

INTELLIGENZA ARTIFICIALE

- Il campo dell'Intelligenza Artificiale o AI (dall'inglese Artificial Intelligence) tenta di capire e costruire entità intelligenti.
- Nata nel 1956. Quale definizione di Intelligenza? Quale definizione di Intelligenza Artificiale?
- Alcune definizioni:
 - È lo studio di come far fare ai calcolatori cose che, ora come ora, gli esseri umani fanno meglio (definizione transitoria...., scacchi....)
 - È la costruzione di un manipolatore di simboli che è in grado di soddisfare il test di Turing

1

TEST DI TURING

- Interazione con un terminale in cui io posso fare domande e ottengo risposte. Dall'altra parte c'è o una persona o un computer. Se dopo 30 minuti non sono in grado di distinguere fra persona e computer....
- Il computer deve avere le seguenti capacità:
 - Elaborazione del linguaggio naturale;
 - Rappresentazione della conoscenza;
 - Ragionamento automatico;
 - Apprendimento automatico.

2

1

TEST DI TURING GLOBALE (interazioni fisiche dirette)

- Visione artificiale;
- Robotica.
- Questioni filosofiche legate all’Intelligenza Artificiale:
Al debole: è possibile costruire macchine in modo che agiscano *come se* fossero intelligenti?
- Argomenti contro:
 - Ci sono cose che i computer non possono fare indipendentemente da come li si programmi (incompletezza dei sistemi formali);
 - Fallimento nel lungo periodo (apprendimento);
 - L’effettiva costruzione di programmi appropriati non è fattibile (complessità della conoscenza).

3

PRINCIPIO DI IA FORTE

- E` possibile costruire macchine che pensino intelligentemente? (che abbiano *menti coscienti reali*?)
- Solleva alcuni dei problemi concettuali più difficili di tutta la filosofia.
- I fondamenti dell’IA sono:
 - Filosofia;
 - Matematica (logica);
 - Psicologia;
 - Linguistica;
 - Ingegneria dei calcolatori.

4

STORIA DELL'IA IN PERIODI

- **(1943-1956)**

- *La gestazione dell' IA*
- reti neurali, programmi per il gioco degli scacchi, dimostratori di teoremi;

- **(1952-1969)**

- Entusiasmo Iniziale, Grandi Aspettative: prima il calcolatore era solo concepito come elaboratore aritmetico. General Problem Solver, Programmi per il Gioco della Dama in torneo, Reti Neurali, Il linguaggio LISP.

- Due filoni:

- McCarthy (Stanford) Logica
- Minsky (MIT) Visione anti-logica, Micromondi (mondo a blocchi)

5

STORIA DELL'IA IN PERIODI

- **(1966-1974)**

- *Una dose di realtà*
- Alcuni programmi non erano davvero *competenti* (ELIZA, traduzioni puramente sintattiche), altri erano intrattabili (esplosione combinatoria). Le reti neurali erano inadeguate.

- **(1969-1979)**

- *Sistemi basati sulla conoscenza: la chiave del potere?*
- Conoscenza intensiva su un dominio di ampiezza limitata. Sistemi Esperti.

6

STORIA DELL'IA IN PERIODI

- **(1980-1988)**

- *L'AI diventa un'industria*
- Sistemi Esperti commerciali di successo;
- Progetto quinta generazione giapponese (1981);
- Compagnie per lo sviluppo di sistemi di AI;
- Fondi per la ricerca.

- **(1986-oggi)**

- *Il ritorno delle reti neurali*
- Algoritmo di apprendimento con propagazione all'indietro.
- Delusione sui sistemi basati sulla conoscenza.

7

EVENTI RECENTI E SVILUPPI FUTURI

- Più realismo e nuova consapevolezza.
- Superamento dei micromondi;
- AI in molti campi pratici quali pianificazione, scheduling, agenti intelligenti distribuiti;
- Ulteriori aree :
 - Internet, conoscenza troppo ampia che va filtrata in modo intelligente.

