

Lezione 3

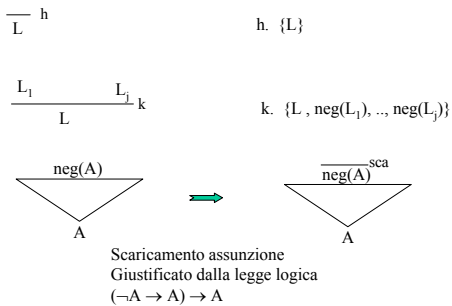
- A) Disgiunzione e negazione, continua
Cenno su FOL
B) Introduzione alla Pianificazione

$$\leftarrow \neg \wedge \rightarrow \leftarrow \leftrightarrow \forall \exists \sigma \cup \subseteq$$

A1. La procedura top down

- Usiamo sempre la forma clausale normale
 $A_1 \vee \dots \vee A_h \vee \neg B_1 \vee \dots \vee \neg B_k$
 sotto forma di insieme $\{A_1, \dots, A_h, \neg B_1, \dots, \neg B_k\}$
- Una KB si suppone sempre ridotta in forma clausale; è quindi un insieme di clausole
 1. C1

 n. Cn
- Le regole di inferenza della procedura top down sono le seguenti
 - dove usiamo: $\text{neg}(A) = \neg A$ e $\text{neg}(\neg A) = A$



DEF. Assunzioni: foglie dell'albero non scaricate

Esempio

- | | |
|-------------------------------|-------------------------|
| 1. $a \vee b \vee d.$ | $\{a, b, d\}$ |
| 2. $c \leftarrow a.$ | $\{c, \neg a\}$ |
| 3. $c \leftarrow b \wedge e.$ | $\{c, \neg b, \neg e\}$ |
| 4. $e.$ | $\{e\}$ |

c

Esempio

- | | |
|-------------------------------|-------------------------|
| 1. $a \vee b \vee d.$ | $\{a, b, d\}$ |
| 2. $c \leftarrow a.$ | $\{c, \neg a\}$ |
| 3. $c \leftarrow b \wedge e.$ | $\{c, \neg b, \neg e\}$ |
| 4. $e.$ | $\{e\}$ |

$$\frac{a}{c} 2$$

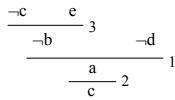
Esempio

- | | |
|-------------------------------|-------------------------|
| 1. $a \vee b \vee d.$ | $\{a, b, d\}$ |
| 2. $c \leftarrow a.$ | $\{c, \neg a\}$ |
| 3. $c \leftarrow b \wedge e.$ | $\{c, \neg b, \neg e\}$ |
| 4. $e.$ | $\{e\}$ |

$$\frac{\frac{\neg b \quad \neg d}{a} 1}{c} 2$$

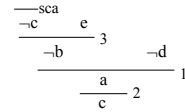
Esempio

1. $a \vee b \vee d.$ $\{a, b, d\}$
2. $c \leftarrow a.$ $\{c, \neg a\}$
3. $c \leftarrow b \wedge e.$ $\{c, \neg b, \neg e\}$
4. $e.$ $\{e\}$



Esempio

1. $a \vee b \vee d.$ $\{a, b, d\}$
2. $c \leftarrow a.$ $\{c, \neg a\}$
3. $c \leftarrow b \wedge e.$ $\{c, \neg b, \neg e\}$
4. $e.$ $\{e\}$

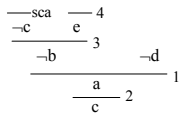


Esempio

1. $a \vee b \vee d.$ $\{a, b, d\}$
2. $c \leftarrow a.$ $\{c, \neg a\}$
3. $c \leftarrow b \wedge e.$ $\{c, \neg b, \neg e\}$
4. $e.$ $\{e\}$

Non abbiamo più
clausole da applicare;
abbiamo dimostrato

c *con assunzione* $\neg d$



Come usare il calcolo

- In genere si dimostra un letterale L a partire da un insieme di assunzioni non risolvibili $L1, \dots, Lk$
- si dimostra cioè $L1 \wedge \dots \wedge Lk \rightarrow L$
- equivalentemente $L \vee \neg(L1) \vee \dots \vee \neg(Lk)$
- Nell'esempio precedente:
 $\neg d \rightarrow c$ equivalente a $c \vee d$
- In generale in una KB disgiuntiva le risposte sono delle disgiunzioni (conoscenza **NON DEFINITA**)

