

Domanda 1

Completo

Punteggio
ottenuto 0,00 su
3,00Contrassegna
domanda

Il GOLOG è un linguaggio di programmazione che fa uso del situation calculus implementato con la logica classica. Spiegare il significato dei "Successor state axioms" facendo riferimento al seguente esempio:

$$\begin{aligned} Poss(a,s) \Rightarrow [broken(x,do(a,s)) \equiv \\ \exists r \{ a = drop(r,x) \wedge fragile(x,s) \} \vee \\ broken(x,s) \wedge \neg \exists r \{ a = repair(r,x) \}] \end{aligned}$$

le azioni in GOLOG, presentato nel 1994, sono specificate dando delle precondizioni e degli effetti.

I Successor state axioms vengono usati per esplicitare gli effetti delle azioni.

Commento:

Domanda 2

Completo

Punteggio
ottenuto 1,50 su
2,00Contrassegna
domanda

Nella logica temporale Branching-Time il tempo è modellato con un albero infinito su cui si possono formulare *path formulas* o *state formulas*. Dire se le formule $A\pi$ e $E\pi$ sono *path* o *state formulas* e spiegare il loro significato.

Le state formula sono le seguenti:

- 1) p
- 2) alfa OR Beta
- 3) not alfa
- 4) $A(\pi)$
- 5) $E(\pi)$

Dove alfa e' una state formula mentre pi si riferisce alle path formula.

Le path formula sono invece:

- 1) alfa
- 2) π OR p
- 3) not π
- 4) $X\pi$
- 5) π U p

Dove alfa e' uno state formula mentre π e p sono path formula

$A(\pi)$ significa "per tutti i futuri", ovvero viene usato per controllare tutti i percorsi (partendo da uno stato iniziale)

$E(\pi)$ significa "per qualche futuro", ovvero viene usato per controllare solo alcuni percorsi (partendo da uno stato iniziale).

$A(\pi)$ e $E(\pi)$ sono duali.

Commento:

Domanda 3

Completo

Punteggio
ottenuto 2,00 su
2,00



Contrassegna
domanda

Commentare la formula che definisce un **persistent goal** secondo Cohen e Levesque.

$$\begin{aligned} (\text{P-GOAL } i \varphi) \equiv & (\text{GOAL } i (\text{LATER } \varphi)) \wedge (\text{BEL } i \neg \varphi) \wedge \\ & [\text{BEFORE } ((\text{BEL } i \varphi) \vee (\text{BEL } i \mathbf{G} \neg \varphi)) \\ & \neg(\text{GOAL } i (\text{LATER } \varphi))] \end{aligned}$$

un Goal persistente e' un achievement goal che puo' essere rimosso solo per due motivi: o l'agente lo ha raggiunto, oppure si e' reso conto che non e' mai raggiungibile.

Piu' formalmente possiamo dire: Ha un goal φ che possibilmente diventera' vero e crede che φ non e' attualmente vera.

Prima che il goal vada in "drop" una delle seguenti condizioni deve essere vera:

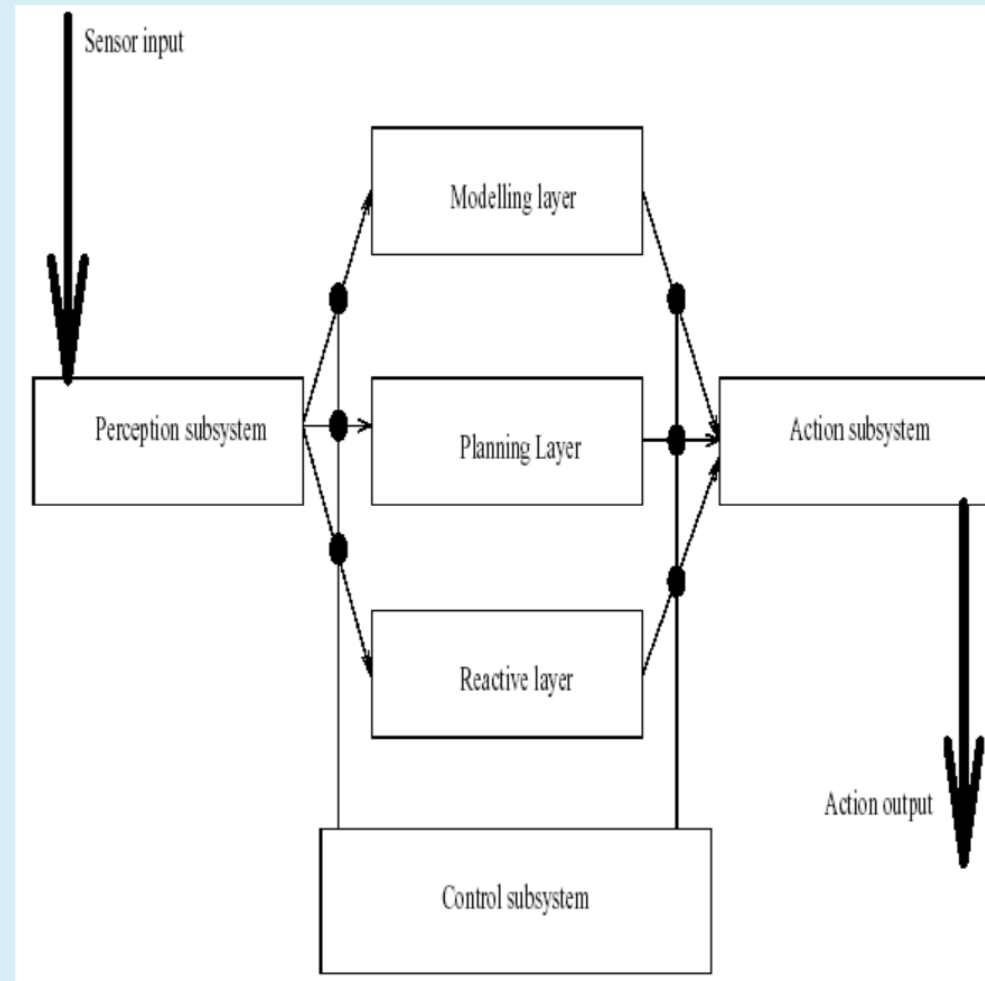
- 1) l'agente crede che il goal sia soddisfatto.
- 2) l'agente crede che il goal non sara' mai soddisfatto.

