

IA basata sulla logica

Lezione 2: Strumenti

19/3/2003

IA basata sulla logica
Mario Ornaghi

1

- La lezione scorsa abbiamo visto la nozione di agente razionale e l'approccio logico nelle sue linee generali
- In questa lezione daremo una breve e incompleta panoramica degli strumenti utilizzati in quest'ambito

19/3/2003

IA basata sulla logica
Mario Ornaghi

2

1. Principali strumenti

- LISP
 - Dovuto a McCarthy
 - Basato sulle liste e sulle nozioni di lambda-
astrazione e di applicazione di funzione
 - Primo esempio di linguaggio funzionale; sono
stati sviluppati numerosi linguaggi funzionali

19/3/2003

IA basata sulla logica
Mario Ornaghi

3

- Prolog
 - Kowalski, Colmerauer
 - Linguaggio logico: le computazioni possono essere
considerate come ricerche goal-oriented di
dimostrazioni in un frammento della logica del primo
ordine
- Usato come
 - Linguaggio di programmazione
 - DCG grammars: elaborazione del linguaggio naturale
 - Datalog: modello logico per le basi di dati
 - Sistema per la rappresentazione della conoscenza
 - Usato più in generale in vari aspetti della IA

19/3/2003

IA basata sulla logica
Mario Ornaghi

4

- Estensioni di Prolog
 - Programmazione logica di ordine superiore
 - Lambda-prolog
 - Logical frameworks, ad es. Isabelle
 - Constraint Logic Programming (CLP)
 - S-Models
 - ...

19/3/2003

IA basata sulla logica
Mario Ornaghi

5

- Shell per Sistemi esperti
 - Un sistema esperto rappresenta la conoscenza di un esperto in un determinato settore
 - Una rappresentazione tipicamente usata è data da un insieme di regole implicative e il motore inferenziale può procedere per forward o backward chaining
 - PROLOG può essere usato (backward chaining)
 - Vi sono shell appositamente studiate per sistemi esperti
 - TIPICHE SHELL:
 - CLIPS: forward chaining
 - JESS: algoritmo Rete (esistono implementazioni interfacciate a Java)

19/3/2003

IA basata sulla logica
Mario Ornaghi

6

2. Cenni sul Prolog

- Sintassi
 - Atomo: $p(t_1, \dots, t_n)$
 - Esempio: `maggiore(succ(X), X)`
 - Letterale: Atomo, `not(Atomo)`
 - Clausola: Atomo `:-` Letterale, .., Letterale.
 - Programma: sequenza ordinata di clausole.

19/3/2003

IA basata sulla logica
Mario Ornaghi

7

- Semantica logica:
 - $A :- L_1, \dots, L_n$ si legge: $L_1 \wedge \dots \wedge L_n \rightarrow A$
 - Se non si usa la negazione ogni programma ha un modello di minimo e le computazioni dell'interprete sono aderenti alla verità nel modello minimo
 - Se si usa la negazione vi possono essere più modelli

19/3/2003

IA basata sulla logica
Mario Ornaghi

8

Esempio

```
scappa(X) :- gatto(X), vede(X,Y), cane(Y).  
gatto(felix).  
cane(fido).
```

gatto(felix).
cane(fido).

```
scappa(X) :- gatto(X), vede(X,Y), cane(Y).  
gatto(felix).  
cane(fido).  
vede(felix,fido).
```

gatto(felix).
cane(fido).
vede(felix,fido).
scappa(felix).

