in grado di operare in ambienti complessi non in ambienti controllati. sistemi cognitivi artificiali operiamo in ambienti complessi quindi non predeterminati non chiusi. Esempio degli ambienti sono

1- **Controllato**

Sappiamo cosa aspettarci e possiamo programmare il sistema per fare ciò che vogliamo. Come robotica industriale. quindi non abbiamo dei sistemi cognitivi perché tutto è noto abbiamo, cioè un tipo di ambiente che va definire in modo preciso quello che è lo spazio possibile delle operazioni di un'agente.

2- **Complesso**

Non sappiamo cosa aspettarci e il sistema deve essere flessibile e adattabile". per esempio, talmente complesso è quello rappresentato da un robot che deve andare a stirare per esempio dei vestiti.

- **2 modi per replicare una facoltà cognitiva**

per avere la capacità di replicare un sistema cognitivo qualsiasi ci sono sostanzialmente due strade:

1- Inventare una nuova soluzione computazionale

Replicare quel tipo di facoltà indipendentemente dal modo in cui lo stesso sistema naturale svolge quel tipo di problema.

2- Prendere ispirazione dalla natura (cognitiva or bio-inspried systems)

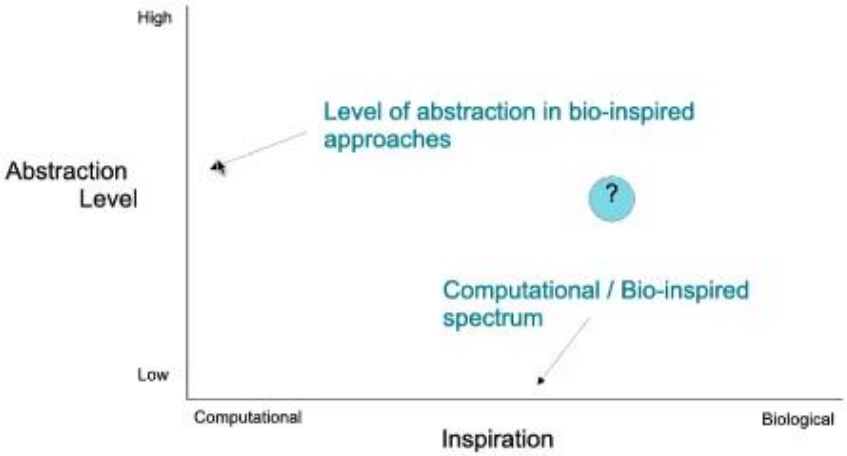
sostanzialmente quello di avere dei sistemi biologicamente o cognitivamente ispirati.

Possiamo pensare di usare uno spazio bidimensionale per posizionare il sistema in base al grado cognitiva e in base al paradigma computazionale utilizzato.

- **Lo spazio 2D della modellazione cognitiva**

dove possiamo inserire su un asse il livello di ispirazione che può essere computazionale Vs biologica naturale. Possiamo identificare quelli che sono i paradigmi di modellazione che andiamo a utilizzare che possono essere paradigmi più impliciti/astratti o meno astratti/impliciti cioè nello specifico i paradigmi simbolici sono paradigmi che si caratterizzano per avere un altro livello. perché ho le regole esplicite che queste regole esplicite possono essere a mano o a prese.

Mentre invece se ci muoviamo lungo una asse che ha un basso livello di astrazione al livello di paradigma di modellazione abbiamo di solito i sistemi connessioni che sono il tipo di sistemi che si colgono più in basso all'interno dell'asse della y perché hanno un basso livello di astrazione cioè io là dentro non riesco a capire bene chi e dove.



- **Inspration e Explentaion in Artificial Congnitive System**

l'idea è quella di utilizzare la simulazione di questi sistemi per cercare di scoprire delle cose nuove diciamo dei meccanismi ignoti rispetto a come funziona il nostro cervello oppure come funziona la nostra mente.

Prendiamo da un sistema naturale come ponte di **ispirazione** e cerchiamo di replicarne alcune funzioni e meccanismi, vengono dalla fine vengono dalle neuroscienze e cose di questo genere d'accordo quindi prendiamo ciò che è noto all'interno di una disciplina rispetto a come funzionano alcuni processi di apprendimento di ragionamento di compressione del linguaggio naturale e etc, e cerchiamo di replicarli dal punto di vista algoritmico creando dei programmi in cui siano strutture dati e in termini di algoritmi che andiamo ad utilizzare ci sia un match tra ciò che succede all'interno del sistema naturale e ciò che noi facciamo accadere all'interno del sistema artificiale.

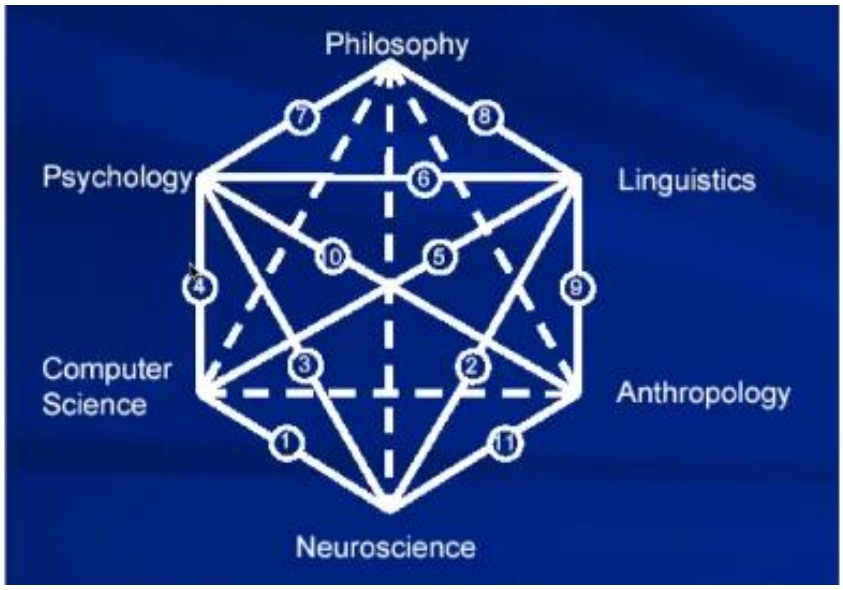
abbiamo anche obiettivo di tipo **esplicazione** significa che noi vogliamo utilizzare questo nostro sistema che andiamo a realizzare per capire delle cose nuove rispetto alle teorie iniziali che abbiamo implementato. cioè vogliamo fare in modo che tramite la simulazione che andiamo a generare, riusciamo a generare nuova conoscenza rispetto al sistema naturale che abbiamo preso inizialmente come ponte di ispirazione.

Questo loop tra ispirazione e esplicazione è uno dei contributi di Scienza Cognitiva. Cioè idea della scienza cognitiva è questo loop tra sistemi biologici e le macchine analoghe.

- **La caratterizzazione principale di sistema cognitivo**

La caratterizzazione principale di questo tipo di approccio è appunto di quello che ci sono due discipline sorelle nella storia dell'intelligenza artificiale che sono AI Classica e le scienze cognitive che sono definite due diverse scienze dell'artificiale. Scienze cognitive va a elaborare con diverse tipe delle scienze come psicologia.

quando si parla di Scienze cognitive si parla di queste discipline che viene detto esagono cognitivo cioè le scienze cognitive va ad identificare la commistione di diverse aree della conoscenza, filosofia, linguistica, antropologia, neuroscienze computer Science in uno specifico intelligenza artificiale e psicologia.



- **Natural/Cognitive inspritation And AI**

nella prospettiva di intelligenza artificiale storica l'idea di prendere spunto dal modo in cui funzionano sistemi biologici era assolutamente naturale cioè l'idea per fare un sistema intelligente artificiale io prendo spunto da quelli che sono dei sistemi naturali che fanno le stesse cose. Per fare un sistema intelligente a partire dalla metà degli anni 80 c'è stato

un cambiamento di paradigma legato a diversi motivi commerciali perché l'idea è stata quella di andare a focalizzarsi sul replicare intere funzioni cognitive. è difficile andare a focalizzarsi su task compiti ristretto, cioè io non voglio capire come funziona la visione negli esseri umani e andare a replicarla all'interno di una macchina, io voglio soltanto fare un sistema che fa riconoscimento di faccia quindi solo un Task che viene fatto in modo ottimali.

parliamo ancora di sistemi cognitivi artificiali perché ci sono una serie enorme di cose che sistemi di intelligenza artificiale non sono in grado di fare, significa:

1- svolgono questi compiti peggio
2- quando analizziamo il gap tra intelligenze naturali e intelligenze artificiali non in singoli compiti ma a livello generale vediamo che la situazione vuol dire questo capo esplode; quindi, c'è una distanza ancora enorme tra quello che un animale può fare in un ambiente naturale e che intelligenza arifticiale non è in grado di farlo.

Ci sono due prospettive storiche:

- **Cognitivismo (Approccio Simbolico)**

l'idea dell'approccio simbolico è che l'intelligenza sia negli esseri umani che nelle macchine è la capacità di andare ad elaborare dei simboli delle strutture simboliche.

1- Focus su funzioni esecutivi di alto livello (ragionamento, pianificazione, linguaggio)
2- rappresentazioni sono strutturale (Simboliche) (physical symbol).
3- Ha la prospettiva architettura (relativa all'integrazione di diverse funzioni) cioè si cerca di vedere come andare ad integrare diverse funzioni cognitive non mi interessa solo il linguaggio mi interessa integrare il linguaggio e visione.
4- Inspirazione cognitiva (hieristic-driven approach)

Nouvelle AI

1. Ha focus su precisione,
2. rappresentazione distribuite, e la prospettiva è sul singolo sistema
3. ispirazione sono biologici.

Uno degli errori comuni che si fa è quello di pensare che per esempio approcci neurali siano quelli più cognitivamente o biologicamente ispirati, questo è falso. cioè scegliere un paradigma piuttosto che un altro significa andare a modellare un problema a un livello di astrazione piuttosto che un altro. però per rendere cognitivamente o biologicamente valido un programma che noi andiamo a costruire, l'elemento importante che dobbiamo

andare a considerare sono i vincoli aggiuntivi che vengono dalla biologia oppure dalla Cognitivo che noi andiamo a sovra imporre su questi tipi di programma.

Reti semantiche

Strumento di intelligenza artificiale cognitivamente ispirato è stata la realizzazione delle reti semantiche che sono una dei principali sistemi simbolici che si utilizzano ancora oggi. sono nate da un'idea di andare a modellare in modo psicologicamente il funzionamento della memoria umana quindi l'idea di dare una struttura a grafo a quella che è la nostra conoscenza che ora per noi è un qualcosa di acquisito.

questa idea nasce da un approccio di tipo cognitivo dove i nodi di questo grafo rappresentano i concetti dell'attivazione di questi nodi funziona in base al meccanismo di spreading activation che viene utilizzato anche nei sistemi connessisi. Spreading activation è alla base dei sistemi di connessione mistica quindi per esempio l'introduzione delle reti semantiche va a mischiare alcuni elementi che poi sono diventati elementi cardine di programmi di architettura connessioni con elementi cardine di aspetti tipo simbolico.

un altro esempio di AI Simboico è un sistema che si chiama General problem solver che utilizza quella cosiddetta euristica mezzi fini cioè Means End Analysis. Era in grado di risolvere delle semplici dimostrazioni logiche. Era basato sull'analisi dei protocolli verbali cioè si è visto gli esseri umani come matematici, logici ed è stato chiesto a queste persone di ragionare ad alta voce questo è il protocollo verbale, ragionare ad alta voce e spiegare quali erano i meccanismi che loro mettevano in atto per dimostrare un teorema.

l'idea è che Esiste uno stato finale (Goal) da raggiungere c'è uno stato di partenza all'interno del dominio noi ci troviamo e poi ci sono una serie di operatori che in qualche modo devono essere selezionati per ridurre la distanza simbolica tra lo stato di partenza e lo stato goal che è quello a cui vogliamo arrivare.

- **Frame (Minsky)**

Sono stati primi strumenti della rappresentazione della conoscenza e del senso comune che ci permette di rappresentare del Senso comune e il ragionameto della deduzione classica.

Frame 1	
Concept 1	
Attribute 1	Value 1
Attribute 2	Value 2
Attribute 3	Value 3
...	...

- **Scripts (Shank And Abelson)**

Strutture dati per rappresentare conoscenza di senso comune legata agli eventi (e.g. es. “andare fuori a cena") e usata in natural language processing per permettere ad agenti intelligenti di rispondere a domande riguardanti semplici storie.

Script Restaurant: e.g. entrare al ristorante, chiedere un tavolo, sedersi, consultare il menu, mangiare, pagare etc. Attraverso questo sttruttura dato possiamo rispondere alle domande banali come:

Esempio “Mary went to a restaurant, ordered a salmon. When she was paying she noticed that she was late for the next appointment”,

Domanda: *“Did Mary eat dinner last night?”.*

Si, ha mangiato perché ha consumato quello che avevo ordinato prima però nella frase non c'è scritto da nessuna parte che Maria ha effettivamente cenato quindi questa è una capacità abduttiva, è un saltare alle conclusioni che noi facciamo in automatico. però per un sistema intelligenza artificiale rispondere a questo tipo di domande è molto difficile. i primi sistemi in grado di affrontare questo tipo di problema erano sistemi che utilizzano gli script in cui ci sono tutte le informazioni tipico di senso comune legate appunto alla scena.

- **Architetture cognitive (Newell)**

Un'architettura cognitiva implementa la struttura invariante del sistema cognitivo, Idea è catturare gli elementi comuni sottostanti a diversi tipi di agenti intelligenti e fornire un framework in cui si possa modellare tale comportamento intelligente. Il primo lavoro nacque negli anni‘80s con la prima versione dell’architettura cognitiva SOAR (Newell, Laird and Rosenbloom, 1982).

Quando un sistema biologicamente/cognitivamente ispirato ha un potere esplicativo rispetto al sistema naturale preso come fonte di inspirazione? Per rispondere a questo tipo di domanda abbiamo bisogno di introdurre la differenza tra due diversi paradigmi di progettazioni

Paradigmi di progettazione

1- *Approccio Funzionalista*

Un termine che è preso a prestito dalla filosofia della mente vuole dire che un sistema artificiale per poter avere un potere esplicativo rispetto al sistema naturale preso come fonte di ispirazione, era sufficiente avere una sorta di equivalenza debole tra le procedure che erano implementate all'interno di questo sistema e corrispondenti processi cognitivi che erano noti all'interno di un trend di un sistema naturale preso come fonte di ispirazione. per dire io posso utilizzare il potere esplicativo del mio artefatto basta una equivalenza funzionale cioè il mio sistema artificiale deve funzionare COME, dove significa avere una equivalenza superficiale in termini di input e output.

se il mio sistema è in grado di prendere gli stessi input del sistema naturale e di produrre gli stessi Output del sistema artificiale allora io posso utilizzare la simulazione del mio sistema per fare un'inferenza inversa.

MA l'analisi delle termini di corrispondenza dell'output non è una buona metrica di valutazione rispetto alla plausibilità psicologica cognitiva di un di un sistema

Problemi: Se l'equivalenza è così debole non è possibile interpretare i risultati di un sistema (es. come si interpreta il fallimento di un sistema?). Un modello puramente funzionalista è una **black box** dove (riesco solo a predire il tipo di output dato il tipo di input in base ad una analogia superficiale con il sistema naturale analogo preso come fonte di ispirazione)

Esempio: Un tipico esempio di sistemi funzionalisti si può vedere i rapporti tra un uccello e Jet è un modello funzionalista dell'uccello perché vola cioè ha stesso Output anche l'uccello vola, però il modo in cui arriva a produrre quel tipo di Output ma completamente diverso rispetto a sistema naturale preso come punto di ispirazione. Il Punto è che non possiamo dire che il volo dei Jet in qualche modo ha un potere esplicativo rispetto al modo in cui volano altri tipi di volatili. Ma elemento in comune è entrambi devono obbedire alle leggi dell'aerodinamica.