Domanda 4

Completo

Punteggio
ottenuto 1,50 su
2,00Contrassegna
domanda

Descrivere i componenti della architettura **TouringMachines** rappresentata nella figura seguente. In particolare, dire se si tratta di una architettura orizzontale o verticale, e spiegare il significato del "control subsystem".



La Touring Machine è una architettura a stratificazione orizzontale consistente di una struttura con sottosistemi di percezione e di azione che si interfacciano direttamente con l'ambiente e con tre livelli di controllo racchiusi in un framework di controllo che fa da mediatore tra i livelli.

Il reactive layer è implementato come un set di regole di situazioni-azioni.

Il planning layer costruisce piani e seleziona azioni da eseguire in ordine per completare i goal dell'agente.

Il modelling layer contiene una rappresentazione simbolica del "cognitive state" delle altre entità dell'ambiente dell'agente.

Tutti e tre i livelli comunicano tra loro e sono racchiusi in un framework di controllo (Control subsystem) che usa regole di controllo.

Domanda 5Parzialmente
correttaPunteggio
ottenuto 2,00 su
4,00Contrassegna
domanda

La seguente matrice di payoff si riferisce al “game of chicken” in cui i due giocatori **I** e **J** guidano ciascuno la propria auto verso un precipizio. Ogni giocatore può sterzare o andare dritto.

	I : dritto	I : sterza
J : dritto	I = 0, J = 0	I = 1, J = 3
J : sterza	I = 3, J = 1	I = 2, J = 2

Dire quali delle seguenti affermazioni sono vere.

Scegli una o più alternative:

- ☐ a. [I : dritto, J : dritto] è un *equilibrio di Nash*.
- ☐ b. [I : dritto, J : sterza] è un *equilibrio di Nash*.
- ☐ c. “sterza” è una *strategia dominante* per I.
- ☒ d. [I : sterza, J : sterza] è *Pareto optimal*. ✓

Risposta parzialmente esatta.

Hai selezionato correttamente 1.

La risposta corretta è: [I : dritto, J : sterza] è un *equilibrio di Nash*., [I : sterza, J : sterza] è *Pareto optimal*.

Domanda 6Parzialmente
correttaPunteggio
ottenuto 2,00 su
3,00Contrassegna
domanda

Per modellare con la logica modale diverse proprietà degli agenti può essere necessario modificare la logica aggiungendo degli **assiomi** o, in alternativa, restringere la classe dei modelli formulando delle **proprietà** dei frame. In diversi casi la formulazione con assiomi **corrisponde** alle proprietà dei frame.

Dati i seguenti assiomi e proprietà:

$$(a1) \boxed{\Box} \varphi \Rightarrow \varphi$$

$$(a2) \boxed{\Box} \varphi \Rightarrow \Diamond \varphi$$

$$(p1) \forall w \in W . \exists w' \in W . (w, w') \in R$$

$$(p2) \forall w \in W . (w, w) \in R$$

dire quali delle seguenti affermazioni sono vere.

Scegli una o più alternative:

☐ a.

L'assioma (a2) corrisponde alla proprietà (p2).

☒ b.

L'assioma (a1) corrisponde alla proprietà (p2). ✓

☐ c.

