

Spiegare la differenza tra coordinazione, cooperazione e negoziazione

La coordinazione e' una proprieta di un insieme di agenti che eseguono attivita' in un ambiente comune. La cooperazione, invece, si verifica in una situazione di coordinazione in cui gli agenti non sono antagonisti, ma cooperano. La negoziazione e' una coordinazione in cui gli agenti sono competitivi o antagonisti.

Si considerino l'Agent Control Loop V3, V4 e V5. Quali problemi affronta v4 rispetto a v3 e quali problemi affronta v5 rispetto a v4 e come?

Nell'ACL v4 a differenza della v3 i piani non vengono eseguiti subito tutti ma viene aggiunto un ciclo che permette di eseguire e scomporre il piano (i piani sono gestiti come una lista di azioni), avendo la possibilita' di aggiornare per ogni azione eseguita le credenze in base alla percezione e ponendo anche un controllo effettuato ad ogni azione eseguita, dove si controlla se il piano e' ancora coerente con le credenze e le intenzioni per poterlo continuare ad eseguire. Nella v5, invece, a differenza della v4 sono stati aggiunti dei controlli al ciclo dell'esecuzione del piano, difatti ora da quel ciclo non si esce piu' solo se la lista dei piani e' vuota, ma anche se l'obiettivo e' stato raggiunto prima che possa finire il piano oppure se ci si e' accorti che l'obiettivo e' impossibile da raggiungere.

Nella logica epistemica si assume che per l'operatore K (conoscenza) valgano le seguenti proprieta':

1. La conoscenza e' non contraddittoria
2. Cio' che si conosce e' vero
3. Vale l'introspezione positiva

4. Vale l'introspezione negativa

Fornire i quattro assiomi che rappresentano le proprietà:

5. $K \phi \rightarrow \neg K \neg \phi$

6. $K \phi \rightarrow \phi$

7. $K \phi \rightarrow K K \phi$

8. $\neg K \neg \phi \rightarrow K (\neg K \neg \phi)$

Cosa si intende per strategia pura e strategia mista nella teoria dei giochi?

Per strategia pura si intende una politica deterministica che indica l'azione esatta (o le azioni esatte) da scegliere in ogni situazione. Quando non vi sono strategie pure che possono formare un equilibrio bisogna pesare le azioni in modo stocastico. Attraverso una visione probabilistica si ottiene una strategia mista, ovvero una politica non deterministica, che guida le scelte in funzione di probabilità

Spiegare che cosa si intende per strategia dominante nella teoria dei giochi. Si consideri la seguente matrice di payoff, dove sia Alice che Bob possono eseguire o l'azione a1 o a2. Spiegare se Alice ha una strategia dominante

	A a1	A a2
b a1	a:0,b:0	a:1,b:3
b a2	a:3,b:1	a:2,b:2

Una strategia dominante per un giocatore A domina qualsiasi altra strategia per A. Dato un giocatore A e date due strategie s1 e s2 per A, si dice che s1 domina s2 se il risultato di s1 è migliore di s2 indipendentemente dalla strategia dell'avversario (o degli avversari).

In questo caso, Alice non ha una strategia dominante, perché se Bob sceglie a_1 , allora per Alice l'azione migliore è a_2 , ma se Bob sceglie a_2 , allora per Alice l'azione migliore è a_1 .

Spiegare sinteticamente la motivazione e la struttura della subsumption architecture proposta da Brooks

L'architettura a subsumption è divisa gerarchicamente a livelli. Ogni livello rappresenta un task e ha un insieme di behaviours associati. Ogni behaviour è descritto da una coppia (c,a) dove c è la condizione formata da un insieme di percezioni e a è l'azione. Un behaviour scatta quando l'agente riceve una percezione p tale che p fa parte della condizione c . I behaviours sono prioritizzati in funzione dell'inibizione.

Le tesi di Brooks a favore di questa struttura sono legati all'idea che si possa costruire un comportamento intelligente senza dover rappresentare modelli complessi come quelli dell'IA simbolica e senza un ragionamento esplicito, ma semplicemente interagendo con l'ambiente.

Spiegare la differenza tra practical reasoning e theoretical reasoning

Il practical reasoning è un ragionamento guidato dalle azioni, mentre il theoretical reasoning è un ragionamento guidato dalle credenze.

