Knowledge Based Systems

(Introduzione all'Intelligenza Artificiale)

versione 1.0

Delf

Super Engineer
Email: delfme76@hotmail.com

Indice

Knowledge Based Systems.....1
Disclaimer 3
Note 5
Introduzione all'Intelligenza Artificiale 7
Diamoci una posizione ontologica.....13
Entriamo nel vivo di un K.B.S. 17

Disclaimer

Io inizierei con una affermazione molto in voga nel panorama underground internazionale:

" …bla…bla…bla…. Le persone dovrebbero essere giudicate in base alle loro conoscenze piuttosto che ai propri denari….bla…bla…bla….le multinazionali….bla bla bla…circolazione delle idee….bla bla…".

Io questa cosa qui non l'ho mai capita. Si vuole combattere la globalizzazione dei saperi, la standardizzazione delle menti che questa società cerca di imporre, fino ad arrivare ad individuare nella differenziazione dei modi di ragionare e della qualità dei risultati intellettuali prodotti un elemento di distinguo, confronto tra gli uomini, ma anche un catalizzatore delle sinergie realizzabili tra menti diverse. Tutto questo senza rendersi conto che di fatto si sta arrivando a contrapporre un elite ad un'altra elite!! Assurdo…! L'idea di base, come spesso accade, è stata stravolta oppure viene usata come bandiera propagandistica o come scudo di menti avide che succhiano come sanguisughe un sapere maturato da altri con enorme sacrificio personale e di cui non sono degni! Ma questo è il web….si posta e via…non si può vedere il volto di chi allunga la mano e prende…scarica bit d'informazione, fa un sapiente collage, ne ricava parole con cui inizia ad esprimersi pur continuando a non sapere un cazzo di niente. Ci sarebbero molte cose da dire ma ancor di più molte cose da fare….Vedete, un sistema chiuso è come un figlio nato male già dal principio. Esso può essere efficiente ed anche efficace, capace di produrre cose buone per un certo periodo di tempo, ma alla lunga è destinato a morire. L'unica cosa che non muore mai, bensì evolve sempre, è un sistema aperto dinamico….evolve perché scambia energia con l'esterno e scambia energia con

l'esterno perché evolve. Questo è il web e ciò che esso scambia è l'informazione. E' con questo spirito che vi invito ad accogliere questi scritti che vengono concepiti con l'intento primario di non fornire l'informazione in sé ma l'informazione accompagnata da una giusta chiave di lettura. Ciò che vogliono realizzare non è la diffusione di un determinato sapere tecnologico o altro ma lo stimolo all'apertura mentale, incidere sulla forma mentis del lettore il fine più ambizioso. Perciò prendete tutto con le pinze ed "INDAGATE" …. Indagate sempre. Buona lettura a tutti.

Note

Livello di difficoltà: avanzato??!
Requisiti necessari per la lettura di questo articolo: stabilitelo voi :-)
Per tutto ciò che veniamo a sapere: ringraziamo il nostro Padre Celeste che conosce l'esigenze della nostra anima,del nostro corpo e della nostra mente, e non trascura nessuna di esse.

Questa serie di tutorial vengono concepiti con lo scopo di fornire una conoscenza di base per lo sviluppo dei K.B.S. che rappresentano, unitamente allo sviluppo di reti neurali, una delle principali tecnologie applicative dell' I.A. Il materiale che fornisco è scritto nei ritagli di tempo in cui mi diletto nel partecipare alla diffusione di quelle conoscenze difficilmente raggiungibili sia per le loro difficoltà intrinseche sia per un voluto tentativo di erezione di barriere all'entrata nel sistema "conoscenza tecnologica". Per quanto riguarda le difficoltà intrinseche alla comprensione di determinati argomenti, può essere limitata tramite l'uso di una didattica plasmata opportunamente fino all'inverosimile. Questo comporta una enorme difficoltà nello scrivere i tutorial caratterizzati da enormi tempi di analisi e concepimento degli stessi. In più il fatto che il tempo dedicato alla loro stesura sia molto ridotto da una attività lavorativa sempre più incalzante e caratterizzata da ritmi frenetici e da un dispendio di enormi energie mentali, implica una "uscita" irregolare dei tutorial stessi e una accuratezza della forma stilistica appena sufficiente ad evitare eventuali figuracce! :-))
Perciò personalmente consiglio la lettura di questi testi a gente fornita di enorme pazienza e realmente interessate all'argomento trattato. Fiducioso di

riuscire a portare a termine l'impegno preso (anche perché opportunamente stimolato da Quequero), spero che per voi rappresenti una gradita lettura (altrimenti fermiamo subito la macchina perché il tempo è denaro). Per quanto riguarda eventuali software da utilizzare più avanti vi saranno forniti direttamente senza dovere andare in cerca sul web, anche perché, pur usando lo stesso linguaggio per la scrittura di codice, useremo, nel tempo, diversi ambienti di sviluppo. Voglio ricordare che tutto ciò che ho scritto in questo capitolo è di dominio pubblico (almeno di quel pubblico che lo considera attendibile ed interessante) :))