L'assioma (a1) corrisponde alla proprietà (p1).

☐ d.

La proprietà (p2) modella la serialità.

☒ e.

L'assioma (a2) corrisponde alla proprietà (p1). ✓

☐ f.

La proprietà (p2) modella la riflessività.

Risposta parzialmente esatta.

Hai selezionato correttamente 2.

La risposta corretta è:

La proprietà (p2) modella la riflessività.,

L'assioma (a1) corrisponde alla proprietà (p2).,

L'assioma (a2) corrisponde alla proprietà (p1).

Domanda 8

Risposta corretta

Punteggio
ottenuto 2,00 su
2,00



Contrassegna
domanda

Riordinare le seguenti istruzioni di un Agent Control Loop.

1. ✓
2. ✓
3. ✓
4. ✓
5. ✓
6. ✓
7. ✓
8. ✓
9. ✓

Risposta corretta.

La risposta corretta è: 1. - $B := B0; I := I0$; 2. - while true do, 3. - get next percept, 4. - $B := \text{brf}(B, \rho)$, 5. - $D := \text{options}(B, I)$, 6. - $I := \text{filter}(B, D, I)$, 7. - $\pi := \text{plan}(B, I)$, 8. - $\text{execute}(\pi)$, 9. - end while

Domanda 9

Parzialmente
corretta

Punteggio
ottenuto 0,33 su
2,00



Contrassegna
domanda

Si consideri lo scheduling dei comportamenti di un agente Jade. Quali delle seguenti affermazioni sono vere?

Scegli una o più alternative:

- ☒ a. La funzione di `setup()` è invocata una sola volta. ✓
- ☐ b. Il metodo `action()` è eseguito completamente ogni volta che il comportamento è selezionato dal pool dei comportamenti attivi dell'agente.
- ☐ c. Un comportamento ha tanti metodi `action()` quanti sono i task indipendenti che tale comportamento deve raggiungere.
- ☐ d. Il metodo `action()` è eseguito parzialmente sino ad una condizione di sincronizzazione con altro comportamento presente pool dei comportamenti attivi dell'agente.
- ☒ e. Un comportamento ciclico non valuta mai il risultato dell'esecuzione del metodo `done()`. ✗
- ☒ f. Un comportamento risiede nel pool dei comportamenti attivi fino all'esecuzione del metodo `takeDown()`. ✗
- ☒ g. Il valore restituito dal metodo `done()` determina se un comportamento deve o no essere inserito nel pool dei comportamenti attivi dell'agente. ✓

Risposta parzialmente esatta.

Hai selezionato troppe opzioni.

La risposta corretta è: La funzione di `setup()` è invocata una sola volta., Il metodo `action()` è eseguito completamente ogni volta che il comportamento è selezionato dal pool dei comportamenti attivi dell'agente., Il valore restituito dal metodo `done()` determina se un comportamento deve o no essere inserito nel pool dei comportamenti attivi dell'agente.

Domanda 10

Risposta corretta

Punteggio
ottenuto 2,00 su
2,00Contrassegna
domanda

Si consideri il seguente frammento di codice Jade:

```
public class MyThreeStepBehaviour extends Behaviour {  
    private int step = 0;  
    public void action() {  
        // istruzione1  
        switch (step) {  
            case 0: // perform operation X  
                step++;  
                break;  
            case 1: // perform operation Y  
                step++;  
                break;  
            case 2: // perform operation Z  
                step++;  
                break;  
        }  
        // istruzione2  
    }  
    public boolean done() {  
        return step == 3;  
    }  
}
```

Quali delle seguenti affermazioni sono vere?

Scegli una o più alternative:

- ☐ a. istruzione1 viene eseguita 1 volta.
- ☒ b. istruzione1 e istruzione2 vengono eseguite 3 volte. ✓
- ☐ c. istruzione2 non viene mai eseguita.
- ☐ d. istruzione1 e istruzione2 vengono eseguite 1 volta.
- ☐ e. istruzione1 non viene mai eseguita.

Risposta corretta.

La risposta corretta è: istruzione1 e istruzione2 vengono eseguite 3 volte.

Domanda 11

Risposta non data

Punteggio max.:
3,00Contrassegna
domanda

Si consideri il seguente frammento di codice Jason:

a(1).

a(2).

b(2).

h(X) : a(X) & b(Y) & X = Y <- body1. (p1)

h(3) : a(X) & b(Y) & X = Y <- body2, (p2)

h(X) : a(X) & b(Y) & X = Y <- ?b(1), body3. (p3)

Quali sono le affermazioni vere?