Spiegare il significato delle "commitment rules" del linguaggio Agent-0, facendo riferimento al seguente esempio

```

(COMMIT
  (?agent REQUEST (DO ?time ?action))

  (AND (B ?now (Friend ?agent))
        (CAN myself ?action)
        (NOT ((CMT ?anyone) (DO ?time ?action))))

  (myself (DO ?time ?action))
)

```

Le "commitment rule" rappresentano i commitment che gli agenti si impongono di eseguire in una situazione. Le regole sono divise in 3 blocchi:

- il primo indica la condizione del messaggio
- il secondo indica la condizione mentale dell'agente
- il terzo indica il commitment dell'agente.

In questo esmpio abbiamo che l'agente afferma che:

Se un agente ?agent mi richiede di eseguire un'azione ?action al momento ?time, allora se io credo che adesso sono amico con ?agent e se io posso eseguire l'azione ?action e se non sono impegnato verso nessuno ad eseguire questa azione ?action al tempo ?time, allora mi impegno ad eseguire ?action al tempo ?time.

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

```

modelling layer

perception system -> planing layer    -> action subsystem

reactive layer

control subsystem

```

L'architettura TouringMachines segue uno layering orizzontale. Le percezioni sono rilevate dal perception system che si interfaccia con tutti e tre i layer di modelling, planning, e reactive. Vi e' anche una componente di controllo che permette la comunicazione tra i layer ed e' collegata sia al perception system e sia all'action subsystem. Il ruolo del sistema di controllo e' quello di fornire un insieme di regole che permettono di filtrare gli input verso uno specifico layer ed eventualmente stabilire le priorita' di attivazione. Il layer di modelling si occupa della rappresentazione dell'ambiente e dello stato mentale dell'agente. Il layer di planning, invece, si occupa di stabilire i piani e scegliere le azioni da eseguire in funzione dei goal dell'agente. Il layer reactive e' invece un particolare layer semplice che si occupa di reagire in fretta in certe condizioni potenzialmente catastrofiche, come per esempio, nel caso di un agente in movimento, tentare immediatamente di deviare la propria traiettoria in caso di una collisione imminente.

Commentare la formula che definisce un persistent goal secondo Cohen e evesque

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

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

Piu' formalmente possiamo dire: l'agente i ha un goal ϕ che possibilmente diventera' vero in futuro e crede che ϕ non e' attualmente vero.

Inoltre, prima che l'agente i abbandoni il goal una delle seguenti condizioni deve verificarsi:

- l'agente crede che ϕ e' soddisfatto
- l'agente crede che ϕ non sara' mai soddisfatto

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

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

- Vero: La fase di means-ends reasoning consiste nel decidere come uno stato di interesse deve essere raggiunto
- Vero: la fase di deliberazione consiste nel decidere quale stato di interesse deve essere raggiunto
- Falso: la fase di deliberazione consiste nel decidere come uno stato di interesse deve essere raggiunto
- Vero: la fase di deliberazione precede la fase di means-ends reasoning
- Falso: la fase di means-ends reasoning consiste nel decidere quale stato di interesse deve essere raggiunto

Il GOLOG e' un linguaggio di programmazione che fa uso del situation calculus implementato con la logica classica. Spiegare il significato dei successor state axioms facendo riferimento all'esempio

$$\text{POSS}(a,b) \Rightarrow [\text{broken}(x,\text{do}(a,s)) \equiv \\ \exists r\{a = \text{drop}(r,x) \wedge \text{fragile}(x,s)\} \vee \\ \text{broken}(x,s) \wedge \neg \exists r\{a = \text{repair}(r,x)\}]$$

Il successor state axiom è un assioma utilizzato nel calcolo delle situazioni. In particolare, viene utilizzato per formalizzare tutto ciò che dopo l'esecuzione di una azione non è cambiato.

L'esempio riportato esprime che se un oggetto X è "broken", il suo stato è risultato di due possibili azioni: o si è verificata una azione "drop"

l'oggetto era "fragile" oppure l'oggetto era già "broken" e non è stata eseguita alcuna azione "repair"

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

$A\pi$ e $E\pi$ sono state-formulas. Il loro significato e':

- $A\pi$ e' vera in un modello M e in uno stato S_0 se e solo se π e' vera in ogni cammino uscente da S_0
- $E\pi$ e' vera in un modello M e in uno stato S_0 se e solo se esiste un cammino uscente da S_0 in cui π e' vera

$A\pi$ e $E\pi$ sono formule duali e una puo' essere espressa in funzione dell'altra e viceversa.