Eventuali copie di questo materiale possono essere scambiate liberamente a patto di non violare il senso ed i contenuti originari. Confido solo nel rispetto delle proprietà intellettuali. Se qualcuno di voi avesse una preparazione in materia tale da voler contribuire alla stesura di questa serie di articoli si faccia vivo (più si è, meno si lavora!) ;)

I contenuti che saranno esposti sono da considerarsi una prima stesura ed, ovviamente, caratterizzati da una serie di difetti di cui sono già al corrente e che provvederò a correggere nel tempo! :)

Introduzione all'Intelligenza Artificiale:

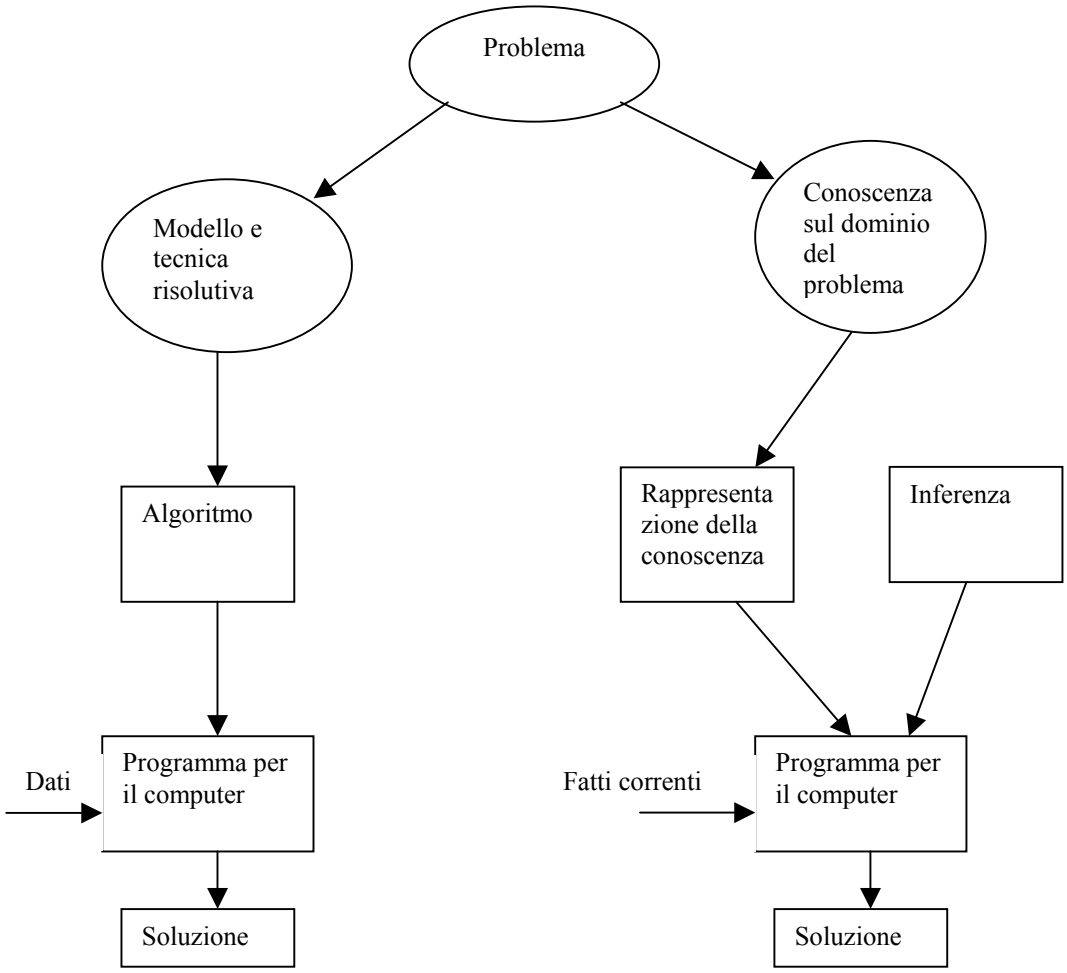
Essa è una disciplina sorta verso la metà degli anni Cinquanta, quando l'avvento dei computer fece intravedere la possibilità di realizzare macchine capaci di comportamenti confrontabili con quelli attribuiti alla mente umana. In effetti, all'euforia iniziale subentrò via via la consapevolezza della grande difficoltà dell'impresa, per molti anni, l'A.I. rimase materia di ricerca in ambito accademico, fornendo contributi importanti all'informatica generale, ma senza approdare a risultati concreti. Una svolta verso l'applicazione pratica si ebbe negli anni Settanta, quando venne introdotta una tecnologia specifica, quella dei "sistemi esperti" o "sistemi basati sulla conoscenza". Questa tecnologia nota come KBS (knowledge based system) non copre l'intero arco d'azione dell'A.I. ma è l'unica, oggi, realmente trasferibile in applicazioni effettive.