Una caratteristica dei sistemi funzionalisti è quella di fare degli errori molto strani diversi con problemmi che fanno esseri umani.

Lecture 2 - Maggio 12

2- Approccio Strutturalista

Va a sostenere la necessità di avere una equivalenza più forte tra le procedure del modo in cui ho costruito un sistema e corrispondenti meccanismi cognitivi che sono noti per funzionare all'interno di un sistema naturale. quindi qui il focus non è soltanto sul fatto che i due sistemi gestiscono lo stesso input e diano lo stesso output qui il focus è anche sui vincoli che devono essere inseriti all'interno di un modello sia simbolico e confessionista e ibrido per generare una computazione di tipo *human like*. quindi ci sono degli elementi strutturali cioè una parte di struttura aggiuntiva che deve essere considerata per riuscire ad ottenere un tipo di computazione che sia compatibile a quello che facciamo noi.

uno dei vincoli che si possono andare a considerare per creare un sistema cognitivo che fanno parte di una prospettiva di tipo strutturalista sono per esempio i limiti di memoria che ne abbiamo. sappiamo che nel nostro buffer non possiamo considerare per esempio più di 6 7 elementi per volta che viene dalla psicologia esperimentale.

secondo questo tipo di prospettiva se vogliamo avere un sistema artificiale che possa avere un ruolo esplicativo rispetto al sistema naturale preso come fonte d'ispirazione non dobbiamo soltanto focalizzarci sul input e output ma dobbiamo andarci a focalizzare anche sui vincoli strutturali che andiamo a sopra imporre all'interno del nostro tipo di programma.

- **Paradosso di Wiener**

questo tipo di prospettiva è paradosso cioè questo è un paradosso noto fin dai tempi della cibernetica che si chiama paradosso Wiener che è dato da questa citazione famosa che dice s*e io voglio costruire un modello perfetto di un sistema naturale, il miglior modello è il sistema naturale stesso.*

Il problema è che non possiamo costruire una replica computazionale di un sistema naturale cioè non abbiamo la tecnologia per farlo quindi ciò di cui ci dobbiamo accontentare è cercare di avere dei buoni modelli o buoni proxy dei modelli che riescono ad approssimare in maniera significativa quelli che sono alcuni vincoli strutturali che noi vogliamo andare a inserire all'interno del nostro problema.

- **Un problema di design**

Questo paradosso di Weiner è un problema di design che Pylyshun ha spiegato:

se non formuliamo alcuna restrizione su un modello otteniamo il funzionalismo di una macchina di Turing. Se applichiamo tutte le possibili restrizioni riproduciamo un intero essere umano.

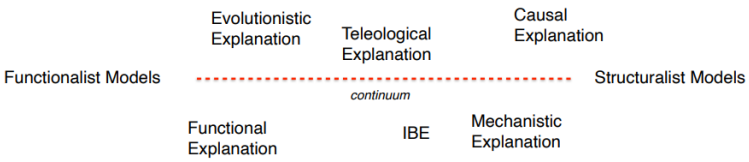
Tra il livello esplicativo del funzionalismo (basato su una analogia macroscopica tra relazioni di input-output nei sistemi naturali e artificiali) e quello microscopico dei modelli strutturalisti ci sono, in mezzo, molti possibili modelli strutturali con diverso potere esplicativo.

- **Modelli funzionalisti**

Stesse specifiche di input-out e **somiglianza superficiale** del modo in cui le componenti di un sistema e i propri meccanismi interni funzionano rispetto al sistema naturale. mi interessa soltanto del fatto che la macchina deve generare lo stesso output di un essere umano non mi non mi importa poi effettivamente come output venga generato non mi importa se viene generato all'umano

- **Modelli strutturalisti**

Stesse specifiche d'input-out spec. + vincoli aggiuntivi che forniscono una maggiore somiglianza tra sistemi naturali e artificiali. i modelli che si collocano verso questo questi tipi di aree sono modelli che ci permettono di attribuire all'output del sistema diversi tipi di spiegazioni. Sia l'approccio cognitivista che quello emergentista novelle AI possono permettere la realizzazione di modelli strutturali della cognizione.

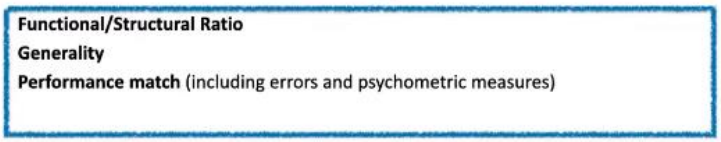


Come facciamo collocare effettivamente questi diversi tipi di sistemi che possiamo avere all'interno di questo continuum cioè quali sono le dimensioni che noi possiamo andare a considerare?

Risposta: Suppuriamo di avere una sorta di match tra quello che è l'output generato dal sistema naturale e l'output generato dal sistema dal sistema artificiale. Attenzione quando parliamo di match dell'output però ok ci riferiamo soltanto i casi di successo.

- **Minimal cognitive grid**

permette di andare a collocare i diversi programmi che vengono sviluppati dai sistemi cognitivi all'interno di questo tipo di continuum funzionalista versus strutturalista.



La prima dimensione cioè la **Ratio** non considera soltanto quanti vincoli consideriamo ma considera sostanzialmente il rapporto tra elementi funzionali ed elementi strutturali. in ogni sistema naturale c'è sempre la necessità di considerare questo rapporto perché noi non abbiamo mai un sistema completamente strutturalista.

un'altro elemento che solitamente viene considerato è l'aspetto della generalita l'aspetto della **generalità** che indica quante cose io posso fare con un certo tipo di *sistema o architettura* cioè posso fare soltanto un tipo di task come il riconoscimento di facce oppure con quell'architettura posso fare riconoscimento di facce o NLP e Etc.

Performance match significa andare a vedere non soltanto il rapporto rispetto ai casi di successo ma anche come tanto nel modo in cui sbagliano gli analoghi biologici che sono stati presi come fonte di ispirazione.

Note: alcuni di architetture sono architetture funzionaliste altre architetture sono invece architetture neurali più strutturalista. per esempio, un'architettura che vengono utilizzate dal Deep learning sono di tipo funzionalista cioè hanno un collegamento molto Lezzi.

Noti Importanti:

1. I sistemi cognitivi artificiali hanno un potere esplicativo rispetto a sistemi naturali presi come fonte di ispirazione solo se sono modelli strutturalmente validi **(realizzabili in modi diversi e modellabili a diversi livelli di astrazione)**.
2. **I sistemi cognitivi artificiali possono fungere da "esperimenti computazionali"** (possono fornire risultati in grado di ridefinire la conoscenza originaria riferita al sistema naturale di riferimento).

La metodologica che seguono gli approcci cognitivi si parte da una Teoria psicologia che cerca di formare una teoria che poi si può crea un programma, al momento che otteniamo il programma possiamo lanciare una simulazione e poi possiamo confrontare l'output della nostra simulazione con la parte sperimentale che di solito viene fatta da altre persone o da psicologi o da biologi o da neuroscienziati o da etologi.

Perché in un sistema artificiale dovrei andare a replicare/prendere spunto dal modo in cui noi risolviamo problemi dato che commentiamo anche molti errori? E se non fossi interessato agli aspetti di spiegazione del sistema?

- **Problema di Linda**

dati questi indizi su questa persona linda dovete andare a dare una classificazione di linda cioè voi quello che sapete

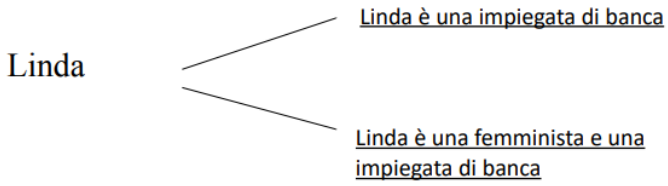
-Linda era giovane negli anni ‘70s

-A Linda piace il colore rosso

-Linda si è laureata in filosofia

- Linda è contraria all’uso di energia nucleare (attivista “green”)

Dobbiamo scegliere uno tra queste due opzioni.



NOTE: la stragrande maggioranza dei ragionamenti di senso comune sono ragionamenti automatici che noi facciamo automatico che utilizziamo e ci permettono in qualche modo di utilizzare il fatto che viviamo in un mondo di informazione incompleta quindi queste

sono delle euristiche nel senso che non è proprio non è giusta la risposta, però nel 99% dei casi è giusto.

Domanda**: **Quale la Differenza tra sistema biologicamente ispirato e sistema cognitivamente ispirato?

la differenza principale riguarda il livello di astrazione che viene considerato. i sistemi biologicamente ispirati sono vincolati ad aspetti di tipo neurale cioè la struttura neurale anatomica del cervello è invece gli aspetti di tipo cognitivo in qualche modo emergono da questi vincoli strutturali di tipo anatomico cioè io non sono interessato nel caso di una modellazione biologicamente ispirato ma sono interessato a vedere se l'informazione che passa da un neurone all'altro della mia rete segue qualche tipo di principio che si ritrova all'interno della corteccia prefrontale oppure di altre parti del cervello.

Quando invece mi occupo di sistemi cognitivamente ispirati mi focalizzo un livello di astrazione più alto quindi utilizzo anche dei paradigmi diversi cioè se in un caso utilizzo equazioni differenziali o sistemi connessionisti e etc, in un altro caso utilizzerò degli approcci ibridi oppure degli approcci simbolici, per esempio, posso decidere di utilizzare l'euristica Means End analysis.

- **Modelli di Razionalità**

Esistono diversi modelli di Razionalità che stati sviluppati, ad esempio

- ***Expected Utility (Morgenstern, von Nuemann):*** l'idea è che noi esseri umani siamo dei decisori ottimali quindi questa era la teoria predominante in tutti i modelli matematici fino a più o meno gli anni 60 70.
- ***Bound Rationalty (Herbert Simon):*** lui ha dimostrato tutti i modelli basati per esempio su una massimizzazione della funzione di utilità attesa Expected Utility siano in realtà dei modelli che non funzionano nel reale e non spiegano il modo in cui noi esseri umani non prendiamo delle decisioni sulla base di una di un principio di massimizzazione dell'utilità perché non siamo dei bravi calcolatori, quella che si definisce razionalità bounded cioè limitata abbiamo una serie di limiti.

quindi l'esempio del problema di Linda è un esempio tipico che viene utilizzato per dimostrare come noi siamo degli esseri a bounded rationality. un essere perfettamente razionale avrebbe scelto sempre l'opzione 1 nel caso della classificazione di linda. Buonded Rationality sia suddivisa su due punti di vista diversi:

  - ***Cognitive Baises (Kahnman, Tversky):*** secondo loro il focus era sull'errore.
  - ***Heuristics (Gingerenzer):*** ha sottolineato l'importanza il ruolo euristico di questo tipo di strategie decisionali. cioè il fatto che noi siamo degli esseri a

bounded non deve essere considerato come il fatto che siamo dei sistemi proni agli errori **MA** deve essere considerate che questi errori sono in realtà delle euristiche che dal punto di vista evolutivo ci hanno permesso di prendere delle buone decisioni anche se non ottimali però buone decisioni in condizioni di incertezza.

- ***Bounded-resource Rationality:*** è legato al fatto che noi siamo degli esseri umani a razionalità limitata che hanno anche delle risorse limitate.

Livelli di Astrazione e Strategie di Modellazione di sistemi cognitivi

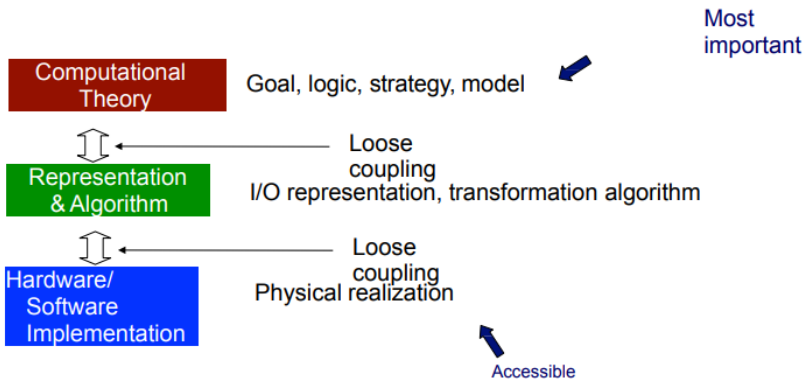
quali sono delle strategie che di solito vengono utilizzate per progettare dei sistemi cognitivi e sistemi artisti cognitivi?

ci sono ce ne sono sostanzialmente due

1- *Gerarchia di Marr*

un'analisi che permette di andare ad analizzare il comportamento di un qualsiasi sistema artificiale sia cognitivamente ispirato che non cognitivamente ispirato utilizzando tre livelli

- **La teoria di Computazionale:** si focalizza soltanto sugli aspetti di input/output cioè qual è il goal di questo programma qual è l'input che gestisce qual è l'output che deve produrre quali sono le strategie che devono essere in gioco.
- **Rappresentazione e Algoritmo:** quali sono le strutture dati gli algoritmi che permettono di implementare la teoria computazionale del livello di sopra
- **Implementazione di Hardware e Software:** il livello più basso.



- **Esempio: Registratore di Cassa**

A livello computazionale, il funzionamento del registro può essere spiegato in termini di aritmetica e, in particolare, in termini di teoria dell'addizione: a questo livello sono

rilevanti la funzione calcolata (addizione), e le sue proprietà astratte, come commutatività o associatività.

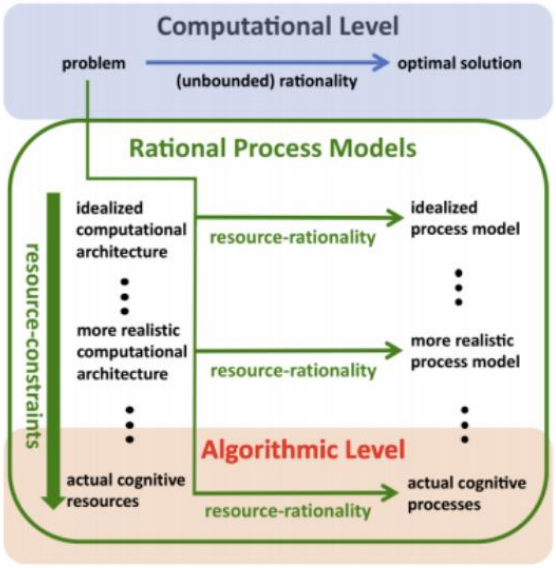
Il livello di rappresentazione e l'algoritmo specificano la forma delle rappresentazioni e i processi che le elaborano: “potremmo scegliere numeri arabi per le rappresentazioni, e per l'algoritmo potremmo seguire le consuete regole di sommare prima le cifre meno significative e 'portare' se la somma supera 9”.

Infine, **il livello di implementazione** ha a che fare con il modo in cui tali rappresentazioni e processi sono realizzati fisicamente; ad esempio, le cifre potrebbero essere rappresentate come posizioni su una ruota metallica o, in alternativa, come numeri binari codificati dagli stati elettrici dei circuiti digitali.