Validità e completezza

- Def. $KB \vdash L1 \vee L2 \vee \dots \vee Lk$ sse esiste un albero di prova (costruito come detto prima) con conseguenza L e assunzioni $L2, \dots, Lk$.
- Validità: $KB \vdash L1 \vee \dots \vee Lk \rightarrow KB \models L1 \vee \dots \vee Lk$
- Completezza nella forma ristretta:
 se KB è soddisfacibile, allora
 $KB \models L1 \vee \dots \vee Lk \rightarrow KB \vdash L1 \vee \dots \vee Lk$

Note

- Il calcolo qui considerato è quello del libro
- Sul libro si trova una realizzazione in DCL del calcolo, compresa la raccolta delle assunzioni, necessaria per estrarre la risposta disgiuntiva.
- Vi sono altri sistemi di dimostrazione, non necessariamente basati sulla forma clausale

A2. FOL e Risoluzione

- La logica del primo ordine piena comprende anche i quantificatori
- Abbiamo già visto sintassi e semantica (limitatamente ai modelli di Herbrandt).
- Accenniamo alla risoluzione come sistema di prova
 - passa per la skolemizzazione, che ha lo svantaggio di distruggere la forma originaria degli assiomi
 - conserva la consistenza, ma la forma di Skolem non è logicamente equivalente alla formula di partenza
 - vi sono altri calcoli, in particolare:
 - sequenti, tableaux

Risoluzione

- Assume che base dati e formula da dimostrare siano in forma clausale
- Una clausola è ancora una disgiunzione di letterali, ma ora le formule atomiche sono della forma $r(t_1, \dots, t_n)$; rappresentiamo, al solito, le clausole come insiemi di letterali.
- La regola di risoluzione è sostanzialmente la risoluzione binaria vista nel caso proposizionale, ma richiede due meccanismi in più:
 - l'unificazione
 - la fattorizzazione

Fattorizzazione

$C\sigma \cup \{L\}$ fattore di $C \cup D$
 sse ogni letterale in $D\sigma$ è uguale ad L ,
 dove σ è un mgu di D

Esempio: $\{r(a), h(f(a))\}$ fattore di $\{r(Y), h(f(Y)), h(f(a))\}$

Operatore res:

$\text{res}(C1 \cup \{A\}, C2 \cup \{\neg B\}, C1\sigma \cup C2\sigma)$ sse $A\sigma = B\sigma$ con mgu σ
 (possiamo commutare: $\text{res}(C1, C2, C3)$ sse $\text{res}(C2, C1, C3)$)

Risoluzione

$$\frac{C_1 \quad C_2}{C}$$

se esistono due fattori C'_1 di C_1 e C'_2 di C_2 tali che
 $C = \text{res}(C'_1, C'_2, C)$

$$\frac{\{A\} \quad \{\neg A\}}{\{\}}$$

Caso particolare: la clausola vuota
 è il **falso** (**l'inconsistenza**)

Esempio

1. $\text{vede}(X, Y) \vee \text{vede}(Y, X) \leftarrow \neg \text{ostacolo}(X, Y) \wedge \text{persona}(X) \wedge \text{persona}(Y)$.
2. $\text{persona}(\text{gigi})$.
3. $\text{persona}(\text{ugo})$.
4. $\text{persona}(\text{mario})$.
5. $\text{ostacolo}(\text{gigi}, \text{ugo})$.

in forma clausale:

1. $\{\text{vede}(X, Y), \text{vede}(Y, X), \text{ostacolo}(X, Y), \neg \text{persona}(X), \neg \text{persona}(Y)\}$
2. $\{\text{persona}(\text{gigi})\}$
3. $\{\text{persona}(\text{ugo})\}$
4. $\{\text{persona}(\text{mario})\}$
5. $\{\text{ostacolo}(\text{gigi}, \text{ugo})\}$

Esempio

$$\frac{\frac{\frac{\{\text{vede}(X, Y), \text{vede}(Y, X), \text{ostacolo}(X, Y), \neg \text{persona}(X), \neg \text{persona}(Y)\} \quad \{\text{persona}(\text{ugo})\}}{\{\text{vede}(X, \text{ugo}), \text{vede}(\text{ugo}, X), \text{ostacolo}(X, \text{ugo}), \neg \text{persona}(X)\}} \quad \{\text{persona}(\text{gigi})\}}{\{\text{vede}(\text{gigi}, \text{ugo}), \text{vede}(\text{ugo}, \text{gigi}), \text{ostacolo}(\text{gigi}, \text{ugo})\}} \quad \{\neg \text{ostacolo}(\text{gigi}, \text{ugo})\}}{\{\text{vede}(\text{gigi}, \text{ugo}), \text{vede}(\text{ugo}, \text{gigi})\}}$$

Nota finale. Usi della risoluzione

- Se KB e query sono in forma clausale, può essere usata come sistema di inferenza, per ottenere dimostrazioni del tipo visto
- Se KB è un arbitrario insieme di assiomi del primo ordine occorre trasformarlo in forma clausale; ciò richiede di eliminare prima i quantificatori in 2 passi:
 - forma normale prenessa
 - skolemizzazione
 - resta solo una quantificazione universale esterna, che può essere lasciata sottintesa
- Poi si passa a forma clausale come nel caso proposizionale

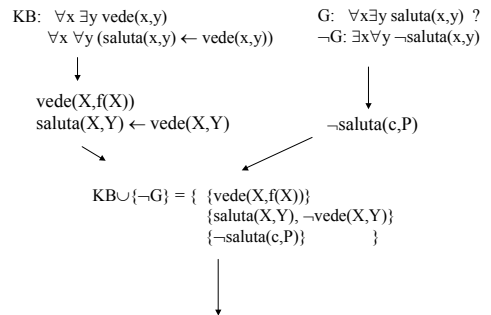
- La skolemizzazione non conserva però l'equivalenza logica (cioè non conserva la stessa classe di modelli), ma solo la consistenza:
- Quindi, se dimostriamo l'inconsistenza della forma clausale, otteniamo l'inconsistenza di KB. L'uso della risoluzione diventa:

Uso della risoluzione per Refutazione

Teorema. $KB \models F$ sse $KB \cup \{\neg F\}$ è inconsistente.

- Per far dimostrare $KB \models F$ dimostro che $KB \cup \{\neg F\}$ è inconsistente, passando a forma clausale.
- Rimandiamo al libro di testo per maggiori dettagli.

ESEMPIO DI DIM. PER REFUTAZIONE



$$\frac{\frac{\frac{\text{saluta}(X1, Y1), \neg \text{vede}(X1, Y1)}{\neg \text{vede}(c, P)} \quad \{\text{vede}(X2, f(X2))\}}{\{}} \quad \{\neg \text{saluta}(c, P)\}}$$

B) Pianificazione Inizio seconda parte

1. Pianificazione: introduzione

- Fino ad ora
 - assunzione mondo statico
 - nell'impianto elettrico vi è una situazione mutevole (fatti contingenti) ma la KB opera in situazioni statiche
- Vi sono diversi sistemi in cui non si può prescindere dal cambiamento (dal tempo); in particolare, agenti che interagiscono con un ambiente mutevole:
 - cambia lo stato del sistema mondo – agente
 - posizione dell'agente nel mondo, temperatura esterna ...
 - in seguito ad azioni dell'agente e/o cambiamenti del mondo

- Agenti reattivi
 - a fronte di determinati eventi reagiscono sempre allo stesso modo; l'agente non cambia
 - esempio: allarme anti-incendio
 - non sono autonomi
- Agenti che debbono raggiungere uno scopo nel tempo, tramite azioni
 - hanno input mutevoli dal mondo (in funzione del cambiamento delle relazioni fra agente e mondo)
 - debbono "ragionare" sul cambiamento/tempo
 - debbono considerare piani di azione (sequenze di azioni atte allo scopo)
 - autonomia

2. Il tempo

- Astrazione orologio
 - tempo discreto, ed es. $t \in \{0, 1, 2, 3, \dots\}$
 - tempo continuo, ad es. $t \in \mathbb{R}$
 - intervalli $(a..b)$, $[a..b]$, $(a..b]$, $[a..b)$
 - puntuale: in ogni punto dell'intervallo
 - gestalt: in tutto l'intervallo ma non necessariamente in sottointervalli

- Astrazione cambiamento: il trascorrere del tempo è evidenziato dal cambiare delle cose
 - tempo basato sugli eventi
 - eventi = accadimenti osservabili ed isolabili nel tempo e nello spazio assunti come "tic-tac"
 - tempo basato sugli stati
 - le azioni mappano uno stato nel successivo; stato 0 (tempo 0), stato 1 (tempo 1), ...

