

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.

1

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;

2

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**

3

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....

4

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

5

UNA NUOVA MACCHINA VIRTUALE



6

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".

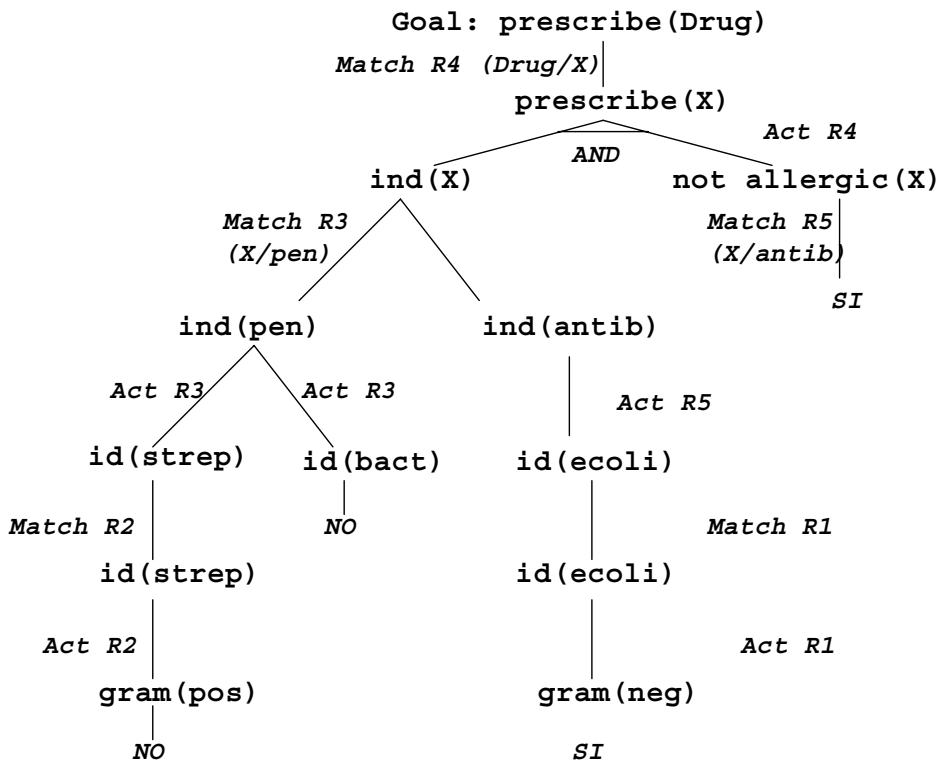
7

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 bactero 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

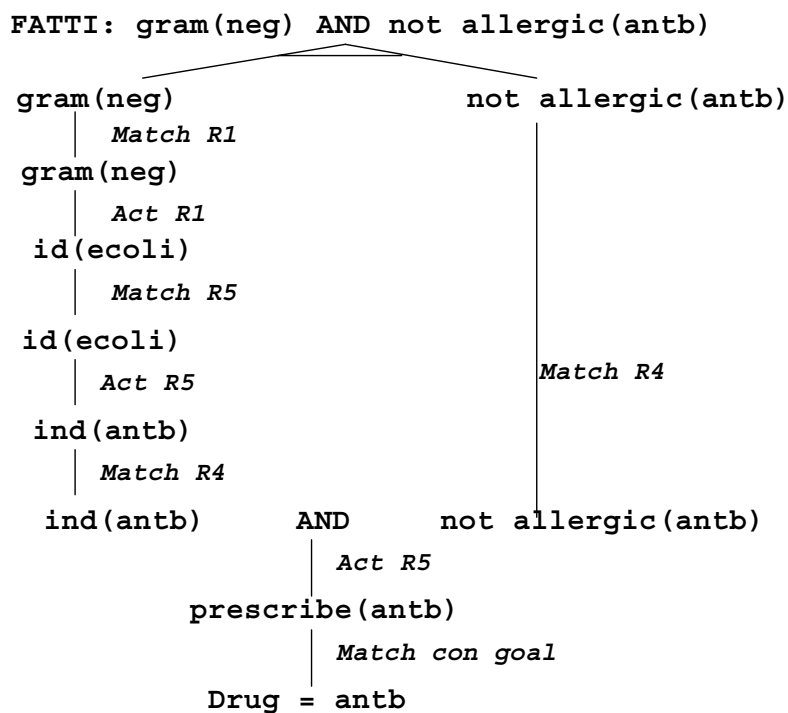
8

CONTROLLO BACKWARD GRAFO AND/OR



9

CONTROLLO FORWARD GRAFO AND/OR



10

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 Gram positiva ?
 - User: NO.
 - Sys: È la coltura Gram negativa?
 - User: SI.
 - Sys: Il paziente è allergico agli antibiotici ?
 - User: NO.
 - Sys: Allora prescriverai una cura a base di antibiotici.

11

SESSIONE DI SPIEGAZIONE

- Sys: È la coltura Gram 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 prescriverai 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.

12

STRUTTURA ALGORITMICA: Parte 1a

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

13

STRUTTURA ALGORITMICA: Parte 2a

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

14