Note: Un problema deve essere modellato prima al livello della teoria computazionale, senza riferimento ai livelli di astrazione più bassi. Successivamente: algoritmi e rappresentazioni Infine: valutare tra diverse implementazioni possibili

I 5 step del Paradigma di Resource Rationality

1- la prima cosa si parte dal computational level dove si riferisce alla teoria computazionale di Mar quindi analizzo un problema in termini di goal, cioè che cosa devo risolvere quali sono gli input e output.
2- si scende e si inizia a vedere quali sono le classi di algoritmi che possono essere utilizzati per risolvere un problema.
3- Una volta che io ho selencato tutti gli algoritmi che mi possono servire per risolvere alcuni tipi di problemi, cerco di andare a vedere quelli sono gli algoritmi che hanno un trade off migliore tra le risorse computazionali che devono essere utilizzati e la cura che viene gestita per andare a modellare quel tipo di problema.
4- cerco di vedere se ci sono delle discrepanze tra le assunzioni che io ho fatto nel mio modello perché qui sono ancora a livello di ipotesi dalle assunzioni che io ho inserito nel mio modello è invece quelli che sono i risultati sperimentali che si trovano in altre discipline.
5- Quindi una volta che ho fatto confronto predittivo cerco di sistemare il modello o i tipi di algoritmi che ho un utilizzato in modo tale che in termini di modale che la performance match inizi ad essere sempre più significativa.


Computational Level
problem
(unbounded) rationality
optimal solution
Rational Process Models
idealized computational architecture
resource-rationality
idealized process model
more realistic computational architecture
resource-rationality
more realistic process model
resource-constraints
Algorithmic Level
actual cognitive resources
resource-rationality
actual cognitive processes

Lecture 3 - Maggio 12

Il Sistema cognitivista

assume che la condizione sia un tipo di computazione o motto in cui ci sono tutta una serie di vincoli che vengono utilizzati da cognitivista ora vedremo un po' quali sono le carte. Le rappresentazioni delle assunzioni sono:

- Esplicite e simboliche
- Denotano oggetti esterni
- Proiettano la realtà esterna nei meccanismi interni di information processing mentale.

Assunzione: accesso condiviso a tali rappresentazioni.

Physical Symbol Systems [Newell and Simon 1975]

alla base di questo tipo di assunzione l'idea è che qualsiasi agente intelligente di tipo generale come siamo noi o altri animali si sia un sistema fisico di singoli cioè è un sistema fisico nel senso che la sua intelligenza dà la capacità di manipolare quelle che sono rappresentazioni mentali che sono dei simboli. Ogni sistema che esibisce intelligenza generale è un physical symbol system. Un sistema simbolico fisico è ‘una macchina che produce nel tempo un insieme in evoluzione di strutture simboliche. Questi sistemi sia naturale che artificiale hanno queste caratteristiche:

- Memoria (contiene informazione simbolica)
- Simboli
- Operazioni (per manipolare simboli)
- Interpretazioni (permettono ai simboli di specificare operazioni)

I “symbol systems" possono essere istanziati su diverse implementazioni. Secondo gerarchia di Marr, *Physical Symbol System* possa avere diversi tipi di implementazione cioè possa girare su diversi tipi di hardware che puo essere non considerato.

a questo tipo di assunzione rappresentazionale seguono anche delle assunzioni di tipo algoritmi cioè all'interno di questa teoria risolvono i problemi sono tramite questa ricerca euristicha. Un sistema di simboli fisici utilizza la sua intelligenza in attività di problem-solving tramite meccanismi di ricerca dove Per essere effettiva la ricerca deve essere efficiente.

Physical Symbol System con lo sviluppo più intensivo delle neuroscienze è stata smentita non è una teoria biologicamente plausibile perché nella nostra testa lo sviluppo di

tecniche di neuroimmagine sempre più sofisticata. non c'è nessun simbolo se mettiamo una persona all'interno di una macchina per fare FMI. mentre svolgere i compiti non ci sono dei simboli che si attivano il livello di attivazione è sempre a livello neurale per quale motivo però questo tipo di approccio ancora perchè nonostante non abbia una plausibilità biologica cioè non ci sono dei simboli a un certo punto per qualsiasi tipo di task. si auto organizzano in modo tale da svolgere delle funzioni simili a quelle assunte all'interno della fisica al *Physical Symbol System.*

sulla parte questo tipo approccio è stato sviluppato una serie di risorse e esplicite simboliche che sono utilizzate ancora oggi tipo pensate a wordnet o ConceptNet. Negli anni 80 cognitivismo era a mano dove dovevo scrivere tutto a mano ma oggi la conoscenza non è solo manualmente ma posso essere presi attravo

- Machine learning
- Modellazione probabilistica
- Modelli miglior
- Logiche migliori

I Sistemi Emergentisi

Ci sono tre elementi che hanno un approccio comune che si chiama approccio **emergentisi.** La cognizione è un processo emergente che avviene tramite meccanismi di auto-organizzazione (self-organization) e realtime, Cioè il fatto che un sistema non è dato una volta per tutte ma si costruisce continuamente nel tempo interagendo con ambiente stesso. Esiste un meccanismo di feedback tra agente e ambiente.

Differenza tra assunzione Emergentisa e Cognitivista

la prima differenza è data dal fatto che

1- agente e ambiente si determinano che questo è molto diverso rispetto all'assunzione cognitivista in cui l'ambiente è un qualcosa di dato a cui noi che siamo tutti agenti con lo stesso spazio condividiamogli stessi tipi di rappresentazioni di quell'ambiente allora ci capiamo, **Ma** qui ogni agente in base al tipo di interazione che fa con l'ambiente si va a creare una propria rappresentazione.
2- negli approcci **Emergentisi** viene dato un ruolo molto importante al corpo cioè gli elementi corporei, fisici e ci permettono di interagire con l'ambiente, secondo gli approcci **Emergentisi** un sistema cognitivo non può non essere un agente embodied deve per forza avere un corpo altrimenti non è un sistema cognitivo.

N.B: in questo caso il focus è sempre stato in qualsiasi d'approcci dinamici e connessi su aspetti legati all'apprendimento.

Auto-Organizzazzione

"**I sistemi auto-organizzanti** sono sistemi fisici e biologici in cui il modello e la struttura a livello globale derivano esclusivamente dalle interazioni tra i componenti di livello inferiore del sistema".

"Le regole che specificano le interazioni tra i componenti del sistema vengono eseguite solo utilizzando informazioni locali, senza riferimento al modello globale."

Emergenza

Un processo attraverso il quale un sistema di elementi interagenti acquisisce schemi e strutture qualitativamente nuovi che non possono essere intesi semplicemente come la sovrapposizione dei singoli contributi.

Approcci Dinamici

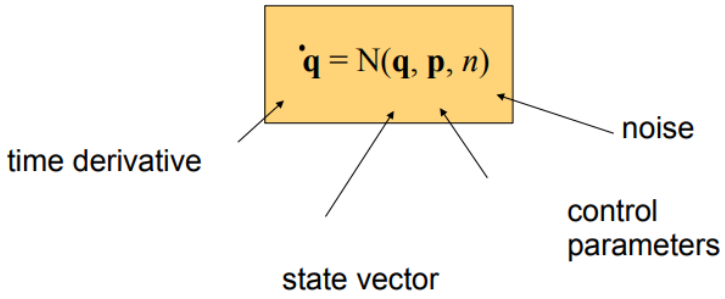
per utilizzando questo tipo di paradigma di modellazione si va a modellare sostanzialmente dei sistemi che hanno queste caratteristiche:

Dissipative: in qualsiasi azione diffondono l'energia

Non-equilibrium: l'azione di questi sistemi è dettato dal tentativo di raggiungere

Non-linearity: la dissipazione non è uniforme: un piccolo numero di gradi di libertà del sistema contribuisce al comportamento.

Il modo per andare a gestire questo tipo di fenomeni è utilizzando delle pressioni differenziali.



Approccio Enaction

un sistema cognitivo deve necessariamente avere un corpo altrimenti non è un sistema cognitivo cioè il controllo del corpo è un elemento centrale. qui l'idea è che la gente ed ambiente si determinano insieme. e noi condividiamo un ambiente nella prospettiva inactive ciascuno di noi si costruisce una vista diversa dell'ambiente. Per comunicare gli agenti con insieme debbano avere completa autonomia con l'esterno.

5 elementi chiave

- Autonomia (nessun controllo esterno)
- Embodiment (corpo)
- Emergence (cognizione emerge dall'interazione)
- Esperienza (intenzione passate modificano agente e ambiente)
- Sense-making (capacità di auto-modifica del comportamento)

Ci sono dei problemi pertutti questi paradigmi e Non sono ugualmente maturi

- **Dynamical systems Enactive (& Dynamical)**

  – Problemi per moderazione di fenomeni cognitivi di alto livello

- **Cognitivist systems**

  – Al momento i più avanzati ma problemi con aspetti percettivi

- **Hybrid systems**

  – Non è chiaro come combinare tali filosofie antagonist

Architettura Cognitivo

Obbiettivo principale è legato all'idea di provare a fornire delle teorie computazionali che legati a intelligenza generale cioè in che modo è possibile andare a creare dei sistemi in grado di svolgere contemporaneamente più tipi di facoltà cognitiva. un'intelligenza generale non nasce mettendo insieme singoli sistemi di intelligenza artificiale specifici per diversi tipi di compito. Dobbiamo andare a vedere i meccanismi indipendenti dei singoli tasti della cognizione che possono fungere appunto da elementi architetturali unici per poter generare qualsiasi tipo per potere gestire qualsiasi tipo di funzionalità.

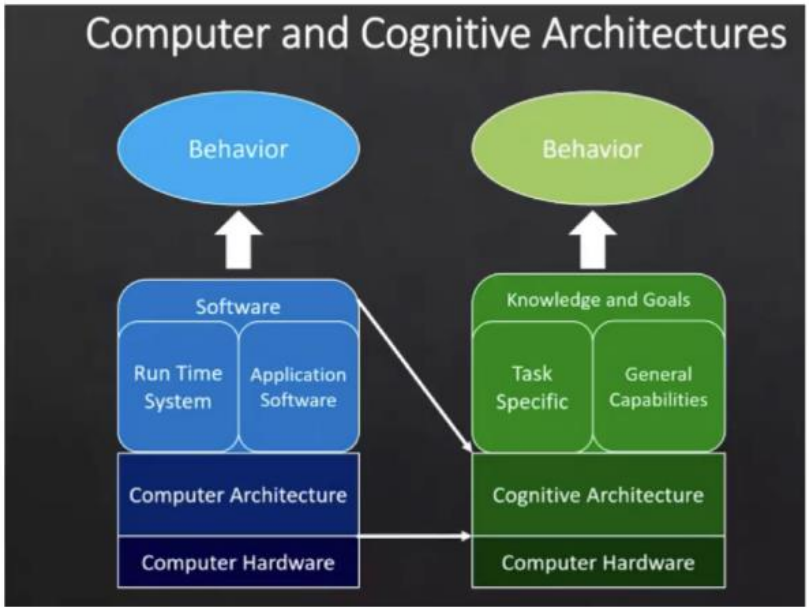
Definizioni:

1- "Il termine "Architetture Cognitive" indica sia modelli astratti di cognizione, in agenti naturali e artificiali, sia le istanziazioni software di tali modelli che vengono poi impiegati nel campo dell'Intelligenza Artificiale (AI). Il ruolo principale delle Architetture Cognitive nell'IA è quello di consentire la realizzazione di sistemi

artificiali in grado di esibire comportamenti intelligenti in un contesto generale attraverso un'analogia dettagliata con il funzionamento costitutivo e evolutivo e i meccanismi alla base della cognizione umana".

2- "Le architetture cognitive sono state storicamente introdotte per tre ragioni principali: i) catturare, a livello computazionale, i meccanismi invarianti della cognizione umana, compresi quelli alla base delle funzioni di ragionamento, controllo, apprendimento, memoria, adattabilità, percezione e azione (questo obiettivo è cruciale nella prospettiva cognitivista), ii) formare le basi per lo sviluppo delle capacità cognitive attraverso l'ontogenesi su lunghi periodi di tempo (questo obiettivo è uno degli obiettivi principali della cosiddetta prospettiva emergente), e iii) raggiungere intelligenza a livello umano, detta anche General Artificial Intelligence, mediante la realizzazione di manufatti artificiali costruiti su di essi".

ci sono delle possibili analogie che si possono far architetture cognitive e le architetture calcolatori standard, immaginiamo di uno stack tipico di architettura per calcolatore abbiamo l'hardware e software e poi le letture per calcolatore che ci permettono ai software condividere un dei meccanismi standard di information processing. lo scopo delle architetture cognitive è esattamente fare lo stesso tipo di lavoro quindi nell'ambito delle cognitive l'idea è che di avere una sorta di middleware che ci permette di andare a definire gli elementi standard di information processing attraverso cui noi vogliamo andare a fare interagire la conoscenza di un singolo agente rispetto ai goal che deve andare a raggiungere.



Esistono circa 84 di diversi tipi delle architetture che ciascuna architettura cognitiva va a implementare una diversa teoria della mente.

I goal sono diversi negli approcci di Architettura cognitive ad esemepio:

le architetture biologiche che lavorano su processi di computazione che si collocano su delle Times scale cioè si collocano in modalità di computazione molto breve, pochi millisecondi.

le architetture psicologica cercano di andare a modellare i tempi di reazioni due di un task. voglio creare un sistema che sia in grado di giocare a poker nel modo in cui gioca un essere umano andando a replicare e cercare di capire quali sono le strategie utilizzate dagli esseri umani. vengono utilizzati per fare certi tipi di mosse

AI Functionality in realtà non hanno una vera funzione esplicativa, non sono interessati alla parte di inferenze inversa però utilizzano una sorta di sistemi generali come blu print per andare a generare un sistema intelligente che faccia delle cose specifiche in un ambito. utilizzare queste strutture cognitivamente o biologicamente ispirate possa dare loro dei vantaggi quando vanno a costruire dei sistemi specifici.

Newell Time Scale of Human Action

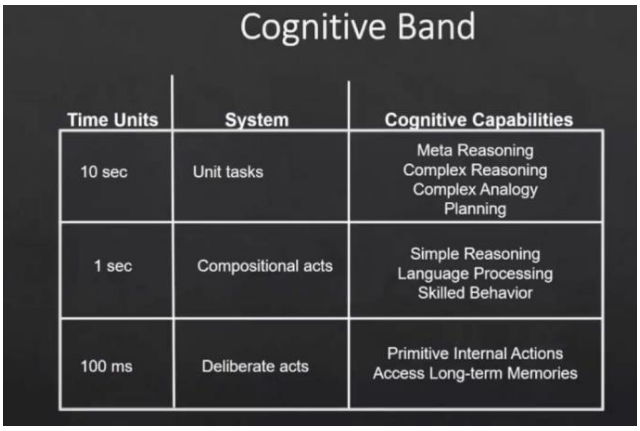
Il comportamento di un agente intelligente possa essere descritto da attività che si possono suddividere in quelle che lui chiama diversi tipi di band biologica cognitiva razionale e sociale, che hanno diversi tipi di Time Scale nel senso che ci sono alcuni tipi di attività di un sistema biologicamente cognitivamente ispirato che avvengono in meno di un millisecondo e questo di solito è quello che viene modellato quando si utilizzano dei sistemi biologicamente ispirati.