19/3/2003

IA basata sulla logica
Mario Ornaghi

9

- **Semantica operativa:**
 - Basata sull'unificazione
 - strategia di ricerca di una prova SLDNF:
 - backward, in profondità
 - Negazione come fallimento
 - La ricerca è in profondità, per cui il programma può andare in loop anche se vi sono soluzioni
 - La negazione può dare problemi di non validità (floundering)
 - No occur check
 - Come conseguenza l'aspetto "programmazione" diventa ineliminabile

19/3/2003

IA basata sulla logica
Mario Ornaghi

10

- Unificazione
 - A unifica con B con unificatore s sse As coincide con Bs
- **ESEMPIO**

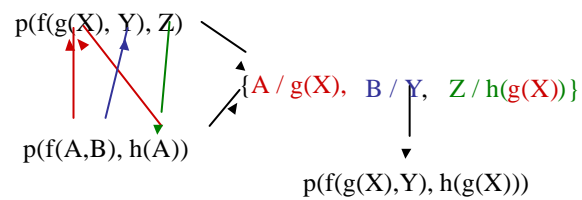
$$p(s(X),s(Y)) \{X/a, V/s(Y)\} = p(s(a),V) \{X/a, V/s(Y)\}$$
- Per non perdere soluzioni usa l'unificatore più generale (mgu, most general unifier):
- **A unifica con B con mgu S sse**
 - $A\sigma = B\sigma = U$
 - Per ogni unificatore δ con $A\delta = B\delta = V$, V è un caso particolare di U
 - cioè $V = U\eta$ per qualche η

19/3/2003

IA basata sulla logica
Mario Ornaghi

11

Esempio



19/3/2003

IA basata sulla logica
Mario Ornaghi

12

Esempio di Question-answering

- `scappa(T, X) :- gatto(T, X), vede(T, X,Y), cane(T, Y).`
- `gatto(T,miao)`
- `gatto(T, felix).`
- `cane(T, fido).`
- `vede(0,fido,felix).`
- `vede(1,fido,felix).`
- `vede(1,felix,fido).`

Query: `scappa(T,X)?`
Answer: `T=0, X=miao; (more?)`
 `T=1, X=felix; (more?)`
 `NO`

19/3/2003

IA basata sulla logica
Mario Ornaghi

13

3. Un Esempio

- Per dare un'idea di come si modelli una realtà in una "base di conoscenza", vediamo un esempio

19/3/2003

IA basata sulla logica
Mario Ornaghi

14

I passi di costruzione di una KB:

- **Analisi del problema**
- **Determinare individui e relazioni**
 - fare le astrazioni adeguate
- **Individuare le proprietà (regole generali e fatti)**
 - Le clausole sono vere nelle interpretazioni desiderate?
 - comporta la verità delle risposte in tali interpretazioni
 - Tutti i casi sono coperti?
 - Non vorremmo buchi nelle risposte

19/3/2003

IA basata sulla logica
Mario Ornaghi

15

Il problema

- Dato un impianto, determinare quali lampadine sono accese e quali prese portano corrente, in funzione delle valvole e degli interruttori.


19/3/2003

IA basata sulla logica
Mario Ornaghi

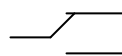
16

Individui

- Astrazione: non interessano leggi di Ohm, ecc.; assumiamo l'esistenza del filo di ritorno, che chiude il circuito, e modelliamo solo i fili di andata, sui quali sono posti gli interruttori;...

 filo

 interruttore

 interruttore a due vie

 lampadina

 presa

 valvola

19/3/2003

IA basata sulla logica
Mario Ornaghi

17

Relazioni e leggi generali

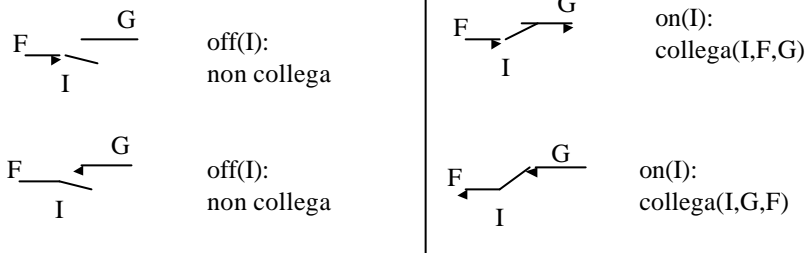
- Se una lampadina è collegata ad un filo e questo è in tensione, la lampadina è accesa
 $\text{accesa}(X) \text{ :- lampadina}(X), \text{connesso}(X,F), \text{tensione}(F).$
- Se una presa è collegata ad un filo e questo è in tensione, la presa porta corrente:
 $\text{corrente}(X) \text{ :- presa}(X), \text{connesso}(X,F), \text{tensione}(F).$