Si consideri il modello di esecuzione dei comportamenti di un agente JADE. Quali affermazioni sono vere?

- ☐ La gestione dei comportamenti di un agente e' di tipo competitivo
- ☒ Quando un comportamento di un agente e' selezionato, il suo metodo action viene completamente eseguito fino al suo return
- ☒ Ogni comportamento di un agente definisce un metodo done
- ☐ La sincronizzazione tra i comportamenti di un agente e' ottenuta mediante le usuali primitive del linguaggio Java
- ☐ La gestione dei comportamenti di un agente e' di tipo pre-emptive

Cohen e Levesque hanno proposto una logica per modellare agenti BDI la cui

semantica e' basata su mondi possibili. Spiegare cosa hanno di diverso i modelli di questa logica rispetto ai modelli della logica modale classica

La logica proposta da Cohen e Levesque e' basata sull'idea del Belief-Desire-Intention e si basa su quattro operatori modali:

- $BEL\ i\ p$, l'agente i crede p
- $GOAL\ i\ p$, l'agente i ha come goal p
- $HAPPENS\ a$, l'azione a accadrà all'istante successivo
- $DONA\ a$, l'azione a e' appena stata eseguita

L'idea dei mondi possibili stabilisce che ogni mondo e' una sequenza infinita e discreta di eventi e i mondi possibili sono interconnessi attraverso relazioni di belief e goal. La semantica all'interno di ciascun mondo e' definita dalla semantica della linear time logic.

Gli operatori BEL e $GOAL$ e le loro interpretazioni sono definiti attraverso gli assiomi della logica modale classica.

Si consideri il seguente esempio (resource producer dalle slide) di Concurrent MetateM. Commentare le regole

rp (ask1, ask2) [give1, give2]

●ask1 $\Rightarrow \Diamond$ give1 commits to eventually give to any agent that asks

●ask2 $\Rightarrow \Diamond$ give2

start $\Rightarrow \Box \neg(\text{give1} \wedge \text{give2})$ can give to only one agent at a time

Two consumers:

rc1 (give1) [ask1]

start \Rightarrow ask1

●ask1 \Rightarrow ask1 ask on every cycle

rc2 (ask1, give2) [ask2]

●(ask1 $\wedge \neg$ ask2) \Rightarrow ask2 an ask2 message is sent on every cycle where, on its previous cycle, it did not send ask2 but received ask1.

1. Se ask1 e' appena stato soddisfatto, allora esiste uno stato futuro in cui give1 sara' soddisfatto. Ovvero se rc1 chiede ask1, rp sara' committed a dare give1 in un istante futuro
2. Se ask2 e' appena stato soddisfatto, allora esiste uno stato futuro in cui give2 sara' soddisfatto. Ovvero se rc2 chiede ask2, rp sara' committed a dare give2 in un istante futuro
3. Dall'inizio e' vero che sara' per sempre vero che non sara' vero sia give1 che give2. Ovvero, rp dara' soltanto ad un agente **alla volta**
4. se e' appena accaduto start, allora ask1 e' vero. Ovvero chiedi ask1 all'inizio
5. Se ask1 e' appena stato soddisfatto, allora ask1. Ovvero se ask1 e' stato appena richiesto, allora richiedi nuovamente ask1, ovvero ciclicamente
6. se e' appena accaduto ask1 e -ask2, allora ask2. Ovvero, se rc1 ha chiesto ask1 e non e' stato chiesto ask2 all'istante precedente, allora rc2 chiede ask2 in questo istante

Quali delle seguente affermazioni relative ad AgentSpeak(L) sono vere?

- ☐ AgentSpeak(L) rappresenta achievement goal in un piano ma non test goal.
- ☐ AgentSpeak(L) ha una funzione di selezione per determinare i piani rilevanti.
- ☒ AgentSpeak(L) distingue tra eventi interni ed esterni

- ☐ AgentSpeak(L) ha una funzione di selezione per determinare gli eventi rilevanti
- ☐ AgentSpeak(L) è implementato in linguaggio Java
- ☐ In AgentSpeak(L) sono rappresentati i belief e i desire ma non le intention.
- ☒ AgentSpeak(L) è un linguaggio dichiarativo
- ☐ Agent Speak(L) ha una funzione di selezione per determinare i belief da adottare.

