

Lezione 9 giugno

Ancora su diagnosi
Conoscenza incompleta, senso comune e
ragionamento basato su assunzioni
Cenni su pianificazione

Ancora su diagnosi

- Abbiamo accennato alla diagnosi di guasti.
- Occorre aggiungere:
 - gli assumibili riguardano cio' che di norma non è guasto, ad es. `ok(fusibile)`, `ok(lampadina)`, ...
 - negli assiomi `ok(X)` viene usato come assumibile dove interviene una componente che si può guastare
 - per ogni malfunzionamento osservabile si introduce:
`false :- accesa(X), non_accesa(X)`.
 - `false :-` si usa piu' in generale per esprimere vari tipi di vincoli sul dominio di problema
- Vedere libro di testo

Conoscenza incompleta e senso comune

Esempio classico.

```
vola(X) :- uccello(X), not(anormale(X)).
uccello(X) :- struzzo(X).
uccello(X) :- pinguino(X).
uccello(X) :- rondine(X).
...
rondine(r).
struzzo(s).
pinguino(p).
anormale(X) :- pinguino(X).
```

```
? vola(X)
X=r
X=s
```

(ma s non vola)

Ragionamento comune: possibili scenari

- Default reasoning
 - default: cosa è normalmente vero
 - spiegazione: argomento per inferire un dato fatto
 - argomenti "competitivi"
- Abductive reasoning
 - diagnosi (riconoscimento, già considerato)
 - progetto (sotto quali ipotesi uno scopo è ottenibile?)
 - apprendimento induttivo (ipotizzo a partire da osservazioni)
- I due scenari si possono ricondurre al ragionamento basato su assunzioni
 - stesso insieme di concetti e tecniche (o framework)

Frameworks per il ragionamento basato su assunzioni

Frameworks per il ragionamento ipotetico:

Rappresentazione del problema $P = (F, H)$, dove

- F = insieme consistente di fatti e regole generali
 - ad es. insieme di clausole di Horn
- H = insieme di assumibili (o ipotesi possibili)

Scenario D in $P = (F, H)$ di un goal G

- $D \subseteq \text{ground}(H)$ tale che $F \cup D$ sia consistente e si derivi G

Esempio di scenari

Problema P:

Fatti F: $\text{vola}(X) \leftarrow \text{uccello}(X) \wedge \neg \text{anormale}(X)$.
 $\text{uccello}(X) \leftarrow \text{struzzo}(X)$.
 $\text{uccello}(X) \leftarrow \text{pinguino}(X)$.
 $\text{uccello}(X) \leftarrow \text{rondine}(X)$.
 $\text{rondine}(r)$.
 $\text{struzzo}(s)$.
 $\text{pinguino}(p)$.
 $\text{anormale}(X) \leftarrow \text{pinguino}(X)$.
 $\text{false} \leftarrow \text{vola}(X) \wedge \text{anormale}(X)$.

Assumibili H:

$\text{anormale}(X)$

Scenari:

$\text{anormale}(p)$: qui $\text{vola}(s)$ vale
 $\text{anormale}(p), \text{anormale}(s)$: $\text{vola}(s)$ e $\text{vola}(p)$ non valgono
 $\text{anormale}(p), \text{anormale}(s), \text{anormale}(r)$: nessuno vola

Spiegazione minimale M di G in $P = (F, H)$:

- M è uno scenario spiegazione di G, cioè $F \cup M \models G$
- nessun sottoinsieme proprio di M lo è

Scenario massimale:

- M è uno scenario spiegazione di G, cioè $F \cup M \models G$
- nessun soprainsieme proprio di M lo è

Esempio:

$\text{anormale}(s), \text{anormale}(p)$ spiega $\neg \text{vola}(s)$, minimale
 $\text{anormale}(s), \text{anormale}(p), \text{anormale}(r)$
 $\text{spiega } \neg \text{vola}(s)$ scenario massimale

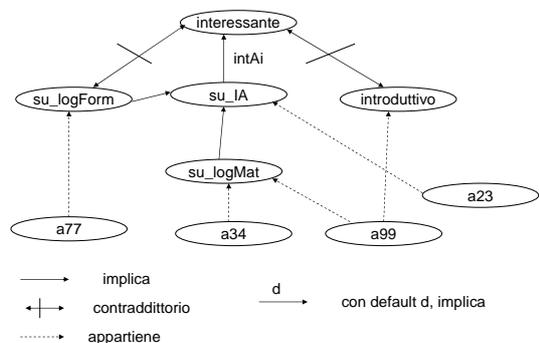
- Si considerano
 - per la diagnosi: le **spiegazioni minimali o argomenti**
 - per la predizione: le **estensioni**: un'estensione è l'insieme delle conseguenze logiche di uno scenario massimale
 - G ammette una spiegazione sse si trova in un'estensione
 - predico G se si trova in tutte le estensioni
- Esempio:
 - $\text{anormale}(s)$ **argomento per** $\neg \text{vola}(s)$
 - $\neg \text{vola}(s), \neg \text{vola}(p), \neg \text{vola}(r)$ **estensione**