19/3/2003

IA basata sulla logica
Mario Ornaghi

18

- Quando un filo è in tensione? Quando è collegato alle valvole e queste non sono saltate; oppure quando è in contatto con un altro filo in tensione, direttamente o tramite un interruttore chiuso a una o due vie.
- Conviene prima formalizzare gli interruttori e i contatti:

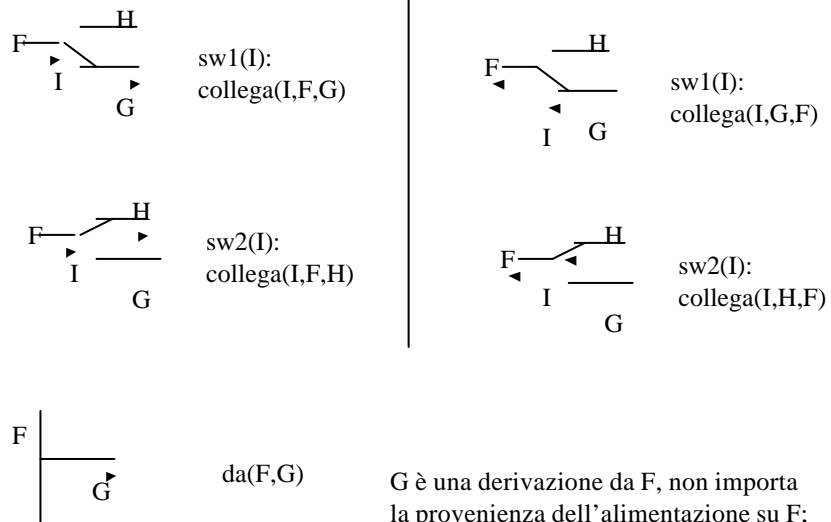


La direzione è dalla alimentazione (le valvole) al filo messo in contatto dall'interruttore

19/3/2003

IA basata sulla logica
Mario Ornaghi

19



19/3/2003

IA basata sulla logica
Mario Ornaghi

20

- Possiamo ora raccogliere i vari casi in un concetto:
 contatto(F,G): il filo F è in contatto con il filo G e
 l'alimentazione proviene da G
ORIENTAMENTO: G è più vicino ad una valvola di F
- L'assiomatizzazione di contatto:

contatto(F,G) :- filo(F), filo(G), da(G,F).
 contatto(F,G) :- filo(F), filo(G), interruttore(I), collega(I,G,F).

- Ora possiamo tornare ai fili in tensione. Un filo è in tensione quando è collegato alle valvole e queste non sono saltate; oppure quando è in contatto con un altro filo in tensione collegato all'alimentazione (direttamente, o tramite un interruttore)

tensione(F) :- filo(F), valvola(V), connesso(F,V), ok(V).
 tensione(F) :- filo(F), filo(G), contatto(F,G), tensione(G).

RICORSIVA
Termina perchè G
è più vicino ad una
valvola, rispetto ad F

KKB. Regole generali.

Le nostre conoscenze generali sono codificate da questo piccolo “sistema esperto”, che chiameremo

elettricista:

accesa(X) :- lampadina(X), connesso(X,F), tensione(F).

corrente(X) :- presa(X), connesso(X,F), tensione(F).

tensione(F) :- filo(F), valvola(V), connesso(F,V), ok(V).

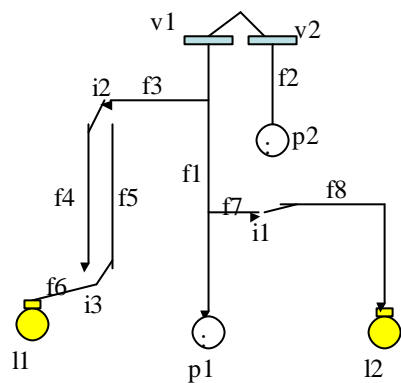
tensione(F) :- filo(F), filo(G), contatto(F,G), tensione(G).

contatto(F,G) :- filo(F), filo(G), da(G,F).

contatto(F,G) :- filo(F), filo(G), interruttore(I), collega(I,G,F).

