

---

---

---

---

---



# PRIMO PARZIALE

2.In "Speech Act Theory", the component of a communicative act corresponding to the action that the speaker expects the receiver to perform in consequence of the communication is called:

- a) locution;
- b) illocution;**
- c) perlocution.

LOCATION  
ILLOCUTION  
PERLOCUTION

contesto del msg  
intenzioni comunicate nel msg  
conseguenze del msg.

3.Elencare le componenti di un atto comunicativo nella "Speech Act Theory":

3) Nella "Speech Act Theory" una "locution" è:

- a) ciò che viene detto;**
- b) l'intenzione con cui qualcosa viene detto;
- c) l'effetto che si vuole ottenere dicendo qualcosa.

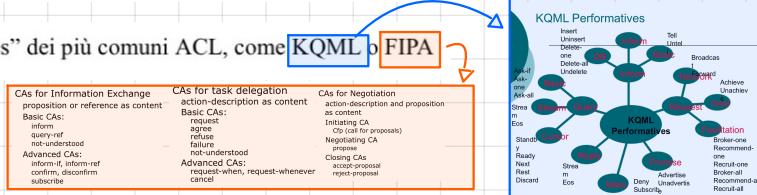
3.In any ACL, each message includes a primitive, often called *performative* (e.g., *inform*). It corresponds to the following component of a communicative act in "Speech Act Theory":

- a) locution;
- b) illocution;**
- c) perlocution.

PERFORMATIVE IN ACL (dalla 9^ lezione comunicazione)

Richiesta, volere, informare)

4) Elencare almeno due "performatives" dei più comuni ACL, come



4.In BDI agents,

- a. Beliefs represent: .....
- b. Desires represent: .....
- c. Intentions represent: .....

BELIEFS caratteristiche dell'ambiente o aggiornamenti dati

Percezioni sensoriali (componente INFORMATIVO del BDI)

DESIRESI mpo sui GOAL, Pieghe e Priorità associate (componente MOTIVAZIONALE del BDI)

INTENTIONS rappresenta l'azione in corso, cioè l'ultimo output della SELECTION FUNCTION (componente DECIBERATIVO del BDI)

6) Nei "practical reasoning agents", dire quale forma di commitment è propria di un agente il quale, se nel corso del suo processo di ragionamento arriva a ritenere che uno degli obiettivi correnti non sia più raggiungibile, rinuncia prontamente all'obiettivo senza aspettare di "scontrarsi" in pratica con un ostacolo invalicabile.

- a) blind commitment;
- b) single-minded commitment;**
- c) open-minded commitment

SINGLE MINDED  
OPEN MINDED

Scatta se raggiungibile  
Scatta, ma può riprendersela

1.Nei "practical reasoning agents", il componente di "option generation":

- a) considera i vari "desires" correnti e seleziona le opzioni da trasformare in "intention";
- b) considera "belief" ed "intentions" correnti genera nuovi "desires";**
- c) genera il piano più efficiente fra le opzioni precedentemente selezionate.

2.Nei "practical reasoning agents", il componente di "filtering":

- a) considera i vari "desires" correnti e seleziona quello da trasformare in "intention";
- b) considera "beliefs", "desires" ed "intentions" correnti e seleziona un "desire" da trasformare in una "intention" da attuare;**
- c) genera il piano più efficiente per l'"intention" selezionata al passo precedente.

• La funzione *deliberate* può essere decomposta in due componenti funzionali distinte:

*generazione di opzioni*

in cui l'agente genera un insieme di possibili alternative; Rappresenta la generazione di opzioni attraverso una funzione, *options*, che prende le credenze correnti dell'agente e le intenzioni correnti e da loro determina un insieme di opzioni (= *desideri*)

*filtraggio*

in cui l'agente sceglie tra le alternative in concorrenza e si impegna a trovare il modo di raggiungerle. In modo da scegliere tra le opzioni in concorrenza, un agente usa una funzione *filter*.