Si consider l'attività di Practical Reasoning, quali delle seguenti affermazioni sono vere?

- ☐ L'attività di means-ends reasoning stabilisce quale stato raggiungere.
- ☐ L'attività di means-ends reasoning produce come risultato le intenzioni dell'agente.
- ☒ L'attività di deliberation stabilisce quale stato raggiungere.
- ☐ L'attività di deliberation stabilisce come raggiungere uno stato.
- ☐ L'attività di means-ends reasoning verifica l'effettiva eseguibilità di un piano per soddisfare le intenzioni.
- ☒ L'attività di deliberation produce come risultato le intenzioni dell'agente.
- ☒ L'attività di means-ends reasoning stabilisce come raggiungere uno stato.
- ☐ L'attività di deliberation delibera la sequenza di azioni che l'agente deve eseguire per raggiungere uno stato.

Due giocatori A e B partecipano a una asta di Vickrey. A e B fanno una offerta maggiore di tutti gli altri partecipanti. A offre 100, mentre B offre 80. Quali delle seguenti affermazioni sono vere:

- ☐ B è il vincitore e paga 100.
- ☐ B è il vincitore e paga 80.
- ☒ A è il vincitore e paga 80.
- ☐ A è il vincitore e paga 100.

- ☒ Una asta di Wickrey è una asta in busta chiusa.

Logica modale: definire l'assioma K e la regola di necessitation.

L'assioma K e la regola di necessitation sono due proprietà fondamentali che valgono su qualunque logica modale.

- Assioma K: $\Box(\alpha \Rightarrow \beta) \Rightarrow (\Box\alpha \Rightarrow \Box\beta)$.
- Necessitation: è una regola di inferenza. Se ϕ è valida allora $\Box\phi$ è valida

Con il termine performativa, nella Speech Act Theory si intende:

- ☐ L'azione risultante dall'atto fisico di chi parla. X
- ☐ L'atto fisico di chi parla.
- ☒ Il significato attribuito da chi parla.

Si consideri il commitment sociale $C(x, y, p, q)$ secondo la definizione data da Munindar Singh. Quali delle seguenti affermazioni sono vere?

- ☐ Se q diventa Falsa, il commitment è violato.
- ☒ y ha l'aspettativa che q diventi vera nel caso che p sia vera.
- ☐ Se p diventa falsa, il commitment è violato.
- ☐ x ha la aspettativa che q diventi vera nel caso che p sia vera. X
- ☒ x è il debitore del commitment.
- ☒ Se p è vera e successivamente q diventa falsa, il commitment è violato.
- ☐ y è il debitore del commitment. X
- ☐ Se p diventa vera, il commitment è soddisfatto.

Si consideri il modello di esecuzione dei comportamenti di un agente in JADE. Dire quali delle seguenti affermazioni sono vere

Scegli una o più alternative:

- ☐ La gestione dei comportamenti di un agente è di tipo competitivo.
- ☐ La gestione dei comportamenti di un agente è di tipo pre-emptive.
- ☒ Quando un comportamento di un agente è selezionato, il suo metodo action viene completamente eseguito fino al suo return.
- ☒ Ogni comportamento di un agente definisce un metodo done.
- ☐ La sincronizzazione tra i comportamenti di un agente è ottenuta mediante le usuali primitive del linguaggio Java.

Si consideri il seguente frammento di codice di un metodo action di un behaviour Jade.

```
public void action() {  
    MessageTemplate pl = MessageTemplate.MatchOntology ("Test-Ontology");  
    ACLMessage msg = myAgent.receive(+pl);  
    if (msg != null) {  
        // Process the message  
    } else {  
        block();  
    }  
    ...  
}
```

Quali delle seguenti affermazioni sono vere?

Scegli una o più alternative:

- ☒ La coda dei messaggi è comune a tutti i comportamenti di uno stesso agente..
- ☒ Se un messaggio non soddisfa il template pl non viene prelevato dalla coda dei messaggi. /
- ☐ Il processo dell'agente risulta in wait all' esecuzione dell'istruzione $block()$ sino alla ricezione del prossimo messaggio. X
- ☐ Alla ricezione di un messaggio l'esecuzione riprende dal punto successivo l'istruzione $block()$.
- ☐ L'esecuzione dell'istruzione $block()$ ferma immediatamente l'esecuzione del comportamento che la contiene.