Ma che cos'è, in effetti, l'A.I. ? Una definizione brillante ma vaga è quella di Hofstadter secondo cui "l'A.I. è tutto ciò che i computer non sanno ancora fare". Esistono ,ovviamente, altre definizioni più precise ma che rivelano una grande discordanza di pareri sugli obiettivi dell'A.I..

Si possono, infatti, distinguere due principali correnti di pensiero: una, la cosiddetta "tesi forte", ritiene che si possano creare macchine che siano un vero e proprio duplicato della mente umana; l'altra la "tesi debole", mira invece pragmaticamente a realizzare macchine capaci di svolgere compiti "intelligenti". Mentre il traguardo della tesi forte sembra molto lontano (se mai raggiungibile), nell'ottica della tesi debole si può dire che esistano ormai realizzazioni interessanti e ampie prospettive di sviluppo. E' su quest'ottica che

focalizzeremo la nostra attenzione. Si può cercare di capire cos'è l'A.I. esaminando i problemi di cui si occupa. Essi possono essere raggruppati schematicamente in due classi. La prima classe ha come obiettivo di dotare le macchine di capacità sensoriali (visione, udito, ecc) e di abilità di manipolazione. Alla seconda classe, invece, appartengono temi che vanno dalla comprensione del linguaggio naturale al sapere giocare a scacchi (giochi a due), dalla dimostrazione automatica di teoremi alla traduzione da una lingua all'altra, e così via. La risoluzione di questi problemi si può ricondurre a una capacità specifica dell'uomo, cioè la "capacità di ragionamento". E' questa capacità l'obiettivo centrale dell'A.I..

Una capacità di questo tipo conferirebbe al computer una fisionomia radicalmente diversa da quella tradizionale. Infatti, da macchina che esegue algoritmi, si trasformerebbe in macchina che "trova" algoritmi:



Approccio DS: il sistema perviene alla soluzione tramite modelli e tecniche predefinite

Approccio KB: il sistema perviene alla soluzione tramite deduzioni sulla conoscenza rappresentata

Proviamo a mettere in evidenza i diversi modi d'essere, attribuibili agli approcci evidenziati in figura. L'approccio D.S. (Decision support system) associabile alla risoluzione di un problema attraverso l'uso della programmazione classica e quello K.B. (knowledge based system) associabile alla risoluzione di un

problema attraverso l'utilizzo di deduzioni logiche successive effettuate attraverso un motore inferenziale (strategia di ragionamento) e su una conoscenza opportunamente rappresentata al calcolatore. Vi ricordate quando all'inizio avete letto che il livello di questo articolo era "avanzato" inteso come difficile ? Sicuramente nel corso della lettura vi sarete detti " cazz qui si capisce tutto facile facile, perché è stato proposto come difficile?" Vi rispondo che se non fate attenzione a questi concetti ed alla teoria di base che rappresentano troverete grosse difficoltà più avanti in fase di programmazione vera e propria!! Infatti dovete cambiare il vostro normale modo di ragionare e ciò avverrà attraverso un illuminazione che avrete nel momento stesso in cui capirete profondamente la differenza tra una realtà rappresentata ed una imposta. Vabbo' andiamo avanti e poi ritorneremo sul concetto in modo più preciso. A sinistra abbiamo lo schema di un vecchio caro D.S. , immaginiamo che il problema sia calcolare l'area di un triangolo. In tal caso il "modello o tecnica risolutiva" sarà una semplice formula (B*H)/2 dove B=base, H= altezza (scusate se a volte offendo la vostra intelligenza ma giuro che lo faccio in buona fede!) ; l'algoritmo sarà rappresentato da una semplice immissione di due campi e dall'esecuzione di una formula. In parole povere in tale caso il computer deve solo eseguire dei calcoli su dati numerici senza conoscere il "significato" dell'algoritmo che esegue e delle informazioni su cui esso opera… (che scemo!!!).