3. Rappresentazione del tempo

- Vediamo alcune rappresentazioni storicamente sviluppate in IA:
 1. Azioni e tempo non codificate nella logica interna del RRS
 - i singoli stati agente-mondo sono rappresentabili internamente
 - es: $at(ugo, tavolo)$
 - non posso chiedere a quale ora ugo è a tavola
 - l'azione di $mangia(ugo)$ attraverso il cambiamento
 - preconditione: $[at(ugo, tavolo), su(tavolo, piattopieno)]$
 - diventa falso: $[su(tavolo, piattopieno)]$
 - diventa vero: $[su(tavolo, piattovuoto)]$

2. Reificazione del tempo/stato

- termini del RRS per rappresentare il tempo/stati
- predicati come : $at(ugo, tavolo, ore(12:30))$
 - posso chiedere : $?at(ugo, tavolo, T)$
 - non posso chiedere cosa vale alle ore 12:30
- Esempio di azioni: $mangia(P, X)$, con tempo discreto
 $su(X, piattovuoto, s(T)) :-$ $tavolo(X),$
 $persona(P),$
 $at(P, X, T),$
 $su(X, piattopieno, T),$
 $mangia(P, T).$

3. Avendo reificato il tempo/stato, meta-predicati:

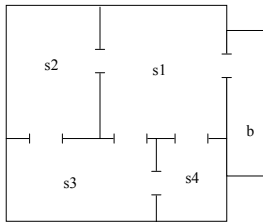
- holds(Predicato, Tempo)
- es: holds(at(tavolo,ugo), ore(12:30))
 - posso chiedere: ? holds(S, ore(12:30))
 - ? holds(su(tavolo,piattopieno),T)
 - si ricordi l'uso di prop nelle reti semantiche
- azioni del tipo detto prima; la maggior capacità espressiva consente meta-regole generali del tipo:


```
holds(S,do(Act,T)) :- next(Act,S1,S),
                        holds(S1,T).
```

4. Classificazione delle relazioni

- È utile classificare le relazioni in
 - statiche
 - dinamiche
 - primitive
 - derivate

Esempio



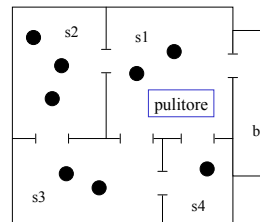
Relazioni statiche primitive.

balcone(b).
stanza(s1).
stanza(s2).
stanza(s3).
stanza(s4).
da(b,s1).
da(s1,s2).
da(s2,s3).
da(s1,s3).
da(s3,s4).
da(s4,s1).

Relazione statica derivata.

conn(X,Y) :- porta(X,Y); porta(Y,X).

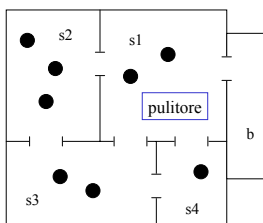
Esempio



Relazione dinamica, tipo STRIPS.

sporco(s1,2).
sporco(s2,3).
sporco(s3,2).
sporco(s4,1).
in(s1,pulitore).

Esempio

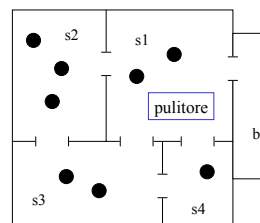


Relazione dinamica, tempo reificato.

sporco(s1,2,0).
sporco(s2,3,0).
sporco(s3,2,0).
sporco(s4,1,0).
in(s1,pulitore,0).
in(s2,pulitore,1).
sporco(s1,0,1).
.....

Query ammessa:
? sporco(S,0,T)

Esempio



Relazione dinamica, tempo reificato, un possibile uso di holds.

holds(sporco, [s1,2], 0).
holds(sporco, [s2,3], 0).
holds(sporco, [s3,2], 0).
holds(sporco, [s4,1], 0).
holds(in, [s1, pulitore], 0).
holds(in, [s2, pulitore], 1).
holds(sporco, [s1,0], 1).
.....

Query ammessa:
? holds(P, [s1|Q], 1)