

Esercizio planning lineare

Si consideri uno stato iniziale descritto dalle seguenti formule atomiche:

```
[state(camera_one,on) , state(camera_two,on) , state(camera_three,on) ,  
in(camera_one, roomB) , in(camera_two, roomB) , in(camera_three, roomB) ,  
object_in(oggetto1, roomA) ]
```

Inoltre lo stato iniziale contiene predicati diversi per ogni coppia di parametri che rappresentano una stanza o una macchina fotografica. Es. diverso(camera_one, camera_two)

Si supponga di voler raggiungere il goal:

take_picture(oggetto1)

Le azioni sono modellate opportunamente come segue:

make_photo(Object)

PRECOND: object_in(Object, Room), in(CameraX,Room), in(CameraY,Room),
state(CameraX, on), state(CameraY, on), diverso(CameraX, CameraY)

EFFECT: take_picture(Object)

move_camera(Camera, Room1, Room2)

PRECOND: in(Camera, Room1)

EFFECT: in(Camera, Room2), not in(Camera, Room1)

Si risolva il problema utilizzando l'algoritmo STRIPS.

STATO state(camera_one,on) state(camera_two,on) state(camera_three,on) in(camera_one, roomB) in(camera_two, roomB) in(camera_three, roomB) object_in(oggetto1, roomA) predicate di diverso	GOAL take_picture(oggetto1)
--	--------------------------------

Inserisco l'azione make_photo

STATO state(camera_one,on) state(camera_two,on) state(camera_three,on) in(camera_one, roomB) in(camera_two, roomB) in(camera_three, roomB) object_in(oggetto1, roomA)	GOAL object_in(oggetto1, Room) in(CameraX, Room) in(CameraY, Room) diverso(CameraX, CameraY) state(CameraX, on) state(CameraY, on) AND DI GOAL <u>make_photo(oggetto1)</u>
--	--

Unifico Room / roomA

STATO state(camera_one,on) state(camera_two,on) state(camera_three,on) in(camera_one, roomB) in(camera_two, roomB) in(camera_three, roomB) object_in(oggetto1, roomA)	GOAL in(CameraX, roomA) in(CameraY, roomA) diverso(CameraX, CameraY) state(CameraX, on) state(CameraY, on) AND DI GOAL <u>make_photo(oggetto1)</u>
--	---

Uso l'azione move

STATO state(camera_one,on) state(camera_two,on) state(camera_three,on) in(camera_one, roomB) in(camera_two, roomB) in(camera_three, roomB) object_in(oggetto1, roomA)	GOAL in(CameraX, RoomX) <u>move(CameraX, RoomX, roomA)</u> in(CameraY, roomA) diverso(CameraX, CameraY) state(CameraX, on) state(CameraY, on) AND DI GOAL <u>make_photo(oggetto1)</u>
--	---

CameraX / camera_one

RoomX / roomB

STATO	GOAL
state(camera_one,on)	<u>move(camera one,roomB,roomA)</u>
state(camera_two,on)	in(CameraY,roomA)
state(camera_three,on)	diverso(camera_one,CameraY)
in(camera_one, roomB)	state(camera_one,on)
in(camera_two, roomB)	state(CameraY,on)
in(camera_three, roomB)	AND DI GOAL
object_in(oggetto1,roomA)	<u>make photo(oggetto1)</u>

Effettuo l'azione move e modifico lo stato

STATO	GOAL
state(camera_one,on)	
state(camera_two,on)	
state(camera_three,on)	in(CameraY,roomA)
in(camera_one, roomA)	diverso(camera_one,CameraY)
in(camera_two, roomB)	state(camera_one,on)
in(camera_three, roomB)	state(CameraY,on)
object_in(oggetto1,roomA)	AND DI GOAL
	<u>make photo(oggetto1)</u>

inserisco la azione move

STATO	GOAL
state(camera_one,on)	in(CameraY,RoomX)
state(camera_two,on)	<u>move(CameraY,RoomX,roomA)</u>
state(camera_three,on)	
in(camera_one, roomA)	diverso(camera_one,CameraY)
in(camera_two, roomB)	state(camera_one,on)
in(camera_three, roomB)	state(CameraY,on)
object_in(oggetto1,roomA)	AND DI GOAL
	<u>make photo(oggetto1)</u>