Scegli una o più alternative:

- ☐ a. (p1) è il solo piano rilevante per l'evento +!h(2).
- ☐ b. (p1) e (p3) sono piani rilevanti per l'evento +!h(2).
- ☐ c. (p1) è il solo piano applicabile per l'evento +!h(2).
- ☐ d. (p1) e (p3) sono piani rilevanti per l'evento +!h(1).
- ☐ e. Nessun piano rilevante per l'evento +!h(3).
- ☐ f. Nessun piano applicabile per l'evento +!h(3).
- ☐ g. (p2) è il solo piano rilevante per l'evento +!h(3).
- ☐ h. (p1) e (p2) sono piani rilevanti per l'evento +!h(2).
- ☐ i. (p1) e (p3) sono piani applicabili per l'evento +!h(2).
- ☐ j. (p1), (p2) e (p3) sono piani rilevanti per l'evento +!h(1).
- ☐ k. Non esiste un piano applicabile per l'evento +!h(1).
- ☐ l. (p1) è un piano rilevante per l'evento +!h(1).
- ☐ m. (p2) è il solo applicabile per l'evento +!h(3).
- ☐ n. (p1) è un piano applicabile per l'evento +!h(1).

Risposta errata.

La risposta corretta è: (p1) e (p3) sono piani rilevanti per l'evento +!h(1)., Non esiste un piano applicabile per l'evento +!h(1)., (p1) e (p3) sono piani rilevanti per l'evento +!h(2)., (p1) e (p3) sono piani applicabili per l'evento +!h(2)., (p2) è il solo piano rilevante per l'evento +!h(3)., (p2) è il solo applicabile per l'evento +!h(3).

Domanda 12

Risposta corretta

Punteggio
ottenuto 2,00 su
2,00Contrassegna
domanda

Si consideri la definizione di practical reasoning, quali delle seguenti affermazioni sono vere?

Scegli una o più alternative:

- ☒ a. La fase di means-ends reasoning consiste nel decidere come uno stato di interesse deve essere raggiunto. ✓
- ☒ b. La fase di deliberazione consiste nel decidere quale stato di interesse deve essere raggiunto. ✓
- ☐ c. La fase di deliberazione consiste nel decidere come uno stato di interesse deve essere raggiunto.
- ☒ d. La fase di deliberazione precede la fase di means-ends reasoning. ✓
- ☐ e. La fase di means-ends reasoning consiste nel decidere quale stato di interesse deve essere raggiunto.

Risposta corretta.

La risposta corretta è: La fase di deliberazione consiste nel decidere quale stato di interesse deve essere raggiunto., La fase di means-ends reasoning consiste nel decidere come uno stato di interesse deve essere raggiunto., La fase di deliberazione precede la fase di means-ends reasoning.

Domanda 13Parzialmente
correttaPunteggio
ottenuto 1,33 su
2,00Contrassegna
domanda

Si consideri il seguente protocollo a commitment:

cfp means create($C(i, p, \text{propose}, \text{accept v reject})$)

accept means none

reject means release($C(p, i, \text{accept}, \text{done v failure})$)

propose means create($C(p, i, \text{accept}, \text{done v failure})$)

refuse means release($C(i, p, \text{propose}, \text{accept v reject})$)

done means none

failure means none

Quali delle seguenti affermazioni sono vere?

Scegli una o più alternative:

- ☐ a. Lo stato sociale vuoto è raggiunto dopo l'esecuzione della sequenza cfp e reject.
- ☐ b. Non è possibile che si verifichi la sequenza cfp e accept.
- ☒ c. Lo stato sociale è $\{C(i, p, T, \text{accept v reject}), C(p, i, \text{accept}, \text{done v failure})\}$ dopo l'esecuzione della sequenza cfp e propose. ✓
- ☐ d. Lo stato sociale è $\{C(i, p, T, \text{accept v reject}), C(p, i, \text{accept}, \text{done v failure})\}$ dopo l'esecuzione dell'azione propose.
- ☒ e. Lo stato sociale vuoto è raggiunto dopo l'esecuzione della sequenza cfp, propose, accept, done. ✓

Risposta parzialmente esatta.

Hai selezionato correttamente 2.

La risposta corretta è: Lo stato sociale è $\{C(i, p, T, \text{accept v reject}), C(p, i, \text{accept}, \text{done v failure})\}$ dopo l'esecuzione della sequenza cfp e propose., Lo stato sociale vuoto è raggiunto dopo l'esecuzione della sequenza cfp e reject., Lo stato sociale vuoto è raggiunto dopo l'esecuzione della sequenza cfp, propose, accept, done.

Domanda 14

Risposta corretta

Punteggio
ottenuto 1,00 su
1,00



Contrassegna
domanda

L'obiettivo del Knowledge Sharing Effort è di sviluppare tecniche, metodologie e strumenti software per la condivisione e il riuso di conoscenza.

Scegli una risposta:

- ☒ Vero ✓
- ☐ Falso

La risposta corretta è 'Vero'.