Quando si cerca di invece studiare un fenomeno di tipo cognitivo si va su quella che viene definita la cognitive band che riguarda tutte quelle azioni quelle decisioni che intercorrono tra i 100 millisecondi e i 10 secondi.

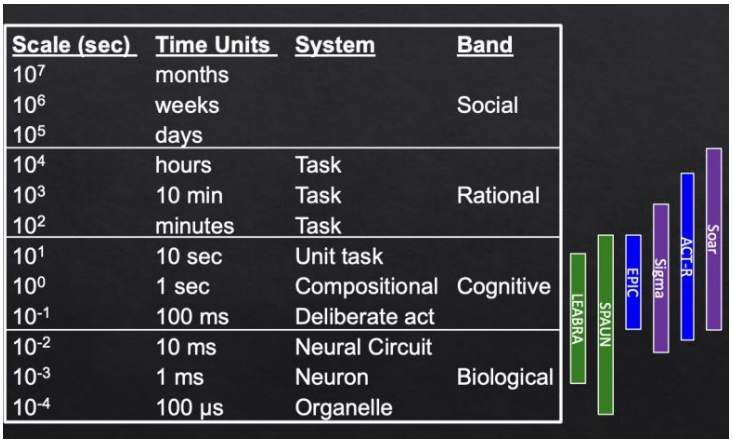
Band razionali hanno a che vedere con meccanismi di computazione che partono da alcuni minuti e ci impiegano delle ore. ancora c'è tutta la band sociale che invece prevede che ci siano dei meccanismi di computazione che partono da alcuni giorni vanno avanti per mesi. il focus principale delle architetture cognitive possiamo che si concentra su aspetti biologici e su aspetti di tipo cognitivo.

Scale (sec)	Time Units	System	Band
10^7	months		
10^6	weeks		Social
10^5	days		
10^4	hours	Task	
10^3	10 min	Task	Rational
10^2	minutes	Task	
10^1	10 sec	Unit task	
10^0	1 sec	Compositional	Cognitive
10^{-1}	100 ms	Deliberate act	
10^{-2}	10 ms	Neural Circuit	
10^{-3}	1 ms	Neuron	Biological
10^{-4}	100 µs	Organelle	

ogni attività conosciuta della nostra mente e del nostro cervello è mappata su questo time scale delle azioni umane. quindi noi sappiamo per alcune attività quali sono le tempistiche che avvengono che vengono utilizzate.



esiste una suddivisione delle nostre capacità cognitive e un mapping diretto con i tempi. a seconda del focus di modellazione diverse architetture cognitive ci si focalizza su unità di tempo di tipo diverso.



Common Model of Congition

Prova a trovare una sorta di sintesi tra questi diversi tipi di architetture, il punto è cercare di vedere quali sono gli elementi in comune nell'ambito di diversità. una area attiva di ricerca riguarda la realizzazione di **common model of Congition** che cerca sostanzialmente di provare a mettere insieme quelli che sono gli elementi cardine che caratterizzano i diversi tipi della mia architettura cognitiva. Che loro sono:

1- Long-Term Memory

una memoria sempre a lungo termine in cui vengono messe delle regole quindi la memoria a lungo termine quella dei nostri ricordi passati è divisa in una memoria fattuale del tipo *Roma è la capitale d'Italia* è un fatto e in una parte procedurale del tipo e *se voglio bere dalla bottiglia devo prima togliere il tappo* tutta la nostra conoscenza viene suddivisa all'interno di questi due tipi di memorie a lungo a lungo.

2- Working Memory

gestisce sostanzialmente quelli che sono le attività correnti cioè i tipi di informazione che prendiamo dalla memoria a lungo termine mettiamo la memoria a breve termine per poterlo utilizzare fare delle azioni oppure per fare altri tipi di ragionamenti.

3- Moduli percettivi e motori

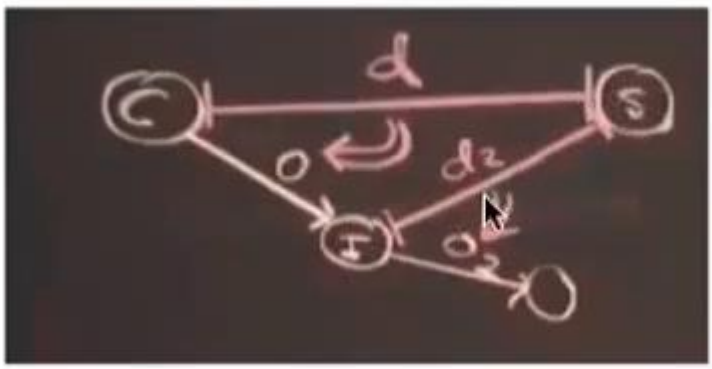
ci permettono di dare azione alla componente corporea alla parte di controllo motorio e invece la parte percettiva gestisce l'interpretazione di quelli che sono i segnali del che vengono dall'ambiente esterno.

Lecture 4 - Maggio 17

GPS – GENERAL PROGRAM SOLVER –1959

un programma in grado di utilizzare una serie di euristiche per poter risolvere diversi tipi di problemi è una delle discipline utilizzate della logica means-end analysis. l'idea di GPS era quello che avevamo un cosiddetta **problem solving hypothesis** cioè abbiamo uno spazio simbolico in cui si va a rappresentare un tipo di problema, abbiamo uno stato iniziale che può essere per esempio rappresentato dal nostro **stato c,** è uno stato goal che può essere rappresentato dal nostro **stato S.** lo scopo è quello di cercare un modo di portare il nostro agente dallo **stato c allo stato S** quindi abbiamo bisogna utilizzare degli operatori che ci permettono di minimizzare fare quella che è la distanza simbolica tra questi due stati.

creo un sotto gol quindi io devo andare da C ad S. non ho un modo per riuscire a risolvere direttamente questo problema quindi calcolo la distanza tra **C** ed **S** e vedo quali sono le mosse possibili che posso fare, praticamente sono gli operatori che posso andare a selezionare che mi portino in un altro stato che chiamiamo **i** che ha una distanza di **d2** dove è una distanza minore rispetto alla distanza originale **D** quindi io mi sto avvicinando alla risoluzione diciamo così del problema. È stata presa in gran parte anche dalla architettura cognitiva SOAR.



Questo approccio aveva una serie dei problemi, per andare a collezionare quali erano le distanze che aumentavano o diminuivano e i tipi di operatori che potevano essere utilizzati al fine di minimizzare queste distanze era un qualche cosa che si trovava all'esterno di programma, quindi, bisognava mantenere una sorta di mega

database che erano spesso mantenuti a mano. il meccanismo di risoluzione di problemi come minimizzazione della distanza dal goal che voglio raggiungere all'interno di uno spazio degli Stati è stato interamente ereditato dall'architettura SOAR.

SOAR

è un'architettura ad agenti che ha ***long term memory*** e una ***short term memory*** e poi ha dei vari *moduli* che sono commessi con elementi percettivi o motori che possono essere visivi. una distinzione tra queste due memorie viene direttamente dalla psicologia cognitiva perché anche noi esseri umani abbiamo questi diversi tipi di memoria.

Altro concetto che viene dalla psicologia congitiva è L'importanza della conoscenza procedurale cioè dalla conoscenza di **regole**. noi abbiamo una serie di regole che sono Built-In della nostra **produzione** e poi abbiamo tutta un'altra serie di regole che sono apprese con l'esperienza, per esempio, non gettare dell'acqua su vicino a un computer che sta usando la corrente. queste regole sono una parte importante della long term memory.

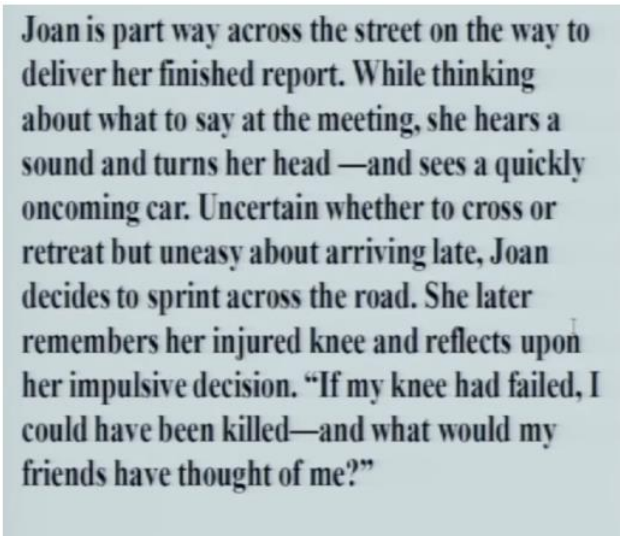
Un aspetto importante però rispetto a molte architetture è che c'è un sistema di **preferenza delle regole**. le regole non sono tutte uguali all'interno di questa architettura, alcune regole in qualche modo assumono più importanza di altre in base al contesto a cui si fa riferimento.

un'altro elemento centrale che è interamente ereditato da GPS è l'idea di risoluzione di problemi, come soluzione di problemi all'interno di uno spazio degli Stati quindi questa idea dello spazio del problema ***(Problem Space)*** è interamente preso da GPS.

uno degli elementi importanti di questa struttura cognitiva è quello di **Universal Subgoaling.** il punto è che per risolvere dei problemi complessi non si riesce ad arrivare subito alla soluzione che vogliamo ottenere. dobbiamo procedere per gradi, vuole dire che se io non riesco a raggiungere immediatamente il goal che voglio andare a raggiungere, devo avere delle procedure che mi permettono di spezzettare il problema in sotto problemi cioè avere una serie di sotto goal. Questo approccio significa di avere una serie di procedure di creazione automatica di

sottogoal che mi permettono di minimizzare la distanza rispetto al problema che voglio andare a risolvere.

Minsky Problem Solving Layer



dice Minsky che un primo livello di problem solving per poter comprendere questa storia è quella di poter anche caratterizzarla e che dal fatto che c'è una **reazione istintiva.** Tutte le altre che capiamo con l'esperienza fa parte di un altro layer di intelligenza che si chiama **reazioni apprese.** e poi c'è il livello del **pensiero deliberativo** cioè la ragazza decide di attraversare la strada a un certo punto giusto. esigere attraversare la strada una volta che ha deciso di attraversare la strada è **il livello riflessivo** cioè riflette sulla decisione per riflettere sul fatto che magari poteva essere pericoloso attraversare in quel momento, poteva aspettare. poi c'è un livello superiore che è **riflessione su se stesso (Self-Reflection)** cioè riflette un certo punto sul fatto che il poteva rompere una gamba, quindi, poteva farsi di nuovo di nuovo male in caso di incidente. e poi c'è **un livello sociale** cioè pensa a cosa gli amici possono pensare di lei.

N.B: non c'è un software in questa architettura. è un'avventura più a livello teorico però ha avuto un grosso impatto sulla comunità l'elemento che connette questi diverse leghe di *problem solving* e la conoscenza di senso comune cioè la conoscenza legata a ciò che tipicamente succede nel mondo.

Subsumption Architecture (Brooks)

negli anni 80 l'idea era che per avere un robot che facesse delle cose intelligenti bisognava avere un **modulo di visione, un modulo di ragionamento,** un modulo di azione e per cercare di migliorare le capacità di robot bisognava andare a migliorare singolarmente questi moduli. il punto è che quando si passava dalla versione precedente le cose che prima funzionavano, non funzionavano piu.

Brooks utilizza l'idea di minsky del fatto che in qualsiasi agente intelligente naturale artificiale debba avere questa capacità di avere problem solving in layer **Ma** lui gira 90 ° questo tipo di architettura verticale cioè la sposta in modo orizzontale. così spostando in orizzontale lui individua una serie di attività crescenti che diciamo per esempio un sistema robotico perché deve essere in grado di risolvere i problemi innanzitutto evitare ostacoli e poi andare per ad esplorare l'ambiente o nuovi tipi di oggetti. l'idea è che per ciascuna di queste capacità cognitive noi possiamo avere tutti e tre i moduli insieme. questo tipo di organizzazione architetturale permetteva che la possibilità che passando da un livello più basso livello superiore non si perdesse quello che si era tenuto a livello precedente.

si dice Subsumption perché man mano che si sale nelle capacità che vengono acquisite non si perde ciò che si è acquisito al livello giu.

Ci sono tre elementi fondamentali di questa proposta:

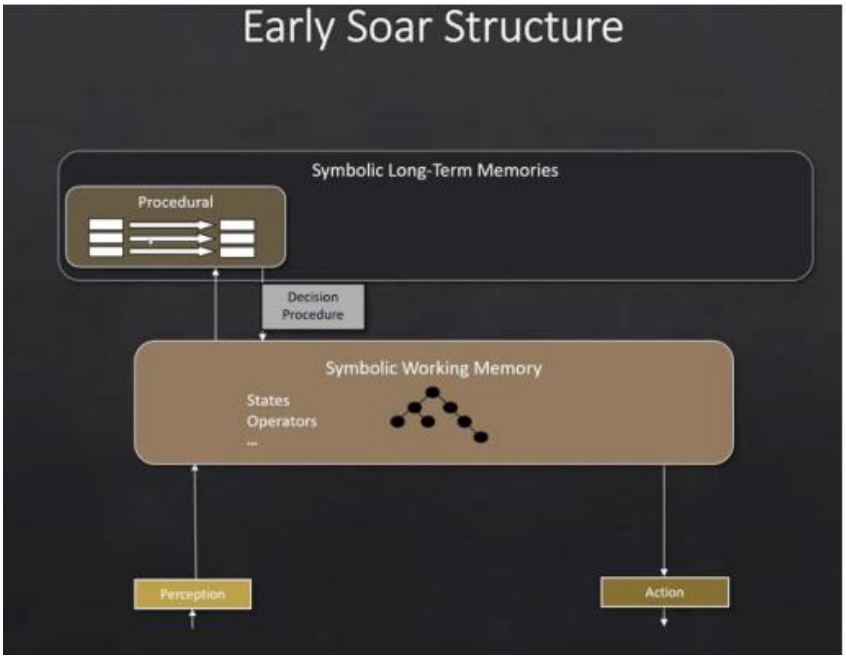
1- non c'è nessuna rappresentazione
2- È emergenista, usa la parola al posto di Modello. non c'è nessun modello esplicito formale del mondo significa che io sono la agente che deve crearsi di volta in volta una sua rappresentazione del mondo tramite l'interazione fisica e riesco a fare con l'ambiente.
3- Che per riuscire ad avere questi diciamo livelli crescenti di capacità va bene cognitiva banalmente se dovessero utilizzare degli automi a Stati finiti.

il modo in cui conosciuti letteratura questo tipo di architettura il modello di architettura oltre al nome pittura di disostruzione è basato sulla ipotesi alla base di questa architettura che si chiama la in inglese feature hypothesis cioè l'idea è che se riusciamo a rappresentare un modello di intelligenza di un insetto che è rappresentabile in questo modo sopra, possiamo poi andare a estendere questo modello che soprattutto reattivo.

Soar Cognitive Architecture

È stato proposto da Newell e Rosenbloom nell'1981, Non è un linguaggio è un middleware software che è scritto in C, tutto la parte di information proccessing viene da parte di questa piattaforma. Per costruire degli agenti che usano i meccanismi information processing generale esiste una sintassi che deve essere utilizzata che ci permette di costruire un agente che poi rispecchia quello che è il ciclo di elaborazione di decisione che già predefinito è già Built-in all'interno di Soar.

