

## II Esonero di RCLC: Fondamenti del 22/06/2017

- 1 Descrivere l'algoritmo di planning SATPlan. Illustrare il funzionamento dell'algoritmo sul seguente planning domain e mostrare la sequenza di passi necessaria per provare il goal. Dare, gli assiomi fra lo stato  $i$  e lo stato  $i + 1$ .

Init(*On\_Floor*;  $\neg$ *Broken*;  $\neg$ *Holding*)  
Goal( $\neg$ *On\_Floor*; *Broken*)

Action(pick\_Up,  
PRECOND: *On\_Floor*  $\wedge$   $\neg$ *Holding*  
EFFECT: *Holding*  $\wedge$   $\neg$ *On\_Floor*

Action(put\_Down,  
PRECOND: *Holding*  
EFFECT: *On\_Floor*  $\wedge$   $\neg$ *Holding*

Action(throw,  
PRECOND: *Holding*  
EFFECT: *On\_Floor*  $\wedge$  *Broken*  $\wedge$   $\neg$ *Holding*

E' possibile che un piano restituito dall'algoritmo eseguito su questo esempio contenga azioni parallele? Se sì, come posso specificare che non ci devono essere azioni parallele?

- 2 Data la KB=(TBox, ABox), dove TBox contiene l'assioma  $Student \sqsubseteq Young$  e ABox contiene gli assiomi:

$\forall has\_Friend.(Student \sqcap Child)(alice)$   
 $\exists has\_Friend.(\neg Student \sqcap \neg Child)(bob)$   
 $has\_Friend(alice, bob)$   
 $(\neg Student)(bob)$   
 $Student(alice)$

descrivere il significato degli assiomi e dire se:

- (1)  $KB$  è consistente;
- (2)  $KB \models Child(bob)$
- (3)  $KB \models (Student \sqcap Child)(alice)$

(4) Se aggiungo alla KB l'asserzione  $has\_Friend(bob, alice)$ , la KB è ancora consistente?

Motivare le risposte.

- 4 Dare la definizione di *answer set* di un programma logico  $P$ . Calcolare gli answer sets del programma  $P$ :

$$\begin{aligned}p &\leftarrow not\ q, r \\q &\leftarrow not\ s, r \\r\end{aligned}$$

## Compito di RCLC: Fondamenti del 22/06/2017 - I parte

- 1 Descrivere l'algoritmo di Partial Order Planning. Eseguire alcuni passi dell'algoritmo sul problema descritto nel seguito. Spiegare cos'è un causal link, cosa è un conflitto e cosa significa risolvere un conflitto; fare un esempio relativo al problema di planning dato. Trovare la configurazione finale (il piano).

Init(*On\_Floor*;  $\neg$ *Broken*;  $\neg$ *Holding*)  
Goal( $\neg$ *On\_Floor*; *Broken*)

Action(pick\_Up,  
PRECOND: *On\_Floor*  $\wedge$   $\neg$ *Holding*  
EFFECT: *Holding*  $\wedge$   $\neg$ *On\_Floor*

Action(put\_Down,  
PRECOND: *Holding*  
EFFECT: *On\_Floor*  $\wedge$   $\neg$ *Holding*

Action(throw,  
PRECOND: *Holding*  
EFFECT: *On\_Floor*  $\wedge$  *Broken*  $\wedge$   $\neg$ *Holding*

- 2 Data la descrizione del problema di planning nell'Esercizio 1, trovare un piano mediante l'esecuzione dell'algoritmo di Regression e, in particolare, descrivere graficamente nei dettagli la configurazione finale.

- 3 Dato il programma:

father(ann,bob).  
father(bob,susy).  
parent(X,Y):- father(X,Y).  
parent(X,Y):- mother(X,Y).  
ancestor(X,Y):- parent(X,Y).  
ancestor(X,Y):- parent(X,Z), ancestor(Z,Y).

per la query "ancestor(john,N)" costruire l'albero delle possibili risoluzioni SLD (compresi i rami che non determinano soluzioni) ed indicare le risposte che si ottengono.

Cosa significa che la procedura è completa per le clausole di Horn? Cosa significa che è corretta?

- 4 Descrivere la Regola di Risoluzione. Fare un esempio di derivazione utilizzando le clausole:

$A \vee B \vee \neg C, C \vee B, \neg B \vee A, \neg A$

E' possibile derivare la clausola vuota?

Cosa sono la risoluzione lineare e la risoluzione input? La risoluzione SLD è input, lineare o entrambe?