8

INTELLIGENZA ARTIFICIALE: DUE CORRENTI

- **MACCHINE INTELLIGENTI:**
 - si costruiscono programmi che raggiungono un alto livello di competenza nella conoscenza di problemi particolari
 - approccio ingegneristico
 - Non ci si occupa di simulare l'attività umana di ragionamento, ma di emularla selettivamente.
- **SCIENZA COGNITIVA:**
 - si cerca di modellare il comportamento umano e i suoi processi di informazione
 - approccio di filosofi, psicologi, linguisti, biologi.
 - Il computer è un mezzo di sperimentazione.
 - Siamo ancora lontani dalla costruzione della macchina "intelligente", per cui ci si è limitati per adesso a problemi più semplici e trattabili.

9

CATEGORIE DI ATTIVITA' DELLA IA

- Attività Normali (ovvero come è possibile che cose così semplici siano così complicate?):
- Processo del linguaggio naturale:
 - Comprensione;
 - Generazione;
 - Traduzione.
- Percezione:
 - Visione;
 - Linguaggio parlato.
 - Ragionamento di buon senso
 - Controllo di Robot

10

CATEGORIE DI ATTIVITA' DELLA IA

- **Attività Formali:**

- Giochi;
- Scacchi;
- Dama
- Matematica e Logica;
- Prova automatica di Teoremi;
- Geometria;
- Calcolo differenziale;
- Dimostrazione di proprietà di programmi.

11

CATEGORIE DI ATTIVITA' DELLA IA

- **Attività Specializzate:**

- Ingegneria
- Progetto;
- Ricerca di guasti (diagnosi);
- Pianificazione della produzione;
- Programmazione automatica
- Diagnosi medica
- Analisi finanziaria
- Analisi scientifica ecc.

12

LA DISCIPLINA DELL'IA

- L'Intelligenza Artificiale è una disciplina giovane e non ancora assestata.
- Dopo una fase empirica si sono trovati alcuni *principi astratti* comuni.
- Intelligenza Artificiale ha tre grosse aree:
 - Strategie di Ricerca
 - Rappresentazione della Conoscenza
 - Applicazioni
 - Visione simbolica e dichiarativa.
- Alternative (sistemi sub-simbolici):
 - Connessionismo, reti neurali;
 - Brooks e la costruzione di piccoli sistemi artificiali reattivi (insetti).

13

LA DISCIPLINA DELL'IA

- Noi seguiremo principalmente la visione simbolica e dichiarativa (Intelligenza Artificiale "classica").
 - I due approcci possono essere integrati.
- **SISTEMI BASATI SULLA CONOSCENZA**
 - Un sistema di Intelligenza Artificiale generalmente esamina un largo numero di possibilità e costruisce dinamicamente una soluzione.
 - "La potenza di un programma intelligente nel risolvere un problema dipende primariamente dalla quantità e qualità di conoscenza che possiede su tale problema". (Feigenbaum)

14

SISTEMI BASATI SULLA CONOSCENZA

- Il programma non è un insieme di istruzioni immutabili che rappresentano la soluzione del problema, ma un ambiente in cui:
 - rappresentare
 - utilizzare
 - modificare
 - una base di conoscenza.
- PRINCIPI ARCHITETTURALI
 - Ogni sistema basato sulla conoscenza deve riuscire ad esprimere due tipi di conoscenza:
 - Conoscenza sul dominio dell'applicazione (COSA);
 - Conoscenza su COME utilizzare tale conoscenza per risolvere problemi - CONTROLLO.