- Si tratta di sviluppare i sistemi di inferenza (algoritmi) per trattare con questo framework
 - vedere l'ultimo paragrafo del capitolo 9 del libro

Ragionamento per default

- Tipicamente non monotono
- In un problema $P = (F, H)$
 H assunzioni di normalità
- Un argomento per un goal G è una spiegazione di G basata sulle assunzioni di normalità H
- Le spiegazioni si possono ottenere per abduzione, come visto con il meta-interprete che ritarda le assunzioni di normalità
 - si possono però avere dei vincoli
 - delle preferenze

Esempio 1



- interessante(X) :- suA(X), **intAi(X)**.
- suA(X) :- suLogMat(X).
- suA(X) :- suLogForm(X).
- suLogForm(a77).
- suLogMat(a34).
- suLogMat(a99).
- suA(a23).
- introduttivo(a99).

false :- interessante(X), suLogForm(X).
false :- interessante(X), introduttivo(X).

$$\frac{7 \quad \frac{\text{suA(a23)}}{\text{interessante(a23)}} \quad \text{intAi(a23)}}{1}$$

- interessante(X) :- suA(X), intAi(X).
- suA(X) :- suLogMat(X).
- suA(X) :- suLogForm(X).
- suLogForm(a77).
- suLogMat(a34).
- suLogMat(a99).
- suA(a23).
- introduttivo(a99).

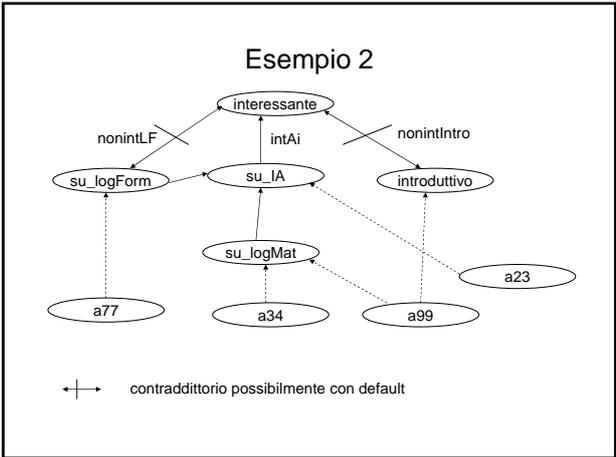
false :- interessante(X), suLogForm(X).
false :- interessante(X), introduttivo(X).

$$\frac{4 \quad \frac{\text{suLogForm(a77)}}{\text{suA(a77)}} \quad \frac{\text{intAi(a77)}}{\text{interessante(a77)}}}{1} \quad \text{MA}$$

- interessante(X) :- suA(X), intAi(X).
- suA(X) :- suLogMat(X).
- suA(X) :- suLogForm(X).
- suLogForm(a77).
- suLogMat(a34).
- suLogMat(a99).
- suA(a23).
- introduttivo(a99).

quindi non posso predire interessante(a77)

false :- interessante(X), suLogForm(X).
false :- interessante(X), introduttivo(X).

$$\frac{4 \quad \frac{\text{suLogForm(a77)}}{\text{suA(a77)}} \quad \frac{\text{intAi(a77)}}{\text{interessante(a77)}} \quad \text{suLogForm(a77)}}{\text{false}} \quad 1$$


- interessante(X) :- suA(X), **intAi(X)**.
- suA(X) :- suLogMat(X).
- suA(X) :- suLogForm(X).
- suLogForm(a77).
- suLogMat(a34).
- suLogMat(a99).
- suA(a23).
- introduttivo(a99).

false :- interessante(X), suLogForm(X), **nonintLF(X)**.
false :- interessante(X), introduttivo(X), **nonintIntro(X)**.

$$\frac{4 \quad \frac{\text{suLogForm(a77)}}{\text{suA(a77)}} \quad \frac{\text{intAi(a77)}}{\text{interessante(a77)}}}{1}$$

assumendo che non valga il default nonintLF(a77), posso spiegare interessante(a77),
ma anche la sua negazione assumendo il default

- interessante(X) :- suA(X), intAi(X).
- suA(X) :- suLogMat(X).
- suA(X) :- suLogForm(X).
- suLogForm(a77).
- suLogMat(a34).
- suLogMat(a99).
- suA(a23).
- introduttivo(a99).