Si considerino le intenzioni, quali delle seguenti affermazioni sono vere?

Scegli una o più alternative:

- ☐ Le intenzioni sono piani, sequenze di azioni che un agente esegue per raggiungere un suo goal.
- ☒ Le intenzioni sono la sorgente della proattività di un agente.
- ☒ Le intenzioni sono dei vincoli che un agente deve rispettare nella scelta delle azioni da eseguire.
- ☐ Le intenzioni di un agente sono soggette ad un processo di revisione dopo ogni azione da esso eseguita. X
- ☐ Le intenzioni di un agente non sono mai soggette ad un processo di revisione.

Spiegare che cosa si intende per Model checking.

Il model checking è un metodo per verificare proprietà di sistemi concorrenti. L'obiettivo è quello di verificare se una proprietà ϕ è valida in un modello M . Una tecnica utilizzata per il model checking è quella refutazione attraverso la trasformazione dei modelli e delle proprietà in automi di Buchi (simili agli automi a stati finiti, ma accettano le stringhe infinite, ovvero una stringa è accettata se uno stato finale è raggiunto infinite volte nell'esecuzione).

Dato l'automato per $\neg\phi$ e dato l'automato per M , per verificare se ϕ è valida in M si costruisce un automato intersezione tra i due automi per $\neg\phi$ e M . Se l'automato non accetta alcuna stringa infinita, allora $\neg\phi$ non è soddisfacibile in M , ovvero ϕ è valida in M . Nel contesto della logica lineare temporale, è possibile trasformare un qualsiasi modello M in un automa di Buchi prendendo il transition system e prendendo tutti gli stati come finali. Anche le formule LTL possono essere trasformate in automi di Buchi che accettano esattamente le sequenze infinite che sono modelli per le formule.

Scrivere l'Agent Control Loop per l'architettura BDI nella version senza Commitment Strategy (Version 3), spiegando la notazione utilizzata

```
B := B0
I := I0
while true do
  p := next-percept()
  B := brf(B, p)
  D := opts(B, I)
  I := filter(B, D, I)
  pi := plan(B, I)
  execute(pi)
```

Dato un insieme di belief iniziali $B0$ e un insieme di intenzioni $I0$, l'agente esegue per sempre le operazioni di :

- ricevere le percezioni attuali,
- aggiornare i propri belief in funzione delle percezioni appena ricevute
- determinare un insieme di possibili alternative (opts) in base alle proprie credenze e intenzioni
- scegliere le proprie intenzioni concorrenti filtrandole tra alternative che competono
- determina un piano da eseguire impegnandosi a raggiungere le intenzioni filtrate
- eseguire tutto il piano

Questo ciclo rende l'agente overcommitted, in quanto l'agente esegue tutto il piano calcolato, anche se c'e' qualcosa che va storto. Per evitare l'overcommitment si puo' per esempio introdurre un ciclo di ripianificazione ed esecuzione parziale.

Si consideri la formula "true U a" della logica temporale lineare (LTL). Qual è il significato della formula?

Scegli una o più alternative:

- ☐ a sarà sempre vera. X
- ☒ Prima o poi a sarà vera.

Quale o quali delle sequenti affermazioni sono vere?

Scegli una o più alternative:

- ☐ Il practical reasoning è n esempio di ragionamento epistemico.
- ☒ Il theoretical reasoning è il ragionamento che riguarda le credenze (beliefs).
- ☒ Il practical reasoning è il ragionamento che riguarda le azioni (actions).
- ☐ Il practical reasoning è il ragionamento che riguarda le credenze (beliefs).

Qualitesi sostiene Brooke?

Scegli una o più alternative:

- ☒ Un comportamento intelligente può essere generato senza rappresentazione esplicita della conoscenza.
- ☒ Un comportamento intelligente può essere generato senza ragionamento simbolico.
- ☐ Un comportamento intelligente non dipende dall'ambiente.

Quale/quali delle sequent affermazioni sono vere?

Scegli una o più alternative:

- ☒ Le intenzioni sono persistenti.
- ☐ Le intenzioni sono il risultato della fase deliberativa del practical reasoning.
- ☐ Un agente può non credere che le sue intenzioni siano raggiungibili.
- ☒ L'insieme dei desire di un agente può non essere consistente. .
- ☒ Le intenzioni sono il risultato della fase means-ends reasoning del practical reasoning.
- ☐ L'insieme delle intenzioni di un agente può non essere consistente. X
- ☐ Un'insieme di intenzioni di un agente non influenza la sua attività di practical reasoning.