15

PROGRAMMA = CONOSCENZA + CONTROLLO

- Questi due tipi di conoscenza dovrebbero essere tenuti **SEPARATI**:
 - ALTA MODULARITÀ
 - PROGRAMMAZIONE DICHIARATIVA
- La conoscenza sul problema è espressa indipendentemente dal suo utilizzo (COSA non COME);
- Alta modularità e flessibilità;

16

PROGRAMMA = CONOSCENZA + CONTROLLO

- I moderni linguaggi e AMBIENTI per Intelligenza Artificiale tendono a riprodurre tale schema;
- Definire un ambiente (linguaggio) in tale schema significa definire come il programmatore può esprimere la conoscenza e quale tipo di controllo può utilizzare;
- Problematiche di **rappresentazione** della conoscenza;
- Problematiche di organizzazione del problema e **strategie di controllo**

17

Linguaggi e Ambienti per Sistemi basati sulla conoscenza: DICHIARATIVO - PROCEDURALE

- **Esempio: Tracciare una circonferenza**
 - DICHIARATIVO
 - La circonferenza è l'insieme dei punti equidistanti da un dato punto detto centro.
 - PROCEDURALE
 - Per tracciare una circonferenza
- **Esempio: Ordinamento di una lista**
 - DICHIARATIVO
 - Il risultato dell'ordinamento di una lista vuota è la lista vuota
 - Il risultato dell'ordinamento di una lista L è L' se la lista L' è ordinata ed è una permutazione di L
 - PROCEDURALE
 - Controlla prima se la lista è vuota; se sì dai come risultato la lista vuota
 - Altrimenti calcola una permutazione L' di L e controlla se è ordinata; se sì termina dando come risultato L' , altrimenti calcola un'altra permutazione di L etc....

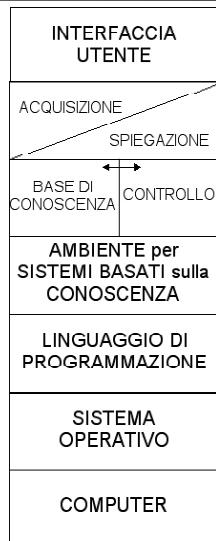
18

VANTAGGI DELL'APPROCCIO DICHIARATIVO

- **ALTO LIVELLO:**
 - è più semplice impostare il problema (catena di IF..THEN);
- **GENERALITÀ:**
 - posso usare la stessa conoscenza in differenti modi;
- **FLESSIBILITÀ**
 - facile modifica, debugging ecc.
 - cambio del controllo

19

UNA NUOVA MACCHINA VIRTUALE



20

10

UNA NUOVA MACCHINA VIRTUALE

ESEMPIO

- **Semplicissimo problema di diagnostica:**
 - prescrivere una medicina in base ai risultati di un esame di laboratorio.
- **GOAL: prescribe (Drug)**
 - cioè "prescrivere una medicina adeguata per un determinato paziente".