5.The AgentSpeak language implements the BDI approach:

- a. fully;
- b.** with some limitations.

7.A Si parli in breve del linguaggio AgentSpeak, spiegando in che senso implementa il BDI e con quali limitazioni. Si scriva una regola esemplificativa.

1.Tell which of the functions below is **not** proper of a purely reactive agent:

- a) perform actions in response to external events;
- b.** select which action to perform via a reasoning process;
- c) establish the most urgent event via priorities.

5.La regola di inferenza che afferma come falsa ogni proposizione non dimostrabile si chiama:

- a. Open World Assumption (OWA);
- b.** Closed World Assumption (CWA);

6.La Closed World Assumption è usata:

- a) nelle Ontologie;
- b.** in prolog e Datalog;
- c) in Description Logic

6) Il linguaggio AgentSpeak è una realizzazione di:

- a) agenti puramente reattivi
- b.** agenti BDI
- c) agenti KGP.

2) L'approccio BDI implementa:

- a) deductive reasoning agents;
- b) purely reactive agents;
- c.** practical reasoning agents.

4) La relazione di inibizione in nella subsumption architecture:

- a) Impedisce all'agente di eseguire azioni non volute in un certo stato;
- b.** Definisce priorità fra comportamenti applicabili;
- c) Impedisce all'agente di comunicare con agenti estranei alla sua comunità.

2) Dire quale/quali delle seguenti proprietà NON sono caratteristiche della subsumption architecture:

- a) priorità fra regole
- b) stigmergy
- c.** ACL

1) Nella subsumption Architecture, gli agenti comunicano mediante:

- a. scambio di messaggi;
- b.** "Stigmergy", o "tracce radioattive";
- c. blackboard centralizzata di messaggi.

3) Indicare quali delle seguenti caratteristiche sono proprie della "subsumption architecture".

- a.** ragionamento simbolico;
- b.** regole reattive con relazione di inibizione;
- c) comunicazione ad alto livello;
- d) tracce percepite dagli altri agenti.

## 5) Considerando la regola AgentSpeak

+film(F) : likes(F) **B** !go\_to\_see(F). dire (con parole vostre)

a) cos'è +film (F)..... TRUCCHERIA EVENT → Add di un BECIFE

b) cos'è !go\_to\_see(F)..... ACQUIMENT GOAL → L'Agente vuole raggiungere un mondo in cui il sottogoal sia valido

## 1) Considerando la regola AgentSpeak:

+concerto (Titolo,Data) : mi\_piace(Titolo) <- !compro\_biglietto(Titolo,Data).

i tre atomi rappresentano (completare):

(a) +concerto (Titolo, Data) : TRUCCHERIA EVENT → Add di un BECIFE

(b) mi\_piace(Titolo) : Contesto

(c) !comprobilletto (Titolo, Data) : ACQUIMENT GOAL → L'Agente vuole raggiungere un mondo in cui il sottogoal sia valido

7.A Shortly discuss the differences between “practical reasoning agents” (of which the BDI agents are an important representative) and purely reactive architectures, such as the “subsumption architecture” and “behavior networks”.

8.A Describe AgentSpeak: discuss its main features also by means of a sample rule.

9.E Consider the following ASP program. Compute its answer sets and explain the procedure that you have followed. Remember that the constraint has to be translated into pure ASP as an even cycle with a handle.

```
flies(X):- obj(X), not noflies(X).  
noflies(X):- obj(X), not flies(X).
```

```
:- obj(X),flies(X), not bird(X).
```

```
bird(a).  
dog(b).  
obj(a).  
obj(b).
```

10.E. Write an Answer Set Program describing the following situation. You have various kinds of food, among which (at least) pasta, meat, fish and cake. You also have red wines and white wines. A meal is composed of at least two kinds of food, but must include pasta (it cannot avoid including it) and cannot include simultaneously meat and fish (only one can be present). Optional: each meal includes exactly one wine, that cannot be red if you have fish. Hint: exploit cardinality constraints.