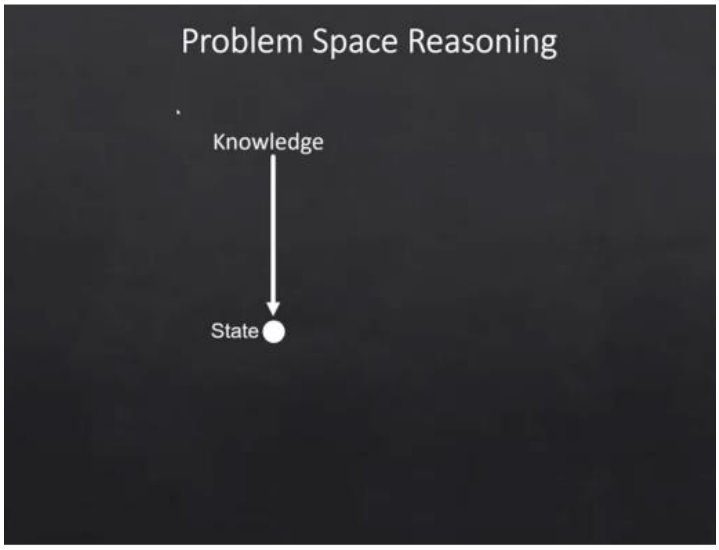
l'architettura iniziale di soar aveva tipo di Long Term Memory che era diviso in modo procedurale dove c'erano delle regole e memoria simbolica che era memoria di Symbolic working memory dove le informazioni sullo stato del mondo erano rappresentate tramite diversi tipi di stati e operatori che potevano essere applicati a questi Stati. poi c'erano due moduli esterni che permettevano a questo sistema di memoria che comunicava internamente anche di comunicare con l'ambiente esterno.



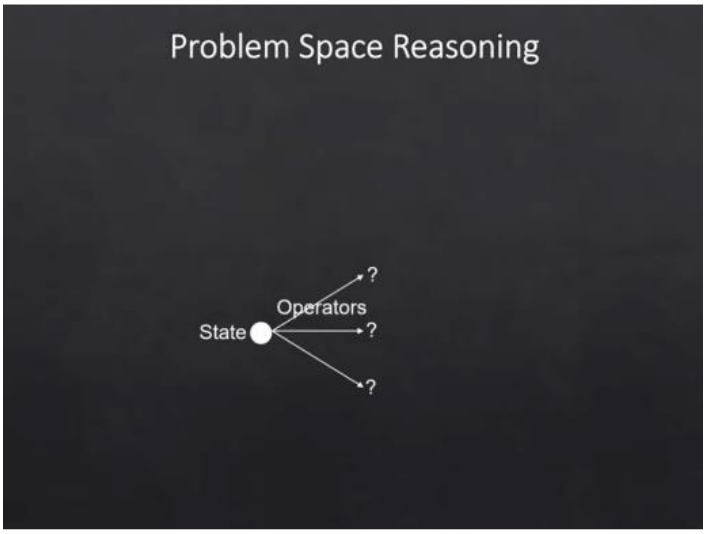
come viene gestita la modalità di comportamento intelligente in Soar?

abbiamo questa idea dei problem spaces che viene ereditato dal GPS quindi noi abbiamo dei vari stati che caratterizzano un problema degli operatori che ci permettono di passare da uno stato all'altro del problema e che sostanzialmente

cercano di modificare lo stato iniziale per cercare di farci andare ad uno stato che sia più vicino al goal.



Abbiamo lo stato Iniziale del problema e un po' di conoscenza che possiamo aggiungere a questo stato prendiamo dai nuovi termini (le regole), lo stato si trova a short term memory. la giunta della conoscenza ci permette di poter avere diversi operatori con cui possiamo andare a selezionare per risolvere un tipo di problema. Per scegliere un operatore il sistema Soar ha un sistema delle preferenze.



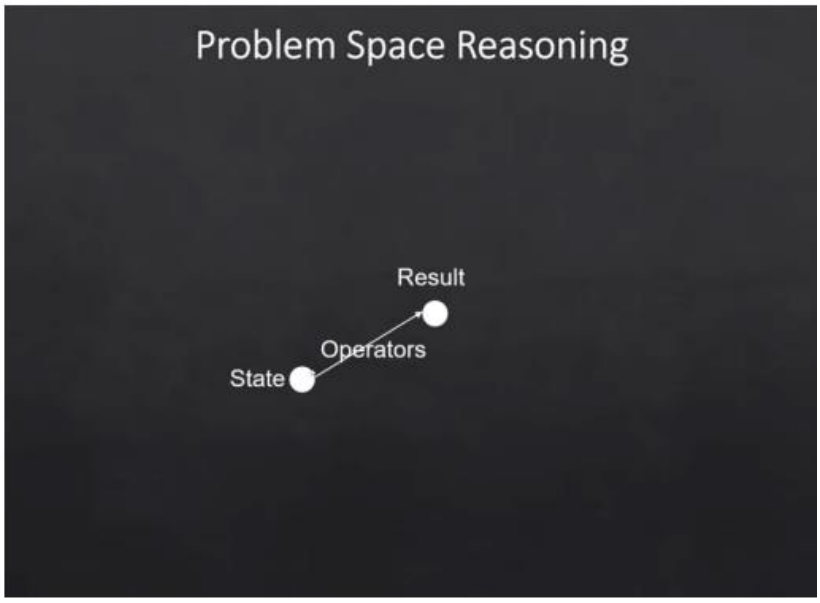
In un modo generale possiamo avere:

- Nessun operatore che posso applicare in quello stato
- Oppure: molti operatori. Quale scegliere?

- Stato di **impasse**

Stato di impasse

l'impasse è la prima parte dell'apprendimento, insomma, una volta che ci troviamo in uno stato di impasse significa che non sappiamo come si può andare avanti quindi dato che non sappiamo come andare avanti l'impasse crea il meccanismo del subgoaling automaticamente. L'applicazione della conoscenza non si applica soltanto allo stato di cose iniziale ma si applica anche ai tipi di operatori che posso andare ad utilizzare rispetto allo stato di cose iniziale che voglio andare a modificare. una volta che applico questa conoscenza ho una tripletta di cose che sono appunto **state iniziale**, l**iste operatori che ho applicato e il risultato dove** il risultato è un nuovo stato iniziale che ho modificato.

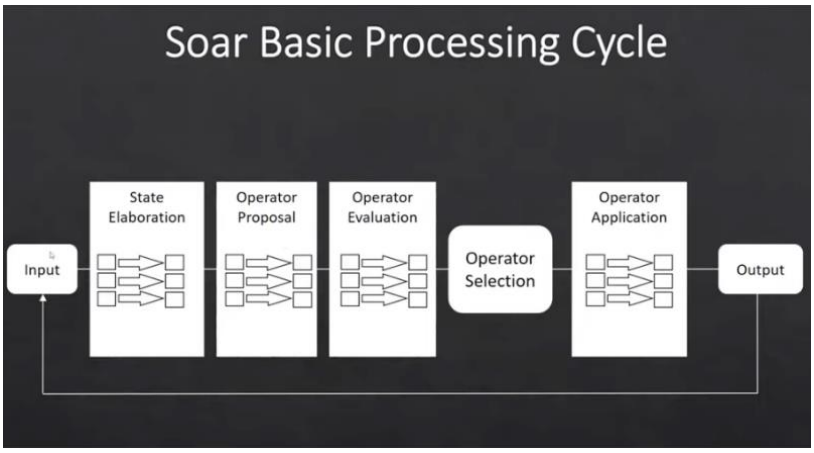


ogni operatore si porta con sé anche la conoscenza su quando può essere applicato perché non può essere applicato in ogni situazione. Anche sul nuovo stato cioè result posso applicare nuova conoscenza.

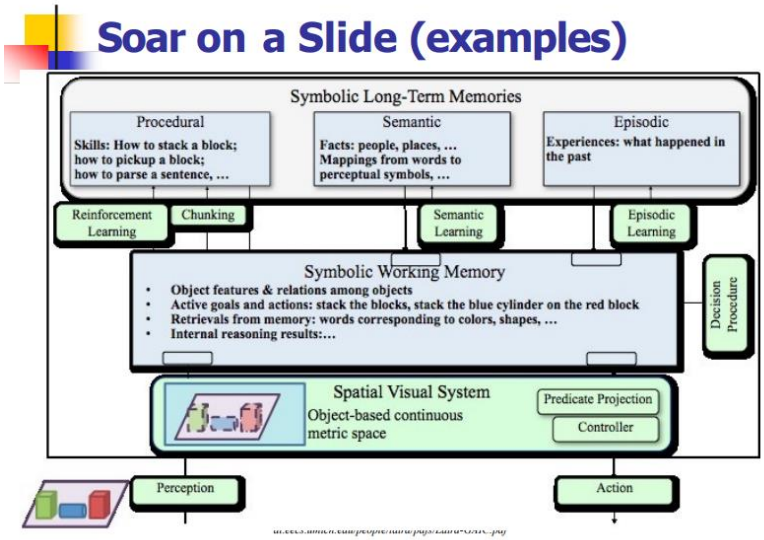
Soar Basic Processing Cycle

Esiste un **Input** che può avvenire tramite diversi modi sia moduli percettivi motori eccetera. c'è **una elaborazione iniziale dello Stato,** quindi**,** viene creata in automatico uno stato iniziale da cui la agente soar parte c'è una proposta di uno o

più operatori che possono essere applicati in questo stato il fatto che possono essere applicate o meno fa parte un built-in dell'operatore. gli operatori veniamo a decidere noi oppure la oppure la agente impara tramite dei meccanismi di learning. c'è una modalità di **valutazione dell'operatore, c'è** una **selezione dell'operatore, un'applicazione dell'operatore, un nuovo stato** che viene raggiunto e poi che diventa l'input per un nuovo ciclo di elaborazione.



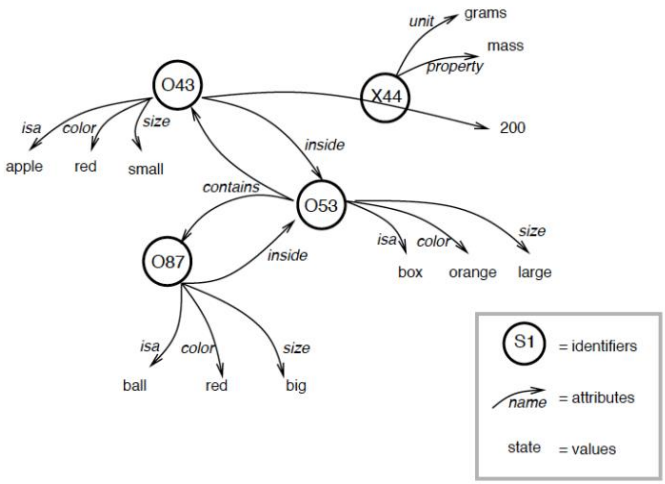
Tutto quello ciclo sarà dicento mille secondi .10 riguarda congntive band.



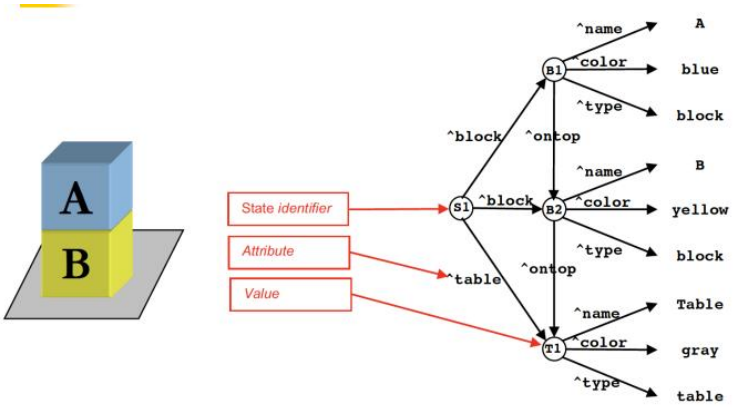
Lecture 5 - Maggio 18

Working Memory in Soar

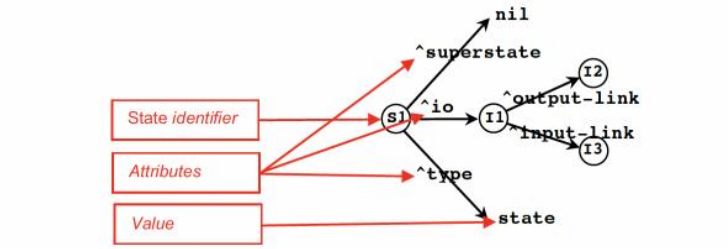
ha la rappresentazione interamente simbolico abbiamo sempre degli stati che sono assegnati automaticamente da Soar. quindi il nome degli stati è assegnato in automatico. Alcuni degli stati fungono da identificatori nel senso che sono identificatori di un insieme di elementi che caratterizzano quello stato. E gli Stati identificatori hanno anche degli attributi, abbiamo dei valori che possono assumere agli attributi.



Lo stato S1 viene sempre attribuito agli elementi della working memory di soar in automatico perché per fare in modo che l'intero ciclo decisionale parta in soar si fa una un'assunzione di esistenza dell'agente di solito questo qui è il primo statement che in automatico viene inserito all'interno della working memory.



questo tipo di rappresentazione che è più legata allo stato di cose in sé c'è anche una sorta di **meta rappresentazione** degli Stati che ha questo tipo di struttura qua anche questa viene generata in automatico.

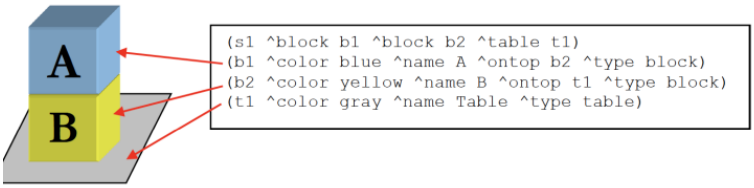



```
S1 ^superstate nil
S1 ^io I1
S1 ^type state
I1 ^output-link I2
I1 ^input-link I3
```

5 statement qui in basso sono degli statement che ci troviamo all'interno della memoria di lavoro. tutti questi elementi che sono rappresentati simbolicamente sono anche detti chunk.

Quando vogliamo costruire un agente Soar dobbiamo tenere traccia di qual è il rapporto tra Stati e sottostati che sono gestiti dall'architettura. quando utilizziamo un'architettura di questo tipo non abbiamo la completa libertà di fare tutto quello che vogliamo nel senso che alcune fasi di processing sono già Built-in dell'architettura quando dobbiamo andare a creare un'agente con quell'architettura significa che rispettiamo quelli che sono i commitment che sono stati inseriti all'interno di questo tipo di middleware,

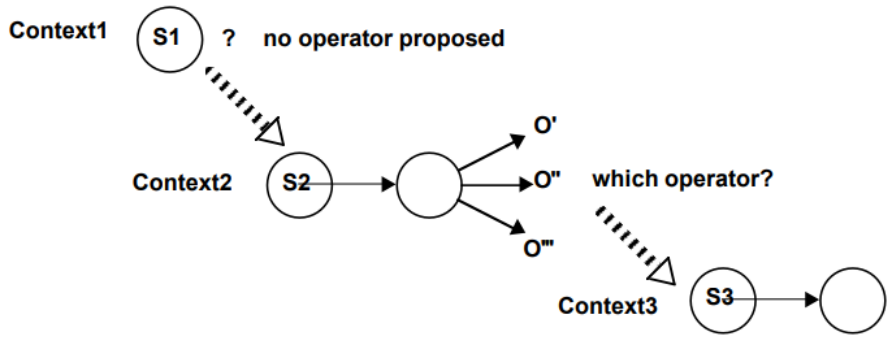
Un tipo della conoscenza modificata in working memory



Risolvere delle impasse - Learning

uno dei criteri fondamentali che ha a che fare con il modo in cui apprendono gli agenti di questa architettura cognitiva è basato sull'idea di risolvere delle impasse. Ci sono diverse strategie, però la strategia principale è data dal fatto di aggiungere nuova conoscenza sul mondo o su come devono essere utilizzati gli operatori. una volta che Soar riesce a risolvere un problema che non sapeva risolvere precedentemente parte una questa procedura di apprendimento che si chiama **chunking**, vuole dire la capacità di apprendere una regola. per esempio, io non so come andare da qui al terzo piano quindi inizialmente farò una serie di prove di per tentativi, una volta che ho imparato come andare al terzo piano la volta successiva che ci devo andare io non rifarò tutto il ciclo decisionale ma applicherò quel chunk di regole quindi quella regola che mi dice se devi andare al terzo piano magari esci fuori e poi prendi l'ascensore.