21

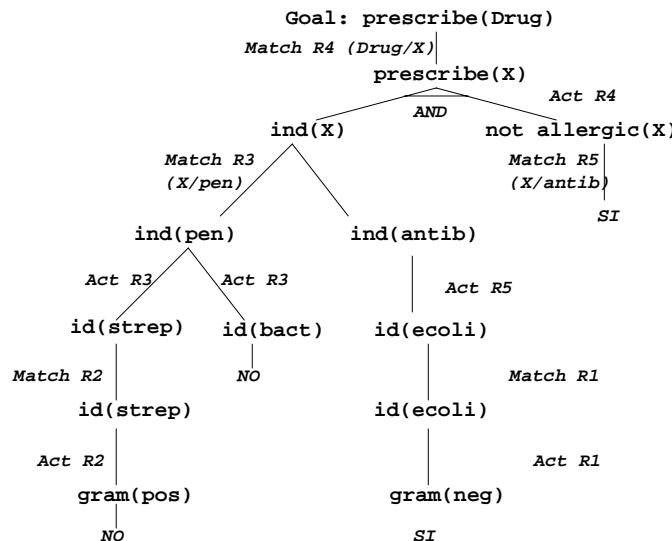
UNA NUOVA MACCHINA VIRTUALE

- **FATTI:**

```
gram(neg).  
not(allergic(antb)).
```
- **REGOLE:**
 - **R1: `gram (neg) → id (ecoli).`**
Se il risultato dell'esame è *gram-negativo* allora l'identità è *enterium-coli*
 - **R2: `gram (pos) → id (strep).`**
Se il risultato dell'esame è *gram-positivo* allora l'identità è *streptococco*
 - **R3: `id(strep) OR id(bact) → ind(pen).`**
Se l'identità è streptococco o batterio allora è bene indicare penicillina
 - **R4: `ind(X) AND not (allergic(X)) → prescribe(X)`**
Se è bene indicare una certa medicina e il paziente non è allergico a tale medicina, allora si può prescrivere tale medicina al paziente
 - **R5: `id(ecoli) → ind(antb).`**
Se l'identità è *enterium-coli* allora è bene indicare antibiotici

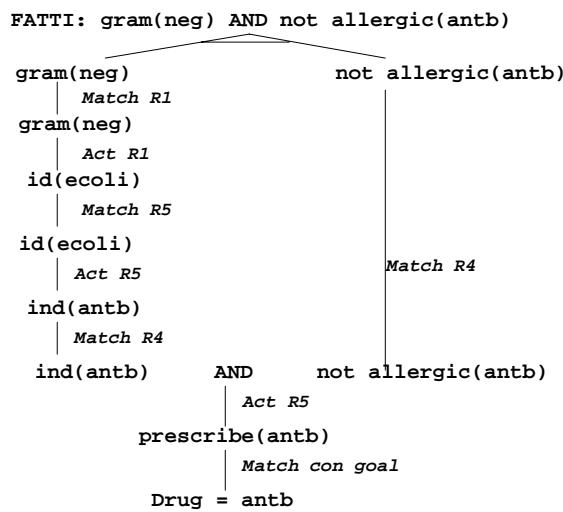
22

CONTROLLO BACKWARD GRAFO AND/OR



23

CONTROLLO FORWARD GRAFO AND/OR



24

CONSIDERAZIONI

- I fatti iniziali sono molto numerosi e dunque è bene domandarli selettivamente all'utente quando necessario.
- L'impostazione del problema preferibile è backward.
- askable: gram e allergic.
- Esempio di sessione di lavoro:
 - User: esecuzione
 - Sys: È la coltura Graam positiva ?
 - User: NO.
 - Sys: È la coltura Graam negativa?
 - User: SI.
 - Sys: Il paziente è allergico agli antibiotici ?
 - User: NO.
 - Sys: Allora prescriverei una cura a base di antibiotici.

25

SESSIONE DI SPIEGAZIONE

- Sys: È la coltura Graam positiva ?
- User: **WHY?** < perchè è importante determinare tale informazione?
- Sys: Tale informazione è di aiuto per determinare se (R1) l'identità dell'organismo è streptococco.
- User: **WHY?**
- Sys: Tale informazione è di aiuto perchè per tale organismo è indicata la penicilina (R3).
- Sys: ... Allora prescriverei una cura a base di antibiotici.
- User: **HOW?** <come puoi concludere tale responso?
- Sys: Ho utilizzato la regola R4: ho prescritto antibiotici poichè sono indicati per tale infezione (prem1) e il paziente non è allergico (prem2).
- User: **HOW Premessa 1 ?**
<come puoi concludere che sono indicati gli antibiotici?
- Sys: Ho utilizzato la regola R5: sono indicati antibiotici poichè l'organismo che ha causato l'infezione è ecoli.