A destra abbiamo lo schema di ciò che ci interessa studiare , il nostro amato K.B..

In tale caso il computer deve conoscere i principi della geometria piana, comprendere il problema che gli viene sottoposto e saper eseguire

ragionamenti sulle conoscenze di cui dispone al fine di costruire una soluzione (ci ha le palle!!). Se guardate lo schemino abbiamo la rappresentazione della conoscenza (del dominio su cui si opera) e poi una cosa che appare difficile da comprendere che è l'inferenza. L'inferenza per ora la definiamo come il modo di ragionare, il meccanismo noto alla macchina e su cui essa basa la possibilità di muoversi nel dominio di conoscenza attraverso deduzioni di fatti parziali (mah…detto cosi sembra complicato ma vi assicuro che andando avanti tutto diventerà più chiaro, anzi riusciremo a scrivere anche del codice). Si capisce che forniti al computer la conoscenza sul dominio (meglio dire una rappresentazione di essa) e un modo di ragionare esso ci chiederà dei "fatti" accaduti e riguardanti il problema da affrontare e da essi genererà una soluzione!! :)

Ora so benissimo che gia smaniate dalla voglia di scrivere del codice o fare qualcosa di pratico ed a dire il vero potremmo già fare quale piccolissimo esempio. Ma come un allenatore di football prima di dare il pallone alla squadra punta molto sulla comprensione degli schemi di gioco o sul ripristino della forma fisica anche noi ora resisteremo alla tentazione di passare, in modo frivolo, alla pratica e continuiamo a friggerci il cervello su quanto di teorico sta alla base dello sviluppo di un sistema esperto. Solo così avremmo le dovute nozioni per poter approfondire in seguito il nostro sapere in materia, leggere qualsiasi libro pubblicato in merito senza incontrare difficoltà oltre il dovuto, insomma potere parlare di A.I. o di una branca di essa anche stando lontano dal calcolatore!! Inoltre posso garantirvi che chi si addentra in tale campo ne viene in quale misura trasfigurato, contaminato nel modo di pensare e di

vedere la realtà. Chi ha detto "esagerato!!!", uno di voi sicuramente perché l'ho sentito con questi orecchi…. Non ci credi? Sta a vedere….

Abbiamo parlato di dover dare in pasto al computer una base di conoscenza o meglio una rappresentazione "formale" della conoscenza inerente ad un qualche micromondo (oggetto del nostro problema). Pensate un attimo a che sconvolgimento profondo abbia implicato la suddetta necessità. Insomma io devo rappresentare la realtà, ma come posso fare una cosa simile senza prima rispondere a domande del tipo "cos'è la realtà?"," cos'è la conoscenza di essa", "cos'è una rappresentazione di essa","cosa fa parte della realtà?", "…e quindi anke cos'è un qualsiasi oggetto contenuto in essa(e quindi anche noi!!) ?"!! Vi rendete conto?? Io prima di incontrare Dio nella mia vita al solo pensare una delle seguenti domande rischiavo di rimanere inebetito a vita! Dobbiamo prendere una "Posizione ontologica" cioè dobbiamo poter arrivare a dire "per noi la realtà è questo! E la conoscenza di essa è quest'altro!"….in realtà dovevano prendere una posizione. Infatti il problema si pose a chi anni or sono si è imbattuto in tale problematica. Il mondo scientifico poteva rivolgersi a filosofi della portata di Platone, Socrate,o altri colleghi famosissimi ed invece iniziarono a guardare con simpatia ad un folle filosofo del linguaggio (un genio, e di quelli estrosi in ogni senso ;) "a buon intenditore poche parole") di nome Wittgenstein. E' stato illuminante in tale campo quasi quanto Einstein lo fu nella fisica, ma non possiamo dilungarci oltre anche in questo e rimando a voi eventuali approfondimenti.