Ci sono diversi impassi in Soar e la risoluzione degli impassi passa attraverso la procedura degli Universal Subgoaling che passa mediante la procedura di c creazione di un sotto goal che è facilmente raggiungibile rispetto allo stato del problema in cui mi trovo.



In questo modo che è abbastanza intuitivo immaginiamo di essere una situazione del problema di tipo S1 e ci troviamo in una condizione in cui non c'è nessun operatore che proposto. viene automaticamente creato all'interno della dell'architettura un sottostato di tipo S2 che viene creato un sotto gol rispetto al goal iniziale che è un nuovo stato all'interno di questa struttura simbolica della working memory e in questo caso il goal di S1 e il goal a partire dallo stato 1 è quello di andare all'interno dello stato S2 per riuscire a selezionare quelli che sono gli operatori applicabili all'interno questo nuovo stato del problema. Questa procedura viene applicata ricorsivamente ad ogni livello.

Tipi di impasse

no change impasse in cui praticamente io non ho nessun operatore candidato per risolvere lo stato in cui la agente si trova non ho nessun operatore che posso applicare. Quindi viene fatta subgoaling.

Seconda possibilità io ho troppi operatori che invece possono essere applicati allo stato in cui mi trovo questo qui si chiama tecnicamente **Tie Impasse**.

Impasse abbastanza frequente così che può capitare in fase di modellazione è **un conflitto tra preferenze** cioè io ho diversi operatori che posso scegliere che però sono in conflitto in qualche modo tra di loro.

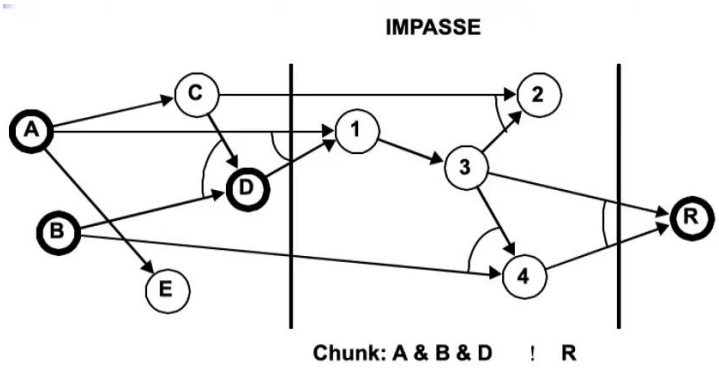
Tutte le tre impasse riguardano sostanzialmente la mancanza della conoscenza relativa agli operatori. il punto e cercare di andare ad acquisire un po' di conoscenze.

Chunking

Una volta che Impasse viene attivata la modalità di Learning che si chiama Chunking che si tratta l'apprendimento di una regola per risolvere un problema. una regola se nella rappresentazione simbolica di Soar è un chunk perciò si chiama Chunking. quindi è un meccanismo per cui io imparo una regola e la prossima volta che mi ritrovo il nuovo stato che ha la stessa situazione delle statiche o precedentemente risolto io non vado a rifare il ciclo di esecuzione di SOAR ma applica la regola che nel frattempo è stata spostata automaticamente nella memoria procedurale e la vada a applicare.

Questa procedura di chunking viene appresa da soar tramite un meccanismo di **backtracing** cioè immaginiamo di avere una sequenza di stati che ha portato tramite l'applicazione di diversi tipi di operatori ad un risultato che ci fa superare l'impasse in cui vi eravate trovati nello stato iniziale.

Una volta che è stato superato questo tipo di impasse Soar fa **backtracing** cioè vede qual è il percorso che è stato, quali sono gli Stati e gli operatori che sono stati applicati e poi li mette in una regola e va a finire all'interno della memoria procedurale.



Esempio di un agente: Hello Word

La sttruttura dati di sintassi di Soar è IF-THEN. Per scrivere questo tipo di programma la regola che deve essere attivata in soar è così:

(if) se io esisto (agente) allora (Then) devi fare un print Hello Word e ti devi fermare.

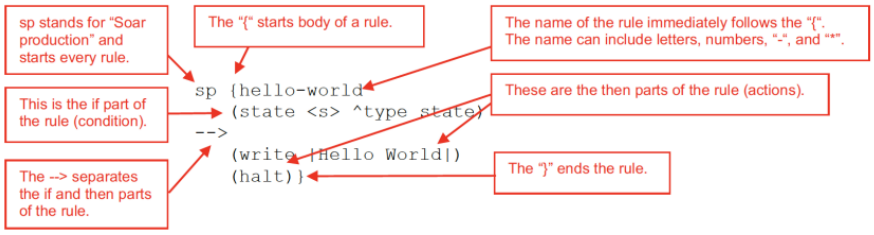
Soar lavora testando la “if” parte (LHS rule, dette anche condizioni)e La parte “then” (allora) della regola è una azione. una regola cambia qualcosa all'interno della working memory cioè lo Stati della working memory subiscono un qualche tipo di cambiamento. in generale l'elemento base per costruire un buon programma è fare knowledge level analysis. La struttura a grafo della memoria di lavoro di SOAR è creata da regole o da sensori in modo automatico.

print “Hello Word”

La sintassi di SOAR per scrivere regole

Ci sono due modalità che possono essere utilizzate per andare a scrivere una regola. Prima regola di e scrivere una regola in soar è una espressione che si chiama SOAR Production (sp) che ogni SP viene attivata sostanzialmente con l'utilizzo di parentesi graffe che si aprono e poi si chiudono alla fine della regola quindi alla fine del then.

prima cosa che bisogna inserire è sostanzialmente il nome della regola quindi qual è il nome della regola che voglio applicare.



per andare ad applicare la regola devo andare ad inserire una prima condizione cioè **la condizione di inizializzazione** del sistema che dice sostanzialmente che se c'è uno stato iniziale <s> che è un tipo di Stato allora p posso andare a fare un, ad esempio, write di questa sequenza di caratteri. una volta che ho fatto questo mi devo fermare attraverso il comando halt.

Quindi il template per costruzione delle regole in modo semplice è come la sintassi seguente.

I nomi delle regole non possono essere una lettera seguita da numeri (es. A12), perché questi sono dei nomi di default che soar assegna agli stati che crea in automatico durante l'elaborazione.

Per avviare un programma in soar debugger possiamo caricare il rule file all'interno di Soar Debbuger. Tramite il commando Run, Quanto ogni agente è creato c'è una inizializzazione di tipo s1 ^type state) nella working memory. Vuol dire "esiste l'agente" (che è la condizione prima per la realizzazione di qualunque regola). A questo punto è possibile far scrivere ad un agente SOAR vari messaggi. Per scrivere un agente SOAR è possibile usare

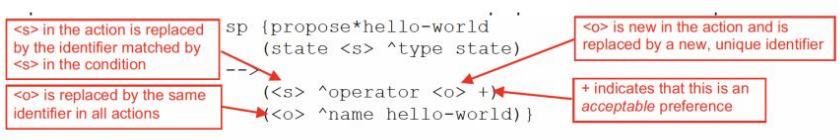
Hello word con operatore

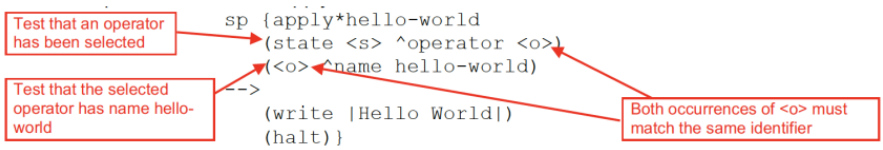
Possiamo provare a scrivere lo stesso programma utilizzando due regole d'accordo allora in pratica una prima regola serve per proporre l'operatore. Secondo regola ci serve per applicare quella regola.

```
Propose*hello-world:
If I exist, propose the hello-world operator.

Apply*hello-world:
If the hello-world operator is selected, write "Hello World" and halt.
```

questa regola ci proporre una regola che si chiama hello-word e poi se esiste questa regola allora (then) esiste un operatore che si chiama hello-word.





P.S. Usare comando “excise —all” per ripulire regole precedenti.

Esempio: hungry – thirsty I

- Un robot può solo mangiare e bere. Inizialmente è affamato e assetato. Goal: fare in modo che non lo sia più
- Lo stato ha 2 attributi
  - hungry, con possibili valori yes and no
  - thirsty, con possibili valori yes and no
- Nello stato iniziale il robot è sia hungry che thirsty, quindi abbiamo **(<s> ^hungry yes ^thirsty yes)**.
- Il goal (o lo stato desiderato) da raggiungere è **(<s> ^hungry no)**.
- Operatori possono essere chiamati Eat and Drink:
  - Eat si può applicare ad ogni stato **(^hungry yes)**, e produce un nuovo stato con **(^hungry no)**
  - Drink si può applicare ad ogni stato **(^thirsty yes)**, e produce un nuovo stato con **(^thirsty no)**.
  - Ogni operatore fornisce conoscenza in termini di produzioni relative a:
    - quando proporre l’operatore
    - come applicarlo

secondo knowledge level analysis dobbiamo provare a vedere quali sono lo stato di partenza e goal. la cosa da provare a fare e vedere quali sono gli elementi di conoscenza

che ci servono quali sono gli operatori che dobbiamo andare ad utilizzare e quando possono essere utilizzati questi tipi di operatori per riuscire a risolvere questo tipo di problema.

Lecture 6 - 19 Maggio

Ogni agente di soar hanno un stato iniziale S che di default deve sempre essere inizializzato per gli agenti Soar questo stato non è uno stato di impasse naturalmente ed è uno stato che non ha nessun superstato.

se questa produzione è vera allora andiamo a dare tutta una serie di informazioni su quello che è diciamo lo stato di cose. Il nome di stato può essere un nome del tipo nome di programma vogliamo che il nostro stato abbia come proprietà un problem Space e poi attacchiamo con a questo problem Space anche quello che è uno stato desiderato cioè qual è il goal che vogliamo raggiungere.

infatti c'è questa proprietà desired a cui è associata questa la variabile di tipo *d* che è in un certo senso specificata nella penultima riga di questa regola che ci dice la variabile di è un qualcosa che ci deve portare ad una situazione in cui la gente non è più affamato cioè in cui la gente non è più hungary.

La prima della prima regola nella parte then vogliamo dare un nome a problema Space che chiamiamo hungary-thirsty e poi a settare l'ultima riga vado a settare lo stato iniziale cioè quella in cui noi abbiamo il nostro agente che sia affamato si assetato.

```
###     I.      Propose the top space
###
### The code in this section proposes a simple space to work in, and a
### simple state to start working in.

sp {ht*propose-space*ht
  (state <s> -^impasse ^superstate nil)
  -->
  (<s> ^name ht-state)
  (<s> ^problem-space <p> ^desired <d>)
  (<p> ^name hungry-thirsty)
  (<d> ^hungry no)
  (<s> ^thirsty yes ^hungry yes)
}
```

dopodiché dobbiamo provare ad inizializzare a cascata le caratteristiche dello stato iniziale cioe definiamo un rule che si chiama inizializza stato hungray-thirsty, **se** ci troviamo in uno stato estero che ha un problem Space name che si chiama hungray-thirsty che è vero perché lo abbiamo settato nella regola precedente **allora** hungray-thirsty sarà il nome del nostro stato che sarà esattamente lo stato di inizializzazione che abbiamo settato nella regola di produzione precedente.

```
## Initialise the top state
## simple way to set up the top-state with a separate production
##sp {ht*init-state*ht
##  (state <s> ^problem-space.name hungry-thirsty)
##  -->
##  (<s> ^name ht-state)
##  (<s> ^thirsty yes ^hungry yes)}
```

dopodiché andiamo a definire quali sono gli operatori che entrano in gioco all'interno di tipo di problema. In questo caso abbiamo due operatori.

Nella prima produzione vado a proporre l'operatore eat e gli diciamo che se ci troviamo nel nostro problem Space che si chiamano anche hungray thirsty e se ci troviamo nella situazione in cui la gente è hungray allora possiamo andare ad utilizzare un operatore o che si chiama eat, la stessa cosa vale per definire operatore drink.

```
## Propose eat.
sp {ht*propose-op*eat
  (state <s> ^problem-space.name hungry-thirsty )
  (<s> ^hungry yes)
  -->
  (<s> ^operator <o>)
  (<o> ^name eat)}

## Propose drink.
sp {ht*propose-op*drink
  (state <s> ^problem-space.name hungry-thirsty )
  (<s> ^thirsty yes)
  -->
  (<s> ^operator <o>)
  (<o> ^name drink)}
```

in realtà in questo programma abbiamo inserito nello stato desiderato nella prima regola di produzione il fatto che vogliamo un agente che non sia più affamato, quindi questo significa che noi dobbiamo andare a specificare una preferenza tra gli operatori, allora nel caso se ci trovassimo nella condizione in cui abbiamo la gente che ha affamato ed assetato e deve arrivare al goal desiderato in cui non è né più affamato né più assetato non c'è una preferenza tra gli operatori. in questo caso qua però siamo andati a specificare una regola desired goal che è quello in cui la gente non sia più affamato. non è che non sia più assetato quindi abbiamo una specifica maggiore perché ci permette di vedere come andare a specificare delle preferenze tra gli operatori.

```
## Eat is better if you are hungry and want not to be
sp {ht*compare*eat*better*drink
  (state <s> ^desired <d> ^problem-space.name hungry-thirsty)
  (<s> ^hungry yes)
  (<d> ^hungry no)
  (<s> ^operator <op-eat> +)
  (<op-eat> ^name eat)
  (<s> ^operator <op-drink> +)
  (<op-drink> ^name drink)
  -->
  (<s> ^operator <op-eat> > <op-drink>)}
```

questo pezzetto del codice ha lo scopo di confrontare quelli che sono i due operatori che abbiamo definito prima. i due operatori Eat e Drink allora quali sono le condizioni?

se ci troviamo in uno stato <s> che ha come goal un certo tipo di Stato desiderato di che si trova in un problem Space che richiamando i thirsty e questo stato S ah l'attributo hungry che è settato a yes mentre invece l'attributo hungry dello Stato goal di è settato a **no** perché noi vogliamo arrivare al goal di cui il nostro agente non è più affamato, e se andiamo ad associare al nostro stato iniziale S l'operatore eat con un simbolo + che significa che è un operatore accettabile per il nostro agente.