26

STRUTTURA ALGORITMICA: Parte 1a

```
program prescribe(input,output);
var Answer: string[2]; presc: boolean;
begin
  presc:=false;
  write('e la coltura Graam Positiva?');
  read(Answer);
  if Answer='Si' then
  begin
    write('il paziente è allergico alla penicillina?');
    read(Answer);
    if Answer='No' then
    begin
      write('allora prescriverei una cura a base di
            penicillina');
      presc:=true;
    end;
  end;
end;
```

27

STRUTTURA ALGORITMICA: Parte 2a

```
if not presc then
begin
  write('coltura Graam Negativa?');
  read(Answer);
  if Answer='Si' then
  begin
    write('il paziente è allergico
          agli antibiotici?');
    read(Answer);
    if Answer='No' then
    begin
      write('allora prescriverei una cura a base di
            antibiotici');
      presc:=true;
    end;
  end;
end;
if not presc then
  write('prescrizione impossibile'); end.
```

28

Soluzione del problema come ricerca in uno spazio degli stati:

- Gran parte dei problemi di Intelligenza Artificiale hanno la ricerca (o controllo) come componente fondamentale.
- I problemi si possono modellare come **Problemi di Ricerca** in uno spazio degli stati (**Strategie di Ricerca**).
- **Spazio degli stati**
 - Lo spazio degli stati è l'insieme di tutti gli stati raggiungibili dallo stato iniziale con una qualunque sequenza di operatori.
- **Lo Spazio degli Stati è Caratterizzato da:**
 - Uno **stato iniziale** in cui l'agente sa di trovarsi (non noto a priori);
 - Un **insieme di azioni** possibili che sono disponibili da parte dell'agente (Operatori che trasformano uno stato in un altro o più formalmente una funzione successore $S(X)$ che riceve in ingresso uno stato e restituisce l'insieme degli stati raggiungibili).
 - Un **cammino** è una sequenza di azioni che conduce da uno stato a un altro.

29

TEST: RAGGIUNGIMENTO DEL GOAL

- La verifica può essere solo l'appartenenza dello stato raggiunto all'insieme dello stato (o degli stati) goal.
- A volte lo stato obiettivo può essere descritto in modo “astratto” attraverso proprietà (si pensi allo stato di scacco matto).
- **Altri obiettivi (non solo raggiungere il goal, ma...):**
 - trovare la sequenza di operatori che arrivano al goal;
 - trovare tutte le soluzioni;
 - trovare una soluzione ottima.
 - In quest'ultimo caso vuol dire che una soluzione può essere preferibile a un'altra.
 - Una funzione **costo di cammino** assegna un costo a un cammino (in gran parte dei casi quale somma del costo delle azioni individuali lungo il cammino).

30

L'EFFICACIA DELLA RICERCA

- Si riesce a trovare una soluzione?
- È una buona soluzione? (con basso costo di cammino – costo in linea)
- Qual è il costo della ricerca? (tempo per trovare una soluzione – costo fuori linea)
- Costo totale di ricerca = costo di cammino + costo di ricerca.
- Scegliere stati e azioni → L'importanza dell'astrazione

31

ESEMPIO: IL GIOCO DEL 8

- Stati: posizione di ciascuna delle tessere;
- Operatori: lo spazio vuoto si sposta a destra, a sinistra, in alto e in basso;
- Test obiettivo: descrizione dello stato finale;
- Costo di cammino: ciascun passo costa 1.

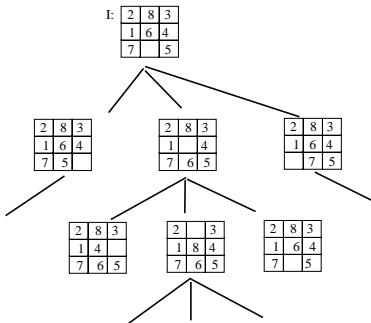
32

ESEMPIO: IL GIOCO DEL 8

Stato iniziale Goal

I:	2 8 3
	1 6 4
	7 5

G:	1 2 3
	8 4
	7 6 5



33

ESEMPIO: MISSIONARI E CANNIBALI

ESEMPIO:

- 3 missionari e 3 cannibali devono attraversare un fiume. C'è una sola barca che può contenere al massimo due persone. Per evitare di essere mangiati i missionari non devono mai essere meno dei cannibali sulla stessa sponda (stati di fallimento).
- Stato: sequenza ordinata di tre numeri che rappresentano il numero di missionari, cannibali e barche sulla sponda del fiume da cui sono partiti.
- Perciò lo stato iniziale è: (3,3,1) (nota l'importanza dell'astrazione).