Unifico CameraY / camera_two e RoomX / roomB

STATO	GOAL
state(camera_one,on)	
state(camera_two,on)	<u>move(camera two,roomB,roomA)</u>
state(camera_three,on)	
in(camera_one, roomA)	diverso(camera_one, camera_two)
in(camera_two, roomB)	state(camera_one,on)
in(camera_three, roomB)	state(camera_two,on)
object_in(oggetto1,roomA)	AND DI GOAL
	<u>make photo(oggetto1)</u>

Effettuo l'azione move

STATO	GOAL
<code>state(camera_one,on)</code> <code>state(camera_two,on)</code> <code>state(camera_three,on)</code> <code>in(camera_one, roomA)</code> <code>in(camera_two, roomA)</code> <code>in(camera_three, roomB)</code> <code>object_in(oggetto1,roomA)</code>	<code>diverso(camera_one, camera_two)</code> <code>state(camera_one,on)</code> <code>state(camera_two,on)</code> AND DI GOAL <u><code>make_photo(oggetto1)</code></u>

Tutte le precondizioni prima di `make_photo` sono vere nello stato iniziale

STATO	GOAL
<code>state(camera_one,on)</code> <code>state(camera_two,on)</code> <code>state(camera_three,on)</code> <code>in(camera_one, roomA)</code> <code>in(camera_two, roomA)</code> <code>in(camera_three, roomB)</code> <code>object_in(oggetto1,roomA)</code>	<code>make_photo(oggetto1)</code>

Effettuo `make_photo` e ottengo il goal

STATO	GOAL
<code>state(camera_one,on)</code> <code>state(camera_two,on)</code> <code>state(camera_three,on)</code> <code>in(camera_one, roomA)</code> <code>in(camera_two, roomA)</code> <code>in(camera_three, roomB)</code> <code>object_in(oggetto1,roomA)</code> <code>take_picture(oggetto1)</code>	

Esercizio Planning non lineare

È dato lo stato iniziale descritto dalle seguenti formule atomiche:

[at(location1), have_battery, handempty]

- le azioni sono modellate opportunamente come segue:

take_picture(Location)

PRECOND: have_battery, at(Location), have_camera

DELETE: have_battery

ADD: picture(Location)

putdown_camera

PRECOND: have_camera

DELETE: have_camera

ADD: handempty

pickup_camera

PRECOND: handempty

DELETE: handempty

ADD: have_camera

charge_battery

PRECOND: not have_battery

DELETE: -

ADD have_battery

go(Location1, Location2)