Diamoci una posizione ontologica:

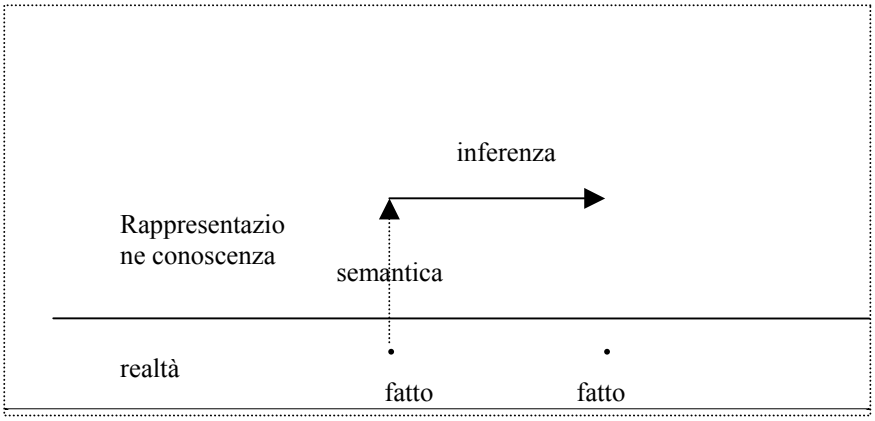
Assumiamo una posizione ontologica per arrivare a degli sviluppi pratici. Come già accennato bonariamente assumere una posizione ontologica ci porta a domande quali:

- cos'è la conoscenza
- cos'è la realtà
- cos'è una rappresentazione della realtà

Tutto ciò va opportunamente definito e codificato per essere passato in pasto alla macchina!

Assumiamo come "realtà" tutto ciò che accade o è accaduto (formalità dei fatti). Se la realtà è una collezione di fatti indipendenti l'uno dall'altro le relazioni tra i fatti sono una nostra creazione. La rappresentazione è una espressione dei fatti in un linguaggio (es linguaggio naturale, quello cioè usato dagli individui). Però il linguaggio naturale non è solo nato per rappresentare fatti ma anche emozioni. Noi escludiamo tale parte emotiva e la dipendenza tra i fatti. Questa è una posizione ontologica "molto forte"! (il fatto che il sole sorga oggi non è legato al fatto che sorga ogni giorno, me lo dice la fisica ma per me non è una conseguenza, capita perché semplicemente accade!) Capite che con questa posizione non si va lontano…. In qualche modo, almeno in un micromondo, debbo poter legare i fatti tra di loro.

La conoscenza è un organizzazione delle espressioni su ciò che accade, è accaduto o può accadere (cioè sul reale o sul possibile!). Per cosa è usata la conoscenza?



Quando accadono dei fatti, la conoscenza, sapendo solo cose parziali di essa, attraverso un opportuna inferenza (ipotizzando così l'accadimento di altri fatti), arriva a derivare altri fatti.

Ovviamente tale conoscenza è su un dominio limitato. In oltre essa mi viene fornita da un esperto in un linguaggio naturale e quindi non trattabile. Il nostro compito è quello di studiare strutture e linguaggi (artificiali) per poter rappresentare la conoscenza in un sistema K.B.S. (informatico).

Devo però anche considerare che chi interagirà col sistema non conosce il linguaggio artificiale da noi usato. Facciamo ora una seconda assunzione ontologica (debole, è quella che usiamo): "il mondo è fatto di oggetti e relazioni". Do una individualità e distinguibilità alle cose. Individuiamo l'oggetto e ne rappresentiamo le relazioni col mondo (fra gli oggetti).

Posso avere diversi rapporti:

- con un tipo: ingegnere(delf)
- con un'altra classe: sito(uic)
- con una proprietà: fragile(vaso)
- corrispondenza binaria: ama(Giovanni,Maria)

- corrispondenza n-aria: studente(nome,cognome,matricola,etc) rappresento un record ;)
- funzione: n+1 (n gli argomenti della funzione ed 1 è il risultato, per es lo xor ha n+1=3)
- probabilità: LancioMoneta(testa,1/2)
- etc

Fatta la nostra nuova assunzione ora abbiamo un linguaggio che ci permette di descrivere gli oggetti e i legami (linguaggio della logica).
Per ora gli oggetti li identifichiamo con dei simboli semplici, il cui significato sta nell'uso comune che ne facciamo noi (ha un senso solo nel nostro micromondo ed in un altro dominio ne avrebbe un altro). Vi sembra difficile da capire? Papà vi deve fare un esempio? E vabbo'… immaginiamo una semplice scritta contenente queste parole "sono stato qui per Francesca!"
Per es:

Sono stato qui
Per Francesca

Essa avrebbe un senso se la leggiamo sulle mura di un carcere ed un altro se la leggiamo appuntata su una lavagnetta in un quale ufficio di lavoro, etc non è il top ma è la prima cosa che mi è venuta in mente (ricordatevi che sto scrivendo a getto quindi la qualità ne risente).
Il Turbo-prolog (che è il linguaggio che useremo noi) ci costringe alla dichiarazione dei tipi degli oggetti nella sezione "DOMAINS" ed in essa sono possibili tutti i tipi standard (integer, real, character, string, single, etc). Le

relazioni tra gli oggetti, invece, le indicheremo col nome di predicati e vanno nella sezione "PREDICATES" definiti attraverso un funtore che è il nome del predicato e poi il tipo degli oggetti che sono in relazione:

"funtore(tipo,tipo,tipo,…..)"
Immaginiamo che immetta un qualcosa del genere: modulo2 (integer,integer,integer). Il predicato definisce una relazione tra tre insiemi di interi e non dice quale sia la relazione ma solo la dichiara. Nella sezione CLAUSES posso, invece, esplicitare le clausole rappresentanti la mia conoscenza sul dominio. Continuando l'esempio banale:
modulo2(0,0,0)
modulo2(1,0,1)
modulo2(0,1,1)
modulo2(1,1,0)
Beh si capisce subito che non è altro che la funzione XOR (il terzo elemento della terna è il risultato dello xor tra i primi due). Mandando in run e posizionandomi nell'interfaccia utente potrei chiedere
Modulo2(1,0,x) ed il sistema restituirà 1 (cioè il terzo campo che io cerco).
Una volta che ho rappresentato la conoscenza posso sfruttarla e renderla replicabile nello spazio e nel tempo facendo così a meno dell'esperto (Ho una macchina che pensante).

Entriamo nel vivo di un K.B.S. :

NB: Tutte le architetture serie sono fatte di moduli che interagiscono tra loro ed opportunamente realizzati in fase di progettazione.

Il K.B.S. è un sistema computerizzato che utilizza la conoscenza su un qualche dominio per risolvere un problema inerente al dominio stesso. La prima cosa da fare è vedere nel dettaglio le parti fondamentali di un KBS per apprezzarne le differenze coi DSS. Tutto ciò è stato gia visto all'inizio di questo articolo evidenziando l'esistenza di due moduli a noi del tutto nuovi:

- base di conoscenza
- inferenza

Ricapitolando il contesto dei KBS è l'Intelligenza Artificiale: Essa ha obiettivi diversi a seconda delle assunzioni che facciamo:

Assunzione forte: le capacità della mente umana sono indipendenti dal supporto fisico dove la mente risiede. Cioè si prescinde dal supporto biologico e si trasferiscono i processi che avvengono nella mente su un altro supporto (hardware).

Assunzione debole: si limita a studiare sistemi intelligenti capaci di riprodurre delle conoscenze umane e capacità logiche di rappresentazione ed inferenza. Questa è la strada che percorriamo noi. Col tempo, grazie ad un adeguato interessamento da parte di tutti e che coinvolga conoscenze informatiche, elettroniche e matematiche , potrebbe rivelarsi interessante e possibile portare avanti uno studio basato sull'assunzione forte. Nel frattempo posso garantirvi che le cose fattibili nel campo dei KBS sono cosi tecnologicamente avanzate ed impensabili da iniettare ugualmente un bel po' di adrenalina del nostro cervello!. Potremmo arrivare a costruire la parte software di un piccolo e

semplice robot-guida, oppure un sistema di supporto alla lettura di cartine stradali, o chissà cos'altro!. Rimane il caro buon vecchio discorso…dopo aver appreso qualcosa sta a noi avere le capacità di costruirci sopra qualcosa di interessante ed "originale". Una frase di un celebre film diceva qualcosa di simile a "Lei ti ha indicato la strada ….ora sta a te riuscire a seguirla!" ;)
Ora mi evito da solo l'ingrato compito di riportare lo schema di un KBS poiché non è altro che la parte destra della figura precedentemente introdotta nell'articolo. Quindi mentre leggete guardate lì.
Per affrontare il problema il KBS considera la "conoscenza su un dominio". Dove sta la conoscenza? Sta in un esperto che già ha risolto tali problemi (per es un medico che guarda una lastra sa dire quando c'è una frattura da qualche parte) oppure in maniera implicita o esplicita nell'azienda o gruppo di ricerca (beh…se vogliamo dotare una vettura di un sistema intelligente di guida sappiamo già da noi che se i sensori d'avanti misurano una distanza dalla vettura che precede di 2 metri allora bisogna frenare per ritornare in condizione di sicurezza).Tale conoscenza la devo rappresentare e ciò avviene per mezzo di un linguaggio artificiale nel modulo di "rappresentazione conoscenza". Tale conoscenza la posso rappresentare in modo "estensivo" cioè tutti i possibili casi oppure in modo "non estensivo". In tale caso ora mi trovo di fronte al problema di avere da qualche parte un intelligenza che mi dice, di volta in volta, in che caso mi trovo per poter poi prendere delle decisioni. Questa è l'inferenza. L'accoppiata di tali due moduli viene sviluppata in un "programma" che prende i fatti correnti e genera una soluzione.
Vantaggi e problematicità di tale approccio:

Avere separata la base di conoscenza dall'inferenza rende il programma più modulare. Potremmo avere stessa inferenza e cambiare solo la conoscenza sul dominio, oppure ad una stessa conoscenza applicare diversi metodi di inferenza. Ricordiamoci, infatti, che l'inferenza è la codifica di un ragionamento. Il ragionamento classico (usato nella programmazione ordinaria in C,C++,Visual Basic,etc,etc) è il "modus ponens" (ragionamento deduttivo o deterministico). Se ho una regola che dice:

- se sussiste A allora B.

Tale regola fa parte della rappresentazione della conoscenza (è una conoscenza che ho sul dominio).

"Se piove prendi l'ombrello".

Allora quando accadono i "fatti" (piove o non piove) l'inferenza fa si che si arrivi a B a seconda dei casi correnti che può assumere A. L'insieme di regole fanno parte della K.B. (knowledge base).

Oss: ovviamente le regole non sono il solo paradigma per rappresentare la conoscenza.

Il problema allora è articolato in varie fasi:

- come rappresentare sta benedetta conoscenza
- l'inferenza (trasferire il ragionamento al sistema)
- implementazione

Un sistema di inferenza deve risolvere vari problemi:

- ricercare nella K.B. quali regole utilizzare, cioè quelle pertinenti al problema (fase di matching)
- selezionare almeno una di esse (fase di scheduling)

- eseguire la regola e registrare in memoria le conseguenze (fase di firing)

Posso vederlo come un ciclo. Volta per volta ho informazioni sul problema, allora ricerco regole, scelgo e ne applico talune finche arrivo ad una soluzione del problema originario passando via via per soluzioni "intermedie".

Es:

Se A and B allora D

Se C allora B.

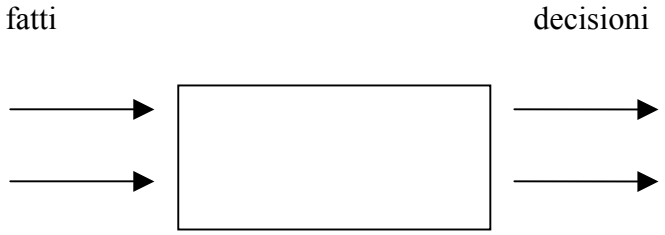
Immaginiamo I seguenti casi correnti :

- accadono A, C

La prima regola non è applicabile però lo è la seconda. Ma la seconda implica anche che accade B. A questo punto diventa applicabile anke la prima regola poiché A and B. Alla fine il sistema restituisce D come risultato.

Vi sono due approcci all'inferenza:

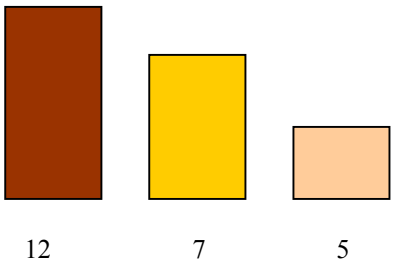
- prendiamo i fatti ed andiamo a propagare in avanti applicando regole ed accrescendo via via la nostra base di conoscenza attraverso deduzioni intermedie (Forward chaining).
- prendiamo ad una ad una le conclusioni finali ed ipotizziamo che siano vere. Andiamo a vedere all'indietro fino ad arrivare ai fatti (infatti per essere vere tali deduzioni vuol dire che devono essere state applicate alcune regole precedentemente, etc). Se i fatti corrispondono vuol dire che ho trovato la soluzione.(Backward chaining).



Un'altra caratteristica dell'inferenza è quella di usare delle "euristiche" (spero sappiate il significato di tale termine altrimenti dovrò provvedere poiché arrivando ad applicazioni avanzate verranno usate delle euristiche) :))

Infatti, come strutture interne alla scatola nera usiamo strutture come gli "spazi di ricerca" (il processo di decisione lo vedo muoversi all'interno di uno spazio che descrive le diverse possibili scelte). L'euristica ci consente, seguendo la conoscenza, di esaminare alcune strade enon tutte quelle presenti all'interno dell'intero spazio di conoscenza. Risparmiamo tempo rendendo più efficiente l'esplorazione.