34

ESEMPIO: MISSIONARI E CANNIBALI

- Operatori: gli operatori devono portare in barca
 - 1 missionario, 1 cannibale,
 - 2 missionari,
 - 2 cannibali,
 - 1 missionario
 - 1 cannibale.
- Al più 5 operatori (grazie all'astrazione sullo stato scelta).
- Test Obiettivo: Stato finale (0,0,0)
- Costo di cammino: numero di traversate.

35

ESEMPIO: CRIPTOARITMETICA

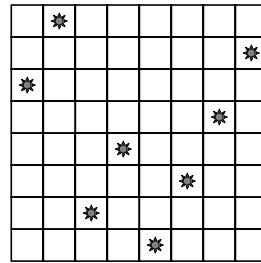
$$\begin{array}{r} \text{S} \quad \text{E} \quad \text{N} \quad \text{D} \quad + \\ \text{M} \quad \text{O} \quad \text{R} \quad \text{E} \quad = \\ \hline \text{M} \quad \text{O} \quad \text{N} \quad \text{E} \quad \text{Y} \end{array}$$

- Operatori: sostituisci tutte le stesse lettere con una cifra che non compare nel rompicapo;
- Test obiettivo: il rompicapo contiene solo cifre e rappresenta una somma corretta;
- Costo di cammino: 0

36

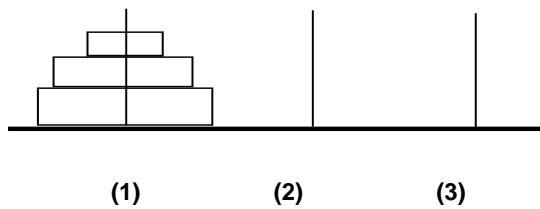
IL PROBLEMA DELLE N REGINE

- Inserire 8 regine su una scacchiera in modo che non si mangino.
- Stati: qualsiasi configurazione da 0 a N regine sulla scacchiera;
- Operatori: aggiungi una regina in un qualsiasi quadrato;
- Test obiettivo: N regine sulla scacchiera, nessuna minacciata;
- Costo di cammino: zero.



37

ESEMPIO: LA TORRE DI HANOI



- Spostare i cilindri concentrici da (1) a (3) nella stessa configurazione di (1) utilizzando eventualmente anche (2). Un cilindro più grande non può essere inserito su un cilindro più piccolo.

38

ALTRI ESEMPI

- Il problema del commesso viaggiatore:
 - Un commesso viaggiatore ha una lista di città che deve visitare tutte una sola volta. Vi sono strade dirette fra ogni coppia di città. Si trovi la strada più breve che il commesso deve seguire per compiere un viaggio completo che inizi e termini in una qualsiasi delle città.
 - (esplosione combinatoria, per 10 città è 10!)
- Il problema della scimmia e la banana:
 - Afferrare una banana appesa al soffitto (avendo a disposizione una sedia e bastone).

39

ESEMPIO: CAPRA, LUPO E CAVOLO

- Portare capra, lupo e cavolo dall'altra parte senza che si mangino (si mangiano se rimangono assieme senza il conducente della barca). Nella barca se ne può trasportare uno solo alla volta.
- Soluzione:
 - Porta la capra sull'altra sponda;
 - Torna indietro
 - Porta il cavolo sull'altra sponda
 - Porta indietro la capra
 - Porta il lupo sull'altra sponda
 - Torna indietro
 - Porta la capra sull'altra sponda