8.A Parlare del modello KGP.

9.E Si scriva un predicato Prolog `split(L1, N, L2, L3)` che dati in ingresso una lista `L1` di interi ed un numero intero `N`, restituisca in uscita due liste, `L2` ed `L3`, la prima contenente gli elementi di `L1` maggiori di `N` e la seconda quelli minori od uguali.

**Esempio**

```
?-split([3,1,7,5],4,L1,L2)
```

dà come risultato:

```
L1=[7,5]
```

```
L2=[3,1]
```

10.E. Si scriva un semplice programma AgentSpeak riguardante un robot che si aggira in un ambiente. Se il robot trova una roccia verde avverte la stazione base. Altrimenti, se la via è libera si muove in avanti, e se è occupata gira (non importa se a destra o a sinistra). Tutte le azioni sono possibili solo se la batteria non è scarica. Per semplicità non occorre trattare il caso in cui invece la batteria sia scarica.

+roccia(V) : verde(V), not batteria\_scarica <- !avverti\_stazione.

+via\_libera : not batteria\_scarica <- avanti.

+via\_ostruita : not batteria\_scarica <- scegli\_direzione(D), gira(D).

DALI:

RoccaVE(v) :- omec (check(v))

check(v) :- verde(v), not BATT-SCA, AVVERTI-STAZ.A  
VIA LIBERA B (v) :- AVANTIA

AVANTI :- not BATT-SCA

VIA OSTRUITA B (v) :- GIRA A

GIRA :- not BATT-SCAR

8) E. Si assuma di trovarsi in inverno. Si definisca un agente DALI che rileva in continuazione la temperatura esterna, e in risposta accende o spegne il riscaldamento. Quando l'agente ha voglia di uscire (in base ad una sua condizione interna), lo fa solo se la temperatura è sopra lo 0.

9) A Si discuta in breve cos'è un'ontologia, perchè è importante per gli agenti e come può/deve essere usata.

10) A Spiegare cosa si intende per "committment" negli agenti, quali forme di committment possono esservi e quale si ritiene preferibile.

7) E Definire in DALI un programma equivalente al programma AgentSpeak della domanda 5, dove in aggiunta la notizia di un nuovo film da vedere viene accettata solo se proveniente da un amico affidabile (consiglio: usare una regola *told*).

5) Considerando la regola AgentSpeak

+film(F) : likes(F)  $\exists$  !go\_to\_see(F). dire (con parole vostre)

a) cos'è +film (F).....

b) cos'è !go\_to\_see(F).....

*told( A , inform(film(F),A) ) :- friend(A), Trusted(A)*  
*inform(film(F),A) :- once(chek(F))*  
*chek(F) :- like(F), goto seeA(F)*

9) E Definire due agenti DALI A e B dove: (i) A si domanda se ha la somma di denaro D per il biglietto del cinema; in caso affermativo, invita l'altro agente B ad andare insieme. (ii) quando B riceve l'invito invia la risposta, che è positiva se l'agente A gli è simpatico e ritiene che il film sia interessante. Per semplicità, si ignori lo standard FIPA ma si assuma che i messaggi abbiano la forma seguente: inviteA(X,Y,D) per l'azione dove X invita Y a fare l'attività D, acceptA(X,Y,D) per l'azione dove X accetta la proposta di Y e refuseA(X,Y,D) per l'azione dove invece X non accetta.

E. Write a simple AgentSpeak or DALI program which manages a library; i.e., it receives requests to borrow a book B. If the requester is trustworthy, then: (i) if (s)he has presently no other book B1 then the book is immediately given, otherwise (ii) the requester is asked to return the book that (s)he has. Requests can simply be formulated (even within *told* rules) simply as *request(A,B)* where A is an agent and B a book. Give for granted that the knowledge base includes beliefs about trustworthy agents, and about loans (which agent has borrowed