- ☒ Un agente crede che le sue intenzioni siano raggiungibili.

Associare il corretto significato.

- La produzione fisica di enunciati da parte di un oratore: **Locuzione**
- Il risultato, le conseguenze provocate dall'atto comunicativo: **Perlocuzione**
- Il significato inteso da part dell'oratore che vuole rendere noto all'ascoltatore **Illocuzione**

Associare correttamente i significati

- Abilita la coordinazione tra i comportamenti di un insieme di agenti: **comunicazione**
- La proprietà di un sistema di agenti di eseguire attività competitive o self-interested in un ambiente condiviso: **Negoazione**
- La proprietà di un sistema di agenti di eseguire attività non antagonist in un ambiente condiviso: **cooperazione**
- La proprietà di un sistema di agenti di eseguire attività in un ambiente condiviso: **coordinazione**

Dire cosa è vero

Scegli una o più alternative:

- ☐ Un equilibrio di Nash è anche un equilibrio con strategie dominanti. X
- ☒ Un equilibrio con strategie dominanti è anche un equilibrio di Nash.

Dire quali delle seguenti affermazioni relative alle logiche temporali sono vere e quali false.

- $Fa = \text{true} \cup a$ significa che a sarà sempre vera: Falso
- $AG(EFp)$ è una formula CTL: Vero
- $G(p \Rightarrow Xg)$ è una formula LTL che è soddisfatta da tutte le sequenze in cui p è vero infinite volte: Falso

- FGp è una formula CTL: Falso

Si consideri un gioco in cui sia Alice che Bob possono eseguire o l'azione a1 o l'azione a2.

Dire quali delle seguenti affermazioni sono vere.

- Se (Alice:a1, Bob:a1) è un equilibrio di Nash, allora a1 è una strategia dominante sia per Alice che per Bob.: **Falso**
- Se a1 è una strategia dominante sia per Alice che per Bob, allora (Alice:a1, Bob:a1) è un equilibrio di Nash.: **Vero**

Associare le affermazioni con la corretta descrizione.

- Non sarebbe razionale da parte mia adottare l'intenzione è se credessi che © non fosse possibile.: Gli agenti non credono che non porteranno avanti le loro intenzioni
- Se ho l'intenzione è, non ci si aspetta che adotti un'intenzione y tale che o e si escludono a vicenda.: Le intenzioni forniscono un filtro per l'adozione di altre intenzioni, che non devono entrare in conflitto
- Se ho l'intenzione o, ci si aspetta che che dedichi risorse per decidere come portare avanti o.: ???
- Se credo $p = y$ e intendo che ϕ , non intendo necessariamente anche ψ .: Gli agenti non devono necessariamente avere intenzione su tutti gli effetti collaterali delle loro Intenzion
- Normalmente non sarebbe razionale da parte mia

credere che avrei realizzato

le mie intenzioni;

le intenzioni possono fallire

Inoltre, non ha senso che se

credo • sia inevitabile che io

lo adotti come intenzione.: In determinate circostanze, gli agenti credono che realizzeranno le loro intenzioni

Quali delle seguenti affermazioni sono vere e quali false?

- Il problema della rappresentazione/ragionamento di un agent deliberativo, è il problema concernente il tempo impiegato per tradurre il mondo reale, un ambiente, in una descrizione simbolica accurata.: **Falso**
- Il problema sottostante della trasduzione di un agent deliberativo, risiede nella complessità degli algoritmi di manipolazione.: **Vero**
- Le risorse di calcolo disponibili attualmente hanno permesso di superare il problema della trasduzione di n agent deliberativo.: **Falso**
- Il problema della rappresentazione/ragionamento di un agent deliberativo, è il problema concernente il tempo di ragionare su una rappresentazione simbolica del mondo e il modo con cui questo è rappresentato.: **Vero**
- Il problema sottostante della rappresentazione/ragionamento di n agent deliberativo, risiede nella complessità dei algoritmi di manipolazione.: **Vero**

Commentare la formula che definisce l'intenzione di eseguire una azione a secondo Cohen e Levesque:

$\text{INTEND } i \ a \equiv P\text{-GOAL } i \ [\text{DONE } i \ (\text{BEL } i \ (\text{HAPPENS } a))]; a]$

La formula esprime che l'agente i intende eseguire alpha (una azione) se ha un goal persistente di raggiungere lo stato in cui riteneva di essere arrivato per eseguire alpha, quindi segue l'azione alpha. L'idea è quella di far applicare l'azione alpha in modo voluto e non accidentale, assicurandosi che l'agente sia committed per garantire questa cosa.