40

SPAZIO DI RICERCA MOLTO AMPIO

- Giochi: scacchi
- Le parole crociate

T	U	O
R		R
E	R	A

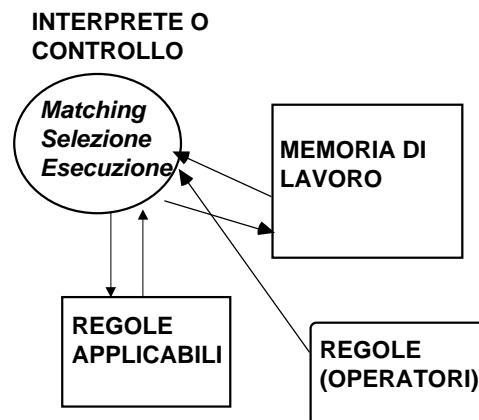
41

Schema architetturale: SISTEMA DI PRODUZIONE

- Insieme di Operatori (regole);
- Uno o più database (Memorie di lavoro);
- Strategia di Controllo.
- MODULARITÀ - FLESSIBILITÀ
- Operatori:
 - `IF <pattern> THEN <body>`
 - non si chiamano per nome, ma si attivano in base al pattern-matching

42

ARCHITETTURA GENERALE:



43

DUE MODALITÀ DI "RAGIONAMENTO"

• FORWARD O DATA-DRIVEN:

- La memoria di lavoro nella sua configurazione iniziale contiene la conoscenza iniziale sul problema, cioè i fatti noti.
- Le regole di produzione applicabili sono quelle il cui antecedente può fare matching con la memoria di lavoro (F-rules).
- Ogni volta che una regola viene selezionata ed eseguita nuovi fatti dimostrati vengono inseriti nella memoria di lavoro.
- Il procedimento termina con successo quando nella memoria di lavoro viene inserito anche il goal da dimostrare (condizione di terminazione).

44

DUE MODALITÀ DI "RAGIONAMENTO"

- BACKWARD O GOAL-DRIVEN:

- La memoria di lavoro iniziale contiene il goal (o i goal) del problema.
- Le regole di produzione applicabili sono quelle il cui conseguente può fare *matching* con la memoria di lavoro (B-rules).
- Ogni volta che una regola viene selezionata ed eseguita, nuovi subgoals da dimostrare vengono inseriti nella memoria di lavoro.
- Il procedimento termina con successo quando nella memoria di lavoro vengono inseriti fatti noti (CONDIZIONE DI TERMINAZIONE).

45

QUANDO APPLICARE BACKWARD E QUANDO FORWARD ?

- Esistono più Stati Iniziali o più Goals?

- Quale è il numero medio di rami generati da un singolo nodo?

- Quale è la modalità di ragionamento più naturale? (spiegazione all'utente)

- BIDIREZIONALE O MISTO:

- È la combinazione dei metodi descritti precedentemente;

- La memoria di lavoro viene suddivisa in due parti l'una contenente i fatti e l'altra i goals o subgoals;

- Si applicano simultaneamente F-rules e B-rules alle due parti di memoria di lavoro e si termina il procedimento con successo quando la parte di memoria di lavoro ricavata mediante backward chaining è uguale o un sottoinsieme di quella ricavata mediante (CONDIZIONE DI TERMINAZIONE).

46

BIDIREZIONALE O MISTO

- L'idea è quella di mantenere ed estendere un insieme di sequenze soluzioni parziali.
 - Un agente con diverse opzioni immediate di esito sconosciuto può decidere cosa fare esaminando prima le differenti sequenze possibili di azioni che conducono a stati di esito conosciuto scegliendo, poi, quella migliore.
 - Il processo di cercare tale sequenza è chiamato **RICERCA**.
 - È utile pensare al processo di ricerca come la costruzione di un albero di ricerca i cui nodi sono stati e i cui rami sono operatori.
- Un algoritmo di ricerca prende come input un problema e restituisce una soluzione nella forma di una sequenza di azioni.
- Una volta che viene trovata la soluzione, le azioni suggerite possono essere realizzate.
 - Questa fase è chiamata **ESECUZIONE**.

47