Poi se andiamo ad associare al sempre al nostro stato un operatore drink con la condizione che è già menzionata, possiamo andare a creare una preferenza tra operatori in cui l'operatore eat è maggiore di operatore drink. questo è un modo per andare a settare le preferenze tra gli operatori.

Sintassi delle preferenze degli operatori in Soar

RHS preferences	Semantics
(id ^operator value)	acceptable
(id ^operator value +)	acceptable
(id ^operator value !)	require
(id ^operator value ~)	prohibit
(id ^operator value -)	reject
(id ^operator value > value2)	better
(id ^operator value < value2)	worse
(id ^operator value >)	best
(id ^operator value <)	worst
(id ^operator value =)	unary indifferent
(id ^operator value = value2)	binary indifferent
(id ^operator value = number)	numeric indifferent

Quando non viene specificata una preferenza diretta all'interno delle regole quello che succede è che SOAR sceglieranno random.

una volta che abbiamo inizializzato lo stato iniziale del problema specificato qual è il goal del nostro problema questo abbiamo fatto nella prima produzione poi abbiamo introdotto gli operatori e le Condizioni di utilizzo degli operatori e quando li applico.

ma quando applico drink poi abbiamo fatto un'altra cosa abbiamo detto che e quando andiamo ad applicare drink allo stato iniziale del problema quello che vogliamo e andare a specificare una preferenza per eat rispetto al all'operatore. una volta che abbiamo fatto tutte queste cose qua finisce la fase di proposta in cui abbiamo sostanzialmente dichiarato quelli che sono gli elementi di conoscenza che servono alla gente per svolgere il proprio problema.

una volta che questa fase di proposta e finita la fase successiva del ciclo di soar è applicare gli operatori perché noi abbiamo già indicato la fase di proposta la fase di valutazione perché abbiamo detto chi deve venire prima rispetto ad un altro. ora ci resta applicare questi operatori.

Ora dobbiamo andare a scrivere delle regole che ci permette di applicare gli operatori. Andiamo a dire per applicare soar production. Alla fine attraverso la regola di terminate dobbiamo in qualche maniera terminare l'operatore.

```
sp {ht*apply-op*eat
   (state <s>  ^operator <o>)
   (<o> ^name eat)
   (<s> ^hungry yes)
   -->
   (write (crlf) |          chomp chomp... |)
   (<s> ^hungry yes - no +)}

sp {ht*terminate*eat
   (state <s> ^operator <o>)
   (<o> ^name eat)
   (<s> ^hungry no)
   -->
   (<s> ^operator <o> @)}
```

l'ultima parte di questo mini programma ci deve dire non soltanto quando terminare l'uso per gli operatori ma quando andare a terminare in generale il programma quindi tutto il ciclo o di routine dell'agente Soar è molto facile andare a definire perché anche qui abbiamo una regola di produzione che possiamo scrivere in cui se ci troviamo in uno stato S e che ha come attributo **desierd d** quindi se lo stato di **desierd** va bene e in qualche

modo anche <e> che si è ottenuto in seguito all'applicazione all'interno dei diversi operatori, se lo stato di **desierd** che abbiamo ottenuto è in qualche modo *compliant* cioè rispetta quella che è la situazione desiderata specificata nella prima regola all'interno dello Stato **d** in questo caso avremo che d ha come Hungray valore di **no** che possiamo anche lasciare così <val> perché in questo caso prende la variabile specificata inizialmente di **d**.

```
### This code terminates the problem solving when the goal is reached.


## How to tell if you can stop
sp {ht*evaluate*state*success
  (state <s> ^desired <d> )
  (<d> ^hungry <val>)
  (<s> ^hungry <val>)
  -->
  (<s> ^success <d>)}


## One of the default rules is brought in to notice that we are
## finished.  (Slightly modified to be more compact and less general.)

sp {default*top-goal*halt*state*success
    :default
  (state <s>  ^desired <eb>)
  (<s> ^success <eb>)
  -->
  (write (crlf)  |  goal for  | <s>  | achieved | )
  (halt)}
```

La parte salt è la regola che imposta, bene il programma in qualche maniera è terminato.

Reinforment Learning

L’idea è di utilizzare procedura di Rewards per fare appredimento di Agenti in modo semplificata abbiamo dei tipi di reward che può avere un'agente in base alle azioni che fa rispetto ad un ambiente esterno, Rewards sono:

1- Intrinisc Reward
2- Extrinsic Reward

l'idea di assegnare dei reward per andare a creare degli agenti generali di tipo intelligente. in realtà cio di cui abbiamo bisogno appunto per creare una DLI. [Reward Is Enough]

Cani di Pavlol

L'idea dell'apprendimento per rinforzo nasce sostanzialmente in ambito neuro fisiologico. a questi cani veniva dato del cibo in alcuni momenti della giornata quando a questi cani veniva portato del cibo contemporaneamente veniva utilizzato un campanello; quindi, sostanzialmente i cani andavano ad apprendere questa sorta di connessione tra il fatto che suonava il campanello vuole dire arrivare cibo. quindi all'inizio ovviamente non avevano questo tipo di conoscenza, successivamente in base alle esperienze ripetute

avevano associato a questo segnale per il tintinnio di un campanello il fatto che doveva in qualche modo arrivare del cibo e questo tipo di associazione è un esempio di apprendimento per rinforzo.

Da questi studi di pavlov nasce tutta una corrente che si chiama una corrente di si chiama bioritmo. è una corrente che nasce insomma soprattutto negli anni 30 40 del secolo scorso che riprende studi di pablov e cerca di dire che allora tutto cio di cui abbiamo bisogno per controllare potenzialmente il comportamento non solo di una persona ma anche di masse di persone è apprendere quali tipi di associazioni, perché naturalmente questi cani apprendevano le associazioni che venivano rinforzate e invece non apprendevano o erano appunto non rifacevano i comportamenti che invece subivano delle punizioni.

Pavlov osservava come questo apprendimento per rinforzo si avesse quando uno dei due stimoli mancava cioè una volta che era stato appreso questa connessione allo sperimentatore bastava andare a suonare la campanella per fare in modo che il cane salvasse.

Ma Non considera per niente quelli che sono gli aspetti cognitivi nel senso che all'interno del bioritmo classico ciò che importa sono gli stimoli di reward oppure di punizione che può subire un agente per poterne determinare il comportamento così intelligente o meno intelligente.

[Yann LeCun] l'idea di avere degli agenti basati soltanto sul learning puro sia soltanto una sorta di ciliegina su questa che lui chiama learning Cake cioè quando parliamo di apprendimento anche all'interno di sistemi di intelligenza una grossa parte della torta è data da quella che è tutta la parte di apprendimento unsupervised, la parte di apprendimento non supervisionato che anche la parte su cui abbiamo dei sistemi che sono abbastanza scadenti perché tutti i sistemi di oggi in cui si parla di Deep learning machine learning eccetera sono perlopiù invece i sistemi supervised. sono sistemi in cui

c'è un addestratore umano e ci sono numerosissimi esempi che vengono utilizzati da questi tipi di approcci per poter fare bene una categorizzazione però il rinforzamento detto che sarebbe soltanto una parte minima di questa di questa torta.

Agente Left-Right

Agente che sceglie di muoversi a sinistra o a destra. Dopo aver deciso la sua destinazione riceve un “reward,” o feedback In questo caso: -1 per movimento a sinistra +1 for destra. Usando RL, l’agente imparerà a muoversi a destra.

Lecture 7 - 24 Maggio

Spiegazione

ci sono diversi tipi di spiegazione che possiamo utilizzare e tutta la letteratura sui tipi di spiegazione che non viene dal dall'informatica ma viene dalla filosofia della scienza che cerca in che modo un modello ha un potere esplicativo rispetto ad un fenomeno che appunto va modellare in tutti gli ambienti.

Ora esiste una classificazione abbastanza standard che viene data che grosso modo si può identificare con queste queste classi.

- Teleologica
- Meccanicista
- Evoluzionistica
- Nomologico-deduttiva
- Funzionalista

Qualsiasi spiegazioni hanno due parti che sono

- **Explanandum** che è la cosa da spiegare
- **Explanans** è l'elemento che fornisce la spiegazione

Se devo provare a spiegare perché ho la febbre questo caso Explanandum quindi la cosa che devo spiegare e il fatto che io abbia la febbre e Explanans può essere la risposta fornita dal medico.

ipotizziamo di voler in qualche modo prendere come spunto da questo fenomeno naturale che è dato dal fatto che i camaleonti cambiano colore che è in risposta a diversi tipi di stimoli, per esempio, cambiano colore se avvertono che c'è un predatore o se ci sono potenziali partner. ci sono Diversi tipi di spiegazione una prima spiegazione può essere i camaleonti cambiano colore per mimetizzarsi e per sfuggire ai predatori. oppure se volessi chiedere per quale motivo utilizza dei colori diversi a seconda dei predatori se predatore è un uccello si sa che il colore del camaleonte diventa più simile allo sfondo. perché gli uccelli di solito hanno mai una vista migliore rispetto tanti tipo di predatori che si trovano nell'ambiente quindi in questi casi quello che viene fornita è una spiegazione che si chiama teleologicache significa scopo cioè io cerco di spiegare per quale motivo accade un fenomeno, cercando di mettere in evidenza qual è lo scopo ultimo che questo fenomeno ha rispetto, per esempio, alla sopravvivenza del nostro camaleonte.

Per rispondere a queste domande, io dico lo scopo di questo comportamento è cercare di non farsi catturare. in qualche modo vado ad attribuire elemento che voglio spiegare quindi al fenomeno cioè io voglio spiegare una intenzionalità perché ha un scopo.

Però questo tipo di spiegazione ci permette veramente di comprendere perché il loro cambiano colore? non ci permette veramente di capire quali sono i fattori che determinano il cambiamento cromatico.

Teleologica

E' possibile assegnare spiegazioni teleologiche per interpretare l'output di un sistema artificiale

Si perché noi attribuiamo con la spiegazione teleologica una qualche intenzionalità sia i sistemi naturali ma artificiali e lo facciamo continuamente perché è un tipo di spiegazione più comoda e veloce. il concetto che è stato introdotto da Bennet di Intentional stance che vuole dire "attribuire intenzionalità al comportamento/output prodotto da un sistema"

E' una buona strategia in ottica esplicativa?

No, non è una buona strategia esplicativa perché in qualche modo è un'assunzione di sistemi che non hanno intenzionalità.

Esempio dei bambini, loro attribuiscono intenzionalità a molte cose rispetto a quelle che si imparano da grande. perché questo giocattolo è caduto? perché voleva andare a posto X o y.

l'attribuzione di un qualcosa di automatico che noi facciamo e poi apprendiamo successivamente corso delle fasi evolutive ad applicare selettivamente che un primo tipo di spiegazione che possiamo utilizzare quando non ne abbiamo disposizione perché noi la applichiamo anche all'interpretazione di un sistema speciale quando veramente non sappiamo che cosa sta effettivamente succedendo quindi è un processo di spiegazione in cui noi come degli osservatori esterni rispetto ad un oggetto che voltiamo naturale o artificiale, proviamo a fornire un modello di spiegazione di quel tipo di comportamento.

Si può essere applicabile in alcuni sistemi biologici tipo **le blatte** oppure altri tipi di microrganismi che utilizzano come riflesso quello di andare in automatico verso fonti di luce.

Se noi spiegassimo teleologicamente questo tipo di comportamento che potremmo dire a vogliamo andare verso la fonte di luce perché magari lì c'è del cibo che vogliono raggiungere. però questo sarebbe un errore cioè attribuire un goal anche ad agenti che

non hanno goal e non hanno intenzionalità che reagiscono soltanto per riflessi tipo da cani di pavlov.

Meccanicista

“il cambiamento di colore dei camaleonti è dovuto alla risposta di alcune cellule del derma dell’animale (i cromatofori) a stimoli nervosi e/o endocrini” => spiegazione meccanicistica

si cerca di rispondere di quali sono i meccanismi che determinano i fattori che determinano un certo tipo di fenomeno. A differenza della spiegazione teleologica non c’è riferimento ad alcuno “scopo” da raggiungere (fuga dai predatori) ma si cercano di spiegare i meccanismi che causano un fenomeno F (i meccanismi possono essere suddivisi in parti). Vuole vedere quali sono gli elementi in gioco e le componenti di questo sistema che alla fine vuole dire camaleonte è composto da parti che interagiscono tra di loro e ci sono dei meccanismi che cercano di spiegare i meccanismi che determinano questo tipo di mutazione.

questo è un buon punto di partenza perché ci permette di fare delle ipotesi su quelli che sono i processi che sono coinvolti all'interno di un fenomeno. Un sistema strutturalista usa questa spiegazione. Perché vuole fornire delle spiegazioni meccanicistiche che io voglio costruire un modello artificiale strutturale, perché voglio andare a scoprire dei meccanismi che nel modello naturale preso come fonte di ispirazione non erano noti.

in questa spiegazione ci sono tipi di spiegazioni che sono **le spiegazioni causali** in cui non soltanto si vanno a vedere quali sono i fattori che influenzano un fenomeno.

Le spiegazione meccanistica vs causale

tutte le spiegazioni causali sono **meccanistiche** perché spiegano effettivamente quelli che sono i meccanismi che causano un fenomeno. però non tutte le spiegazioni meccanicistiche sono causale perché io potrei avere una spiegazione probabilistica di un fenomeno che è quello che ho nella nostra grande maggioranza dei causi senza essere in grado di riuscire effettivamente a determinare quali sono le cause. determinare le cause significa quelle sono delle leggi perché valgono sempre rispetto alla conoscenza del mondo.

Evoluzionistica

“perché colore diversi per uccello/serpente”? Risposta: “il numero di predatori volatili nell’habitat dei camaleonti è maggiore rispetto a quello di altri animali e ciò impone una maggiore pressione selettiva rispetto alla capacità mimetica”

Una Spiegazione a questo tipo di domanda è perché c'è stata la necessità evolutiva da parte di questo agente biologico. Fa riferimento alla storia evolutiva: specializzazione mimetica come prodotto di meccanismi di selezione.

La differenza di questa spiegazione è legata all'interpretazione della domanda: nelle prime 2 spiegazioni si cerca di spiegare perché si manifesta un fenomeno (descrivono i meccanismi o i fini che determinano F); in questa si cerca di spiegare perché membri di una certa classe X "posseggono" le capacità per produrre un certo fenomeno F.

Funzionalista

Cerca di dare una spiegazione dell'output di un sistema sulla base delle componenti che lo caratterizzano (che "funzionano come" componenti analoghe in un sistema biologico). Il problema è che I modelli computazionali che incorporano spiegazioni funzionali spiegano le capacità di un sistema in termini di sottocapacità. Ma questa spiegazione è data dalle ipotesi incorporate nel modello, non dai calcoli eseguiti dal modello sulla base delle ipotesi. Ovviamente questa spiegazione cerca di dare delle spiegazioni per ogni sistema funzionalisti di IA. Spiega output di un sistema facendo riferimento ai sottocomponenti di questo sistema che replicano funzionalmente alcuni elementi che vogliamo andare a modellare all'interno del nostro sistema artificiale.

assunzioni in qualche modo vengono fatti quando si costruiscono, i suoi modelli sono già note quindi spiegare un modello sulla base delle assunzioni della costruzione funzionalista non è aiuto cioè non ci permette di scoprire quali sono i meccanismi che determinano quel tipo di output.

buone spiegazioni

Per costruire una buona spiegazione:

- Sns: Explanans presenta condizioni necessarie e sufficienti rispetto all'explanandum; ↔
- Sn: Explanans presenta condizioni necessarie rispetto all'explanandum
- Ss: Explanans presenta condizioni sufficienti rispetto all'explanandum

Deve essere legato ai concetti di Causalità e Previsione. In una buona spiegazione l'explanans esprime le cause (NS, N o S) per l'explanandum e/o dovrebbe permettere di prevederlo. Se la spiegazione di un fenomeno è intimamente legata alla ricerca delle "cause" allora le spiegazioni teleologiche perdono qualche punto a favore di quelle meccanicistiche.