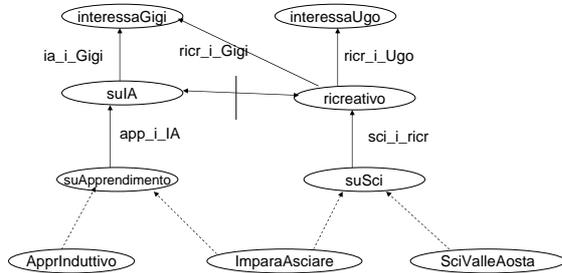
ma se aggiungo la regola di cancellazione

false :- interessante(X), suLogForm(X), nonintLF(X).
false :- interessante(X), introduttivo(X), nonintIntro(X).
false :- suLogForm(X), intAi(X).

$$\frac{4 \quad \frac{\text{suLogForm(a77)}}{\text{suA(a77)}} \quad \frac{\text{intAi(a77)}}{\text{interessante(a77)}}}{1}$$

non predico più interessante(a77)

Esempio 3



Esercizio

- Rappresentare in Prolog
- mostrare che sia $suIA(imparaAsciare)$, sia la sua negazione possono essere spiegate
- Predireste $interessaUgo(imparaAsciare)$?
- Predireste $interessaGigi(imparaAsciare)$?

Risoluzione di argomenti competitivi

- Il problema delle estensioni multiple
 - assunzioni conflittuali (competitive)
 - preferenze (spiegazioni più specifiche) e regole cancellazione
- Semantiche di "predizione": la predizione scettica
 - predice G se G si trova in tutte le estensioni del problema (F,H)

Pianificazione

- Un agente opera in un mondo
- Le azioni dell'agente modificano lo stato del sistema agente+mondo
- si tratta di rappresentare il cambiamento.
- Sarà argomento della parte II del corso.
- In questa parte diamo solo alcune idee guida per un progetto che applichi la ricerca informata (A*) a problemi di pianificazione.

- Rappresentare gli stati:
 - Predicati statici: descrivono parti dell'ambiente non soggette a modifica.
 - Fluenti: fluente = predicato il cui valore di verità può cambiare in seguito ad un'azione dell'agente.
 - Uno stato può essere rappresentato dall'elenco dei fluenti ground veri in esso (per default, gli altri saranno falsi)
- Rappresentare le azioni:
 - effect(A, if Cond then add(AddL)&del(DelL))
 - Cond: condizione abilitante A
 - AddL: i fatti forzato a vero
 - DelL: i fatti forzati a falso

- Azione abilitata in S: Cond vera S (**)
- Eseguire un'azione abilitata in S
do(A,S,S1): $S1 = S \setminus DelL + AddL$
- Pianificatore come ricerca forward nello spazio degli stati
A* +
Compilatore da azioni a neighbours
- Problema di pianificazione: rappresentare un problema e fornire un'euristica
- NOTA (**). Bisogna definire la verità in S. Se si ignora il mondo statico, Cond può essere una lista di fluenti o fluenti negati; in tal caso, Cond vera sse
(F in Cond \rightarrow F in S) e (non(F) in Cond \rightarrow F non in S)

Progetto Ricerca Siti

- Si individui un insieme di argomenti/servizi fra loro collegati, disponibili in rete, all'interno dei quali un utente voglia filtrare quelli di suo interesse
- Si supponga che i risultati grezzi della ricerca siano stati ottenuti e memorizzati opportunamente
- Si assuma che le regole di ragionamento da applicare siano esprimibili in termini di implicazioni con eventuali default, vincoli con eventuali default e eventuali regole di cancellazione, del tipo visto in questa lezione.
- Il progetto dovrebbe fornire un RRS con
 - a) un sistema di rappresentazione di questo tipo di conoscenze
 - b) un corrispondente sistema di ragionamento basato su assunzioniE fornire poi un esempio in cui si applica il RRS

Progetto Caffè

- Un robot si muove in un piano adibito ad uffici, con 2 distributori di caffè. Deve portare dei caffè in determinati uffici. Il robot, prima di distribuire i caffè, calcola un piano, che poi eseguirà.
Costruire un RRS con:
 - a) linguaggio di rappresentazione di ambienti, azioni e stati, pensando ad ambienti bidimensionali in problemi come quello sopra descritto
 - b) si usi A* come sistema di ragionamento, ovvero di calcolo piani.
- Mostrare poi il funzionamento nel caso della distribuzione caffè, provando diverse euristiche