Comeee? Chi ha detto di non aver capito bene??? Non sia mai!! :)) Ci vuole un esempiuccio:



Immaginiamo 3 recipienti di diverse capacità: 12, 7, 5. Il primo è effettivamente riempito e gli altri vuoti.

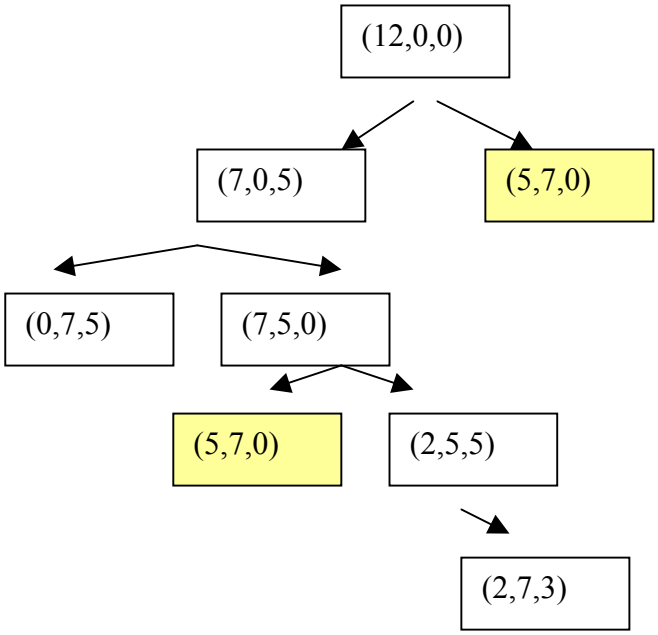
Immaginiamo che il problema che ci si pone è quello di versare esattamente 2 litri nel primo,7 nel secondo e 3 nel terzo ma senza poter usare strumenti di misura!!!. Uno normale dice Cazz… uno che capisce di sistemi esperti dice No-problem. Se non possiamo usare le misurazioni l'unica cosa consentita e di versare acqua da un recipiente all'altro utilizzando solo l'informazioni circa le dimensioni dei recipienti stessi. Creiamo la conoscenza sul dominio con un semplice insieme di terne.

(12,0,0) è lo stato iniziale: primo colmo e gli altri due vuoti.

(7,0,5) è un possibile primo passaggio intermedio: riempio il terzo usando ciò che è nel primo

(5,7,0) è un possibile primo passaggio intermedio: riempio il secondo usando il contenuto del primo

Ma devo tenere conto dei possibili passaggi e di quelli non concessi. Creiamo un semplice spazio di ricerca: Vabbe non seguo tutto le strade ma giusto il tanto per fare capire che da qualke parte ci sarà la terna (2,7,3). La soluzione è (12,0,0) ; (7,0,5) ; (7,5,0) ; (2,5,5) ; (2,7,3)



Come si nota dal grafo se dovessimo elencare tutti i possibili stati sarebbe oneroso, in più alcuni stati si ripetono (vedi colorati). Una euristica mi permette di ottimizzare la ricerca nello spazio in modo da evitare ripetizioni inutili di stato ed altre complicazioni varie. L'inferenza deve sapere le regole di passaggio da uno stato ad un altro. Vedremo in seguito, quando scriveremo del codice vero e proprio, come ciò risulta molto semplice sfruttando la potenza di un linguaggio di programmazione logica (nello specifico il prolog). Nel frattempo vi saluto lasciandovi la possibilità ed il tempo per assimilare i concetti fin qui esposti. Un esercizio carino sarebbe quello di risolvere simulando il ragionamento di un K.B.S. il simpatico problema del lupo,capra,cavolo e barcaiolo:

Bisogna trasferire da una parte ad un'altra del fiume lupo, capra,cavolo. Il barcaiolo può portare solo un passeggero alla volta (anche il cavolo è considerato un passeggero) ma non possono rimanere sulla stessa sponda le seguenti coppie (lupo-capra) (capra-cavolo) per evitare che il lupo mangi la capra o la capra si mangi il cavolo. Nel proseguo vi fornirò la soluzione o alcune dritte in caso ce ne fosse bisogno. In realtà spero che non vi riesca difficile. Sicuramente è un indicatore oltre che di quanto abbiate appreso le notizie fin qui contenute, anche del livello di apertura mentale acquisita durante lo studio della programmazione classica.