Es. la stimolazione degli organi del camaleonte causa l'avvio di certi meccanismi che causano variazioni di concentrazione dei pigmenti nei cromatofori che causano il cambiamento di colore. Nel caso teleologico non si può dire che il cambiamento di colore sia causato dal fine di sopravvivere ai predatori.

Anche rispetto all'aspetto predittivo le spiegazioni meccanicistiche hanno qualche vantaggio (conoscendo i meccanismi di F posso predire il fenomeno). Nel caso teleologico conoscendo il fine che determina F non possiamo fare predizioni accurate su F (il camaleonte potrebbe anche scegliere una strategia alternativa).

Visione Pluralistica

Per Spiegare dei fenomeni complessi l'idea è che possano coesistere tipi e diversi livelli di spiegazione per una posizione che è nota come Pluralismo esplicativo. per spiegare il fenomeno io posso utilizzare diversi tipi di **perché** e posso prendere spunto dal repertorio della filosofia della scienza per andare ad applicare i diversi tipi di spiegazione anche quello che è l'output del mio sistema.

Nomologico-Deduttiva

Questa spiegazione è una derivazione della euristica cioè si riesce a spiegare certo tipo di fenomeno F perché in qualche modo si va ad arrivare logicamente a quelli che sono cioè explanandum che è l'elemento che permette di spiegare deduttivamente il fenomeno che riguarda Explanans.

nella stragrande maggioranza della scienza di oggi non è possibile avere questo tipo di spiegazione perché alla domanda qual è un buon tipo di spiegazione perché bisogna provare a cercare di ottenere una spiegazione logico deduttivo. a parte per casi particolari non è possibile riuscire ad ottenere una conoscenza completa di un fenomeno che ci permette poi di fare queste derivazioni e quindi di fatto i tipi di spiegazioni che vengono utilizzati in contemporanea sono tutte quelle che abbiamo visto che si prova ad utilizzare queste di questi diversi tipi di spiegazione meccanicistica ed etc per cercare di interpretare quello che è output di un certo tipo di sistema.

Sistemi intelligenti di tocur quella funzionalista sono quelli che cercano di risolvere un problema disinteressandosi completamente del modo in cui altri agenti biologici risolvono lo stesso tipo di problema.

Lecture 8 – 25 Maggio

Deep Mining

hanno lanciato il claim che utilizzano la psicologia per capire quello che è il funzionamento delle reti neurali in un compito di classificazione di immagini quindi abbiamo una rete neurale che va a classificare immagine che addestrata su image net che una posso dare a base di immagine che viene utilizzato per molte competizioni anche per algoritmi di Computer Vision

il tipo di grounding che questi ricercatori hanno fatto rispetto alle teorie che abbiamo la psicologia cognitiva che è stato fatto nell'ambito della psicologia dello sviluppo è il tema in questa area enorme della psicologia è stato come i bambini apprendono a riconoscere e categorizzare degli oggetti in che modo i bambini imparano a dirò che questa è una sedia o etc. l'idea è quella di partire da questi tipi di teorie per andare ad ascrivere all'output di queste reti profonde quelle che sono dell'interpretazione rispetto a perché è stato generato un certo tipo di output.

alla parte di teoria di riferimento di questi ricercatori di informatica che appuntano su questo sistema di computer vision si rifà questa teoria che si chiama Gavagai. È una sorta di esperimento mentale che è stato ideato da acquain che insomma era filosofo del linguaggio che si pone questa domanda qua: immaginate di trovarvi in una parte sperduta della Papuasia in cui non conoscete il linguaggio locale. a un certo punto compare all'interno della vostra scena un qualcosa come un coniglio sentite che le persone della tribù locale lo chiamano Gavagai. cioè ogni volta che compare questo tipo di oggetto viene chiamato questo oggetto Gavagai quindi avete siete esposti a questo tipo di segnale. è una situazione simile a quella in cui si trovano i bambini quando devono apprendere una nuova etichetta per un nuovo oggetto perché loro non conoscono la lingua però devono effettuare questo grounding. A che cosa noi assegniamoil grounding di questa etichetta all'intero oggetto che vediamo all'intero coniglio oppure ad alcune sue parti, per esempio, alle orecchie eccetera oppure ancora un associamo questa etichetta Gavagai a tutte le cose che hanno la stessa forma del coniglio. Quali sono le varianze che ci permette di etichettare diversi oggetti?

Vengono fuori delle bias Nel senso che innanzitutto un primo bias è quello del

whole-object bias

noi non assegniamo delle etichette a fare di un oggetto ma ad un oggetto che comprendiamo nella sua interezza. questa buona roba simile a triangolo di canias. noi

abbiamo questa tendenza automatica a completare un accordo e generalizzare e quindi assegnare un'etichetta. in questo caso ad un intero oggetto secondo

Shape Bias

cioè sembra che per esempio la forma dell'oggetto sia un elemento che noi utilizziamo per apprendere e associare queste etichette quando vediamo delle nuove delle nuove istanze è elemento importante. Quando vediamo un oggetto allo shape di Gavagai diciamo che quello è Gavagai. pare che l'elemento principale che ci permetta di fare la categorizzazione e lo shape bias.

Secondo questo bias hanno creato delle reti neurali dove hanno mischiato diversi tipi di reti neurali profondi quindi per esempio una rete neurale conclusionale che si utilizza molto per fare task di classificazione di immagini. un'altra tipo di rete neurale che si chiama matching network che è una versione che permette è ottimizzata per fare one shot learning significa apprendimento basato su uno oppure su pochi esempi. sostanzialmente loro hanno presentato una rete con un oggetto probe che ha la forma come ferro di cavallo e a colore verdino. hanno mostrato alla rete un oggetto A un oggetto che è shape match object cioè che ha la stessa shape dell'elemento a probe con colore diverso. Un oggetto B che ha lo stesso colore a l'oggetto prob però con un tipo diverso.

Hanno Misurato shape come il numero delle volte in cui all'immagine probe viene assegnata sostanzialmente l'etichetta quindi si vede quante volte questo probe è stato associato alla categoria rispetto alla categoria come fa explanatory finding. loro hanno detto che questa rete hanno un shape bias come esseri umani. Conclusione è che Capacità di tool che vengono dalla psicologia cognitiva rendono esplicite quelle che sono le proprietà computazionali implicite, per esempio, delle reti neurali profonde.

Noi esseri umani nel nostro network quando vediamo nuovo elemento da classificare nelle classi A e B che sono caratterizzati da questi elementi va a preferire la classificazione A per rispetto alla classificazione per B.

Però il problema di questo ricerca è che non esiste nessun computational model del human word learning, Perché rete neurale è basato su questo concetto che si può essere come un computational model per human word learning. Qundi non viene supportato da evidenze.

I problemi di Deep Mining

le due architetture neurali sono completamente diverse, c'è l'architettura biologica del nostro cervello su cui si sanno delle cose rispetto al perché viene fornito. lo shape bias viene favorito rispetto ad altri tipi di tendenze, ci sono delle evidenze e spiegazioni

meccanicistiche pero non viene completamente considerato da loro perché loro considerano soltanto output.

sostanzialmente le rappresentazioni interne costruite sono completamente diverse cioè i pattern che in qualche modo vengono fra virgolette attivati all'interno delle due reti. perché ci sono dei tool di visualizzazione adesso che sono stati sviluppati per cercare di interpretare risultati di queste reti neurali non coincidono assolutamente con i pattern che in qualche modo sono stati trovati all'interno di una rete biologica.

Valutation

Ogni Test di macchina ha la Struttura di questo genere qu che si chiama **Behavioral Reportoire Argument** cioè la struttura che si segue

- un'entità viene considerata intelligente se è in grado di esibire un certo tipo di capacità x.
- un sistema artificiale può essere programmato per mostrare e esibire questa capacità
- quindi computer possono essere intelligenti

Obiezioni di questa struttura

la behavioristic

il fatto che un sistema è in grado di esibire uno stesso tipo di quelli che si dice behavior di un sistema naturale in qualche modo non significa che sia in grado di esibire le stesse capacità intellettive che nel sistema naturale sono alla base per l'esibizione di quel comportamento.

Others

Non è possibile che un computer possa fare tutto cioè possa esibire qualsiasi tipo di facoltà

Test di Turing

Un interrogatore che dialoga senza vedere chi c'è dall'altro lato o con un essere umano oppure con una macchina e il test è superato se l'interrogatore umano non è in grado di distinguere con chi sta dialogando. Si puo vedere questo test come un criterio operativo per attribuire intelligenza ad un sistema che significa attribuita la capacità di risolvere un singolo task e questo si scontra di nuovo col fatto che l'intelligenza e un elemento multi-fattore.

Il problema di Turing Test

il risultato di questo test e dipende molto da chi interroga quindi questo è un problema enorme perché è un testo soggettivo interrogatori diversi possono avere interpretazioni. Questo test non adatto per intelligenza generale perché è interamente basato sul linguaggio cioè è basata sulla competenza linguistica.

I suoi principali presunti inconvenienti sono probabilmente i seguenti:

è limitato al comportamento linguistico;

è affetta da sciovinismo umano (cioè è eccessivamente antropocentrica);

è influenzato da un pregiudizio comportamentale (cioè è limitato a prove comportamentali).

Perché linguaggio non è come altre facolta cognitive

Sperimeto è cosi che abbiamo una scatola di tipo rettangolare uin uno degli angoli di questa di questa scatola c'è del food e al centro di questa scatola si mette un topo prat e una volta che è stato inserito il topo al centro di questa scatola in una sorta di micro scatola rotonda all'interno della scatola quadrata si dà un leggero spin. poi si lascia il topo libero e si vede il topo dove va quindi è l'obiettivo di questo esperimento sarebbe per il topo è di andare nell'angolo dove c'è il cibo. Topi in generale quando sono sottoposti a questo tipo di esperimento con una distribuzione Equiprobabilità alcuni 50% vanno verso cibo alcuni 50% vanno verso contrario.

Hanno fatto la stessa cosa con gli esseri umani immaginate di andare a sostituire dei topi con dei bambini e il food con dei giocattoli e ovviamente invece di una scatola avete una stanza quadrata e al centro di questa stanza sono stati messi dei bambini anche questi bambini venivano spinti. quindi c'era uno spin che veniva dato esattamente come per i topi in questo caso i bambini si dirigono in modo equiprobabili lungo gli angoli all'interno della quale c'è in questo caso il giocattolo. Dopo aver messo i bambini è stato rifatto questo tipo di esperimento anche con persone adulte è stato visto che anche le persone adulte sottoposte a questo tipo di esperimento quindi messo in una stanza la prima cosa che fanno e dirigersi con anche i due angoli che si trovano della diagonale qui c'erano degli reward che venivano forniti però lo stesso esperimento ha dato gli stessi risultati per topi bambini in età pre-linguistica quindi i sotto l'anno e mezzo e adulti.

Esiste un'altra variazione che produce degli effetti diversi. iniziano ad esserci delle differenze tra queste tre diverse popolazioni quando per esempio all'interno della stanza in cui sono stati messi gli uomini e i bambini si dipinge un lato di blu dove c'è l'angolo in cui sono presenti anche le ricompense. quando c'è questo tipo di variazione succede che i topi continuano ad andare in modo equiprobabile o lungo quest'angolo o lungo

quest'altro angolo i bambini in età pre-linguistica continuano ad andare in modo come i topi però gli adulti una volta che percepiscono magari con la coda dell'occhio no che c'è questo tipo di elemento in qualche modo riconoscono immediatamente la direzione in cui andare per raggiungere ricompensa.

La differenza tra bambini e adulti è quando riescono a verbalizzare vado a destra o vado a sinistra riescono in qualche modo a dire che OK vado a sinistra perché a sinistra ho visto che c'è il muro blu dove magari quell'angolo c'era qualcosa durante la rotazione. questo è il momento in cui i bambini risolvono il problema come gli adulti cioè al 100% vanno lungo si dirigono direttamente lungo l'angolo dove ci sono i rewards. Questo problema spaziale è risolto grazie alla capacità verbalizzazione.

Total Turing Test

sostanzialmente in pratica invece di andare a considerare un agente software in grado soltanto di rispondere a delle domande immagina che ci sia effettivamente un robot cioè ci sia un'agente con un corpo che sia in grado di svolgere tutta una serie di attività e che sia indistinguibile dalle attività che per esempio in grado di dagli occhi di un osservatore umano rispetto alle attività che è in grado di svolgere.

però una serie di problemi perché per esempio ci sono molte capacità banali per noi anche di manipolazione che non sono attualmente accessibili per i robot.

Super-Simplified Turing Test

Non c'è nessun tipo di barriera adesso cioè direttamente un interrogatore che sa di stare interagendo con una macchina che deve dare un giudizio sostanzialmente su quello che è Human likeness delle risposte oppure del dei feedback.

Il problema è le aspettative dell'interrogatorio. possono esserci dei giudizi diversi rispetto al tipo di output che viene prodotto cioè ci sono dei delle persone che hanno delle aspettative troppo alte rispetto dei criteri di intelligenza che una macchina dovrebbe essere in grado di dimostrare.

Gli altri problemi potrebbero essere

- Chi è l'interrogante?
- Quanto dura la conversazione?
- Di cosa tratta la conversazione?
- Come decide l'interrogante?

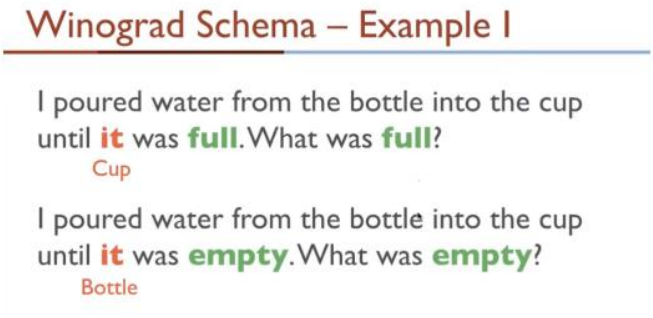
E quindi appunto il test di turing e le sue varianti non può essere visto come un vero test di intelligenza.

Gli altri problemi da menzionare possono essere

- Il completamento del testo è un test di previsione, non un test di composizionalità
- Mancanza di ragionamento di Common Sense

Winograd Schema Challenge

come alternativa al test di turing per valutare l'intelligenza di sistemi. sostanzialmente dal punto di vista linguistico questo qui è un compito che deve risolvere un problema che si chiama anafora resolution la risoluzione delle concordanze grammaticali. Questo è un test behavioristico perché teine conto sul output.



Per rispondere a queste domande dobbiamo fare concordanze e ragionamento.

Pro

Non c'è soggettività. su questi test qua la risposta è chiara cioè si sa perfettamente quale deve essere la risposta.

Con

è di nuovo un test linguistico

nel test di turing è possibile utilizzare tutta una serie di trick quindi è possibile creare dei chatbot che in qualche modo fanno finta di aver capito ma utilizzano una serie di strategie in cui rimandano la risposta poi indietro all'utente. Ma in winograd no è possibile e domande sono binare.