Vediamo la rappresentazione di un particolare impianto.

Bisogna rispettare la condizione di orientamento



impianto:

lampadina(l1).
 connesso(l1,f6).
 lampadina(l2).
 connesso(l2,f8).
 valvola(v1).
 valvola(v2).
 presa(p1).
 connesso(p1,f1).
 presa(p2).
 connesso(p2,f2).
 interruttore(i1).
 collega(i1,f7,f8) :- on(i1).
 interruttore(i2).
 collega(i2,f3,f4) :- sw1(i2).
 collega(i2,f3,f5) :- sw2(i2).
 interruttore(i3).
 collega(i3,f4,f6) :- sw1(i3).
 collega(i3,f5,f6) :- sw2(i3).
 filo(f1).
 connesso(f1,v1).
 filo(f2).
 connesso(f2,v2).
 filo(f3).
 da(f1,f3).
 filo(f4).
 filo(f5).
 filo(f6).
 filo(f7).
 da(f1,f7).
 filo(f8).

19/3/2003

IA basata sulla logica
Mario Ornaghi

25

- L'impianto può avere diversi stati
 - v1, v2 : ok(vj) o saltata(vj), mutuamente esclusivi
 - i1: on(i1) o off(i1), mutuamente esclusivi
 - i2, i3: sw1(ij) o sw2(ij), mutuamente esclusivi.
 - In tutto: N stati, quanto vale N?

statol:

ok(v1).
 ok(v2).
 on(i1).
 sw1(i2).
 sw2(i3).

19/3/2003

IA basata sulla logica
Mario Ornaghi

26

Osservazioni

- **Nell'esempio abbiamo 3 livelli di conoscenza:**
 - **generale:** regole di ragionamento valide in un'intera classe di modelli, che corrispondono i **mondi** del dominio di problema; rappresentano la conoscenza di un "esperto";
 - KB **elettricista**
 - **correttezza:** Le regole sono vere in tutti i mondi?
 - **copertura:** abbiamo dimenticato qualche ragionamento che porta un elettricista a risolvere i problemi in esame?

19/3/2003

IA basata sulla logica
Mario Ornaghi

27

- **particolare statica:** rappresenta le proprietà statiche rilevanti di un particolare modello, corrispondente ad un particolare mondo:
 - KB **impianto**
 - **correttezza:** le proprietà rappresentate sono vere nel modello (mondo) considerato?
 - **copertura:** abbiamo dimenticato qualche proprietà statica che è necessario conoscere per applicare le regole generali?
- **particolare contingente:** ha le stesse caratteristiche di quella statica, ma rappresenta proprietà dinamiche
 - KB **stato1, stato2,**

19/3/2003

IA basata sulla logica
Mario Ornaghi

28

3.1. Altri esempi

- Ragionamento basato su spiegazioni, diagnosi:

accesa(X, perche(connessa(X,F), S)) :-
lampadina(X), connesso(X,F), tensione(F,S).

corrente(X perche(connessa(X,F), S)) :-
presa(X), connesso(X,F), tensione(F,S).

tensione(F,perche(connesso(F,V),ok(V)) :-
filo(F), valvola(V), connesso(F,V), ok(V).

tensione(F, perche(contatto(F,G),S)) :-
filo(F), filo(G), contatto(F,G), tensione(G,S).

ECC.

- Tipi Dati

nat(0).
nat(s(l)) :- nat(l).

sum(X,0,X).
sum(X,s(Y),s(Z)) :- sum(X,Y,Z).

prod(X,0,0).
prod(X,s(Y),Z) :- prod(X,Y,l), sum(l,X,Z).

- altro

- A conclusione:
 - Prima lezione
 - generalità
 - Seconda lezione:
 - Strumenti
 - esempi tesi ad illustrare l'approccio.