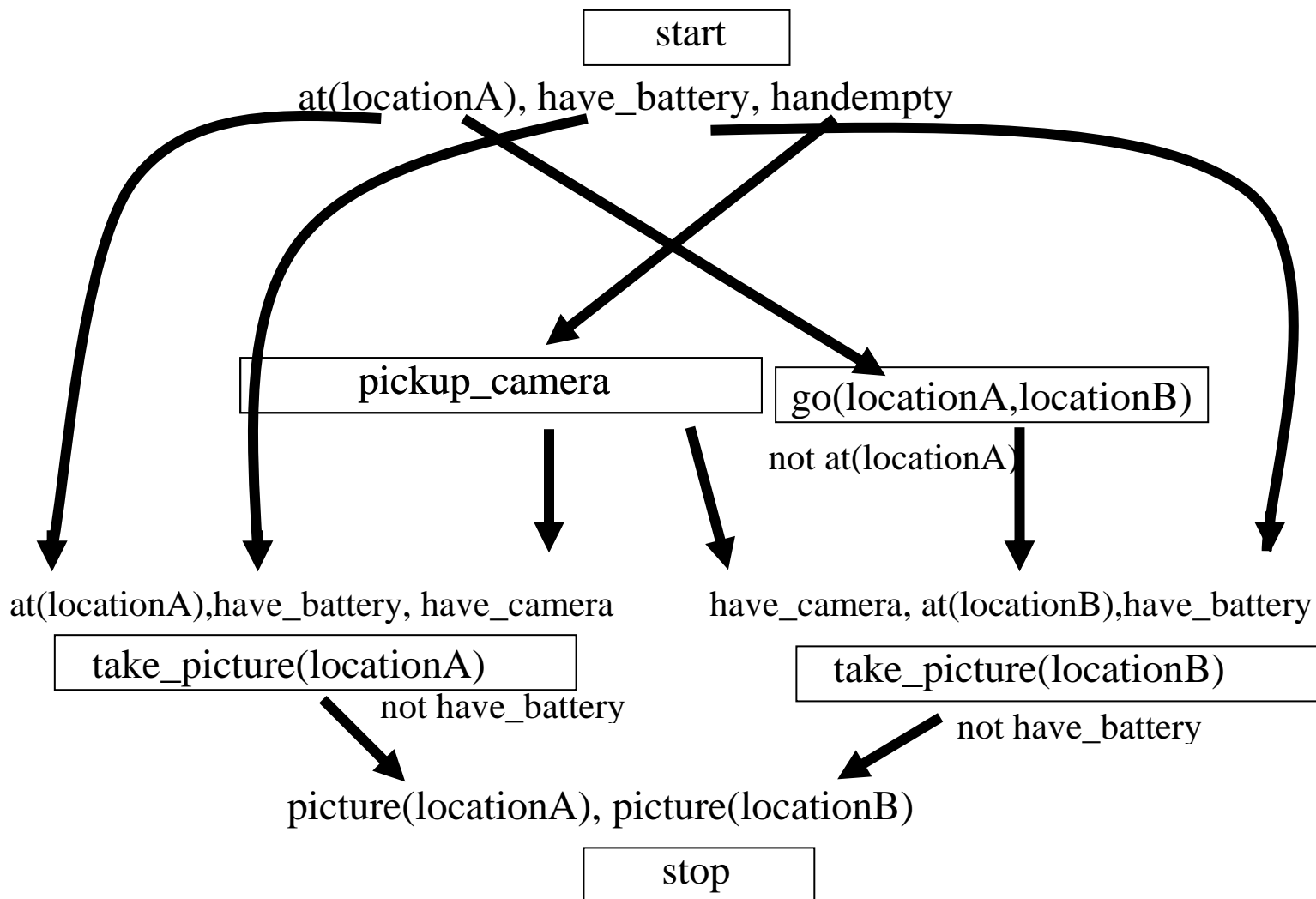
PRECOND: at(Location1)

DELETE: at(Location1)

ADD at(Location2)

e il goal **picture(locationA), picture(locationB)**

Si risolva il problema utilizzando l'algoritmo POP. Si evidenzino i causal link e le minacce incontrate.



Il piano fino a qui creato contiene delle minacce in particolare i causal link $\langle \text{Start}, \text{take_picture}(\text{locationB}), \text{have_battery} \rangle$ e $\langle \text{Start}, \text{take_picture}(\text{locationA}), \text{have_battery} \rangle$ sono minacciati rispettivamente da due azioni $\text{take_picture}(\text{locationA})$ e $\text{take_picture}(\text{locationB})$ rispettivamente. Nessun vincolo di ordinamento riesce a risolvere questo conflitto: e' necessario procedere all'inserimento di una azione di charge_battery

Inoltre, l'azione $\text{go}(\text{locationA}, \text{locationB})$ minaccia il causal link

$\langle \text{Start}, \text{take_picture}(\text{locationA}), \text{at}(\text{locationA}) \rangle$

In questo caso pero' la promotion dell'azione puo' risolvere il conflitto. Introduciamo quindi un ordinamento tra $\text{take_picture}(\text{locationA})$ e $\text{go}(\text{locationA}, \text{locationB})$.

