

LEZIONE 4

PLANNING II

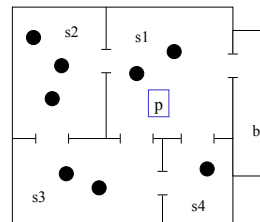
STRIPS, Situation Calculus, Event Calculus

- Riprendendo le considerazioni generali della lezione scorsa, vediamo tre casi di rappresentazione
 - STRIPS (Stanford Research Institute Problem Solver)
 - Situation Calculus (McCarty & Hayes, 1969, ...)
 - Event Calculus (Kowalski & Sergot, 1986, ...)
- È bene distinguere fra rappresentazione e pianificazione
 - pianificatore: algoritmo che determina un piano, ovvero una successione di azioni che porta ad un goal
 - uno stesso pianificatore può essere applicato a rappresentazioni diverse

1. Rappresentazione di STRIPS

- Tempo basato sugli stati.
 - il tempo non è reificato;
 - si hanno i predicati che consentono di descrivere un qualsiasi stato (una "istantanea")
- Rappresentazione delle azioni:
 - precondizione [Body] che deve aver successo nello stato corrente perché l'azione sia possibile
 - delete list [F1,...,Fn] diventano falsi
 - add list [A1,...,An] diventano veri
- Assunzione STRIPS (assioma d'inerzia)
 - se il valore di verità di un fatto non è alterato da un'azione, non cambia

Esempio



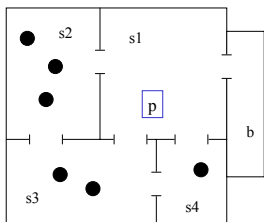
Relazioni statiche
stanza(s1).
.....
da(s1,s2).
.....

Relazioni dinamiche STRIPS.
sporco(s1,2).
sporco(s2,3).
sporco(s3,2).
sporco(s4,1).
in(s1,p).
pulizia(p).

Esegui: **pulisci(p,s1)**

azione pulisci(P,S): pre: [stanza(S),pulizia(P),in(S,P),sporco(S,N)]
del: [pulizia(P), sporco(S,N)]
add: [spostamento(P), sporco(S,0)]

Esempio



Relazioni statiche
stanza(s1).
.....
da(s1,s2).
.....

Relazioni dinamiche STRIPS.
sporco(s1,0).
sporco(s2,3).
sporco(s3,2).
sporco(s4,1).
in(s1,p).
spostamento(p).

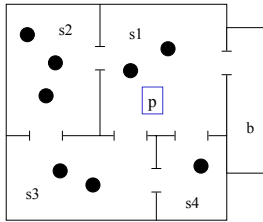
Esegui: **pulisci(p,s1)**

azione pulisci(P,S): pre: [stanza(S),pulizia(P),in(S,P),sporco(S,N)]
del: [pulizia(P), sporco(S,N)]
add: [spostamento(P), sporco(S,0)]

2. Rappresentazione del situation calculus

- Il tempo è rappresentato dagli stati, detti situazioni.
 - Ogni situazione è determinata da un termine:
 - init rappresenta la situazione iniziale
 - do(A,S) la situazione raggiunta con l'azione A a partire da S
 - Una relazione dinamica r(X) in una situazione S è rappresentata da:
 - holds(r(X),S) e varianti, oppure
 - r(X,S)
- a seconda delle convenienze

Esempio



Relazioni statiche
stanza(s1).
....
da(s1,s2).
....

Relazioni dinamiche STRIPS.

sporco(s1,2,init).
sporco(s2,3,init).
sporco(s3,2,init).
sporco(s4,1,init).
in(s1,p,init).
pulizia(p,init).