Un agente Jade può eseguire più comportamenti in parallelo, però il "behaviour scheduling" è non preemprive ma cooperativo e tutto viene eseguito in un singolo Thread.

Vero. Infatti se vi è un behaviour che ha come azione un metodo che va in ciclo infinito, tutte le altre azioni vengono bloccate. Questa scelta è stata presa dai progettisti per facilitare l'integrazione di Jade con dispositivi cellulari con poca memoria e poca potenza di calcolo. Quindi una buona pratica è quella di fornire implementazioni leggere per i metodi `action` per evitare il rallentamento di altre azioni. Nonostante l'utilizzo di un singolo thread, Jade prevede anche alcune astrazioni per rendere la lettura dei messaggi in arrivo "bloccante". Se vi è un behaviour che deve attendere la ricezione di un messaggio, in una situazione single-threaded una prima soluzione potrebbe essere quella di fornire un'azione che verifica ciclicamente se vi è un messaggio in coda a ogni esecuzione. Per evitare questo approccio di busy-waiting, è possibile utilizzare il metodo `block()` fornito nativamente da Jade.

Nella logica epistemica si assume che per l'operatore K (conoscenza) valgano le seguenti proprietà, che possono essere modellate con assiomi:

p1) La conoscenza è non contraddittoria

p2) Ciò che si conosce è vero

p3) Introspezione positiva

p4) Introspezione negativa

- L'assioma $K\phi \Rightarrow \neg K\neg\phi$ modella la proprietà p4: Falso
- L'assioma $K\phi \Rightarrow KK\phi$ modella la proprietà p3: Vero
- L'assioma $\neg K\phi \Rightarrow K(\neg K\phi)$ modella la proprietà p1: Falso
- L'assioma $K\phi \Rightarrow \phi$ modella la proprietà p2: Vero

Dire se le seguenti affermazioni sono vere o false.

- L'intenzione è lo stato di cose che un agente ha scelto e al quale si impegna a raggiungere.: Vero
- Le intenzioni non vincolano il futuro practical reasoning.: Falso
- Se un agent fallisce un primo tentativo per ottenere l'intenzione a, esso tenterà di eseguire un piano alternativo per raggiungere la propria intenzione.: Vero
- Gli agenti non credono che le proprie intenzioni siano possibili fino a che non le raggiungono.: Falso
- Le intenzioni son chiuse rispetto l'implicazione.: Falso

Due giocatori A e B partecipano a una asta di Wickrey. A e B fanno una offerta maggiore di tutti gli altri partecipanti. A offre 100, mentre B offre 80. Quali delle seguenti affermazioni sono vere e quali false?

- A è il vincitore e paga 100: Falso
- B è il vincitore e paga 80: Falso
- B è il vincitore e paga 100: Falso
- A è il vincitore e paga 80: Vero
- Una asta di Wickrey è una asta in busta chiusa: Vero

Dire se le seguenti affermazioni sono vere o false.

- Se un agente ha l'intenzione a, ci aspettiamo che l'agente adotti una intenzione b tale che a e b siano mutuamente esclusive. Falso
- Gli agenti credono che le proprie intenzioni siano possibili. Vero

- Le intenzioni non sono chiuse rispetto l'implicazione Vero
- Le intenzioni sono persistenti Vero
- L'intenzione e' lo stato di cose che un agente desidera Vero

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

- ☒ 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.
- ☐ Un comportamento ha tanti metodi action() quanti sono i task indipendenti che tale comportamento deve raggiungere.
- ☐ Il metodo action() è eseguito parzialmente sino ad una condizione di sincronizzazione con altro comportamento presente pool dei comportamenti attivi dell'agente.
- ☐ Un comportamento ciclico non valuta mai il risultato dell' esecuzione del metodo done). x
- ☐ Un comportamento risiede nel pool dei comportamenti attivi fino all' esecuzione del metodo takeDown(). X
- ☒ Il valore restituito dal metodo done() determina se un comportamento deve o no essere inserito nel pool dei comportamenti attivi dell'agente.