$in(s2,p,do(va(p,s1,s2),init))$

2.1.Codifica delle azioni e del loro effetto

- Può convenire l'uso di un meta-predicato `poss(A,S)`
 - A è possibile nello stato S
 - ad esempio:
`poss(va(P,A,B),S) :- in(A,P,S), sporco(S,0).`
- L'effetto di un'azione indica il nuovo valore della relazione dinamica modificata:
 - con il meta-predicato `holds` è possibile dare la legge di cambiamento:
`holds(P,do(A,S)) :- poss(A,S),
toTrue(P,A,S).`
 - e poi codificare `toTrue`.
 - Assumendo il mondo chiuso, `toFalse` vale per default se non si indica `toTrue`

Il frame problem

- MA:
- Il principio di inerzia richiede che ciò che esplicitamente non cambia resta invariato:
- ciò che era vero e deve restare vero
 - gli assiomi che mantengono veri gli atomi prima veri ed invariati son detti frame axioms
 - sono costosi (poche parti cambiano, molte restano invariate)
 - frame problem: come convivere con questo problema

ESEMPIO. Per ogni predicato un frame axiom. Ad es. per in:

```
/* frame axiom:*/
in(St,P,do(A,S)) :- nonMod(A,in), poss(A,S), in(St,P,S).
in(St1,P,do(va(P,St,St1),S)) :- poss(va(P,St,St1),S).
....
```

`nonMod(A,in) :- not(A = va(____)).`

Con il metapredicato `holds`, un unico frame axiom

```
holds(Pred,do(A,S)) :- nonMod(A,Pred),
poss(A,S),
holds(Pred,S).
```

3. Rappresentazione dell'event calculus

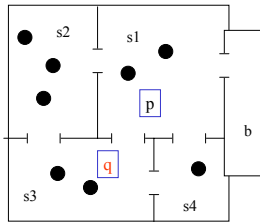
- Tempo orologio, possibile intervalli di tempo
- Meta-rappresentazione delle relazioni e degli eventi nel tempo:
 - ordinamento < sui tempi
 - `holds(P,T)`: P vale al tempo T
 - `event(E,T)`: si ha l'evento E al tempo T
 - `inziates(E,P,T)`: l'evento E al tempo T rende vero T
 - `terminates(E,P,T)`: l'evento E al tempo T rende falso P
 - `clipped(P,T1,T2)`: nell'intervallo (T1..T2) occorre un evento che rende falso P

La meta-teoria dell'event calculus (dal libro):

```
holds(P,T) :- event(E,T0),
T0 < T,
inziates(E,P,T0),
not(clipped(P,T0,T)).
```

```
clipped(P,T1,T2) :- event(E,T1),
terminates(E,P,T0),
T0 < T1,
T1 < T2.
```

Esempio



Relazioni statiche

Relazioni dinamiche EC.

holds(sporco(s1,2),0).
holds(sporco(s2,3),0).
holds(sporco(s3,2),0).
holds(sporco(s4,1),0).
holds(in(s1,p),0).
holds(in(s3,q),0)

terminates(pulisci(P,S),sporco(S,N),T) :- poss(pulisci(P,S),T).
initiates(pulisci(P,S),sporco(S,0,T)):- poss(pulisci(P,S),T).

poss(pulisci(P,S),T) :- holds(in(S,P),T),
holds(sporco(S,N),T),
N > 0.

terminates(va(P,S1,S2),in(S1,P),T) :- poss(va(P,S1,S2),T).
initiates(va(P,S1,S2),in(S2,P),T):- poss(va(P,S1,S2),T).

poss(va(P,S1,S2),T) :- holds(in(S1,P),T),
holds(sporco(S1,0),T).

Relazioni statiche
Relazioni dinamiche EC.

holds(sporco(s1,2),0).
holds(sporco(s2,3),0).
holds(sporco(s3,2),0).
holds(sporco(s4,1),0).
holds(in(s1,p),0).
holds(in(s3,q),0)

event(pulisci(p,s1),1)
event(pulito(p,s1),3)
event(pulisci(q,s3),1)
event(pulito(q,s3),2)
event(va(q,s4,s1),3)
event(va(p,s1,s2),4)

ESERCIZIO: dare holds ai tempi 1,2,3,4

4. Breve confronto

- STRIPS e situation calculus hanno tempo basato sugli stati
 - situation calculus più espressivo di STRIPS: è possibile definire STRIPS nel situation calculus, non viceversa
- Event calculus:
 - tempo orologio
 - possibile ragionare su tempo discreto, continuo, intervalli
 - nel caso di più agenti, vi è un tempo comune ed eventi paralleli sono possibili, mentre nel caso basato su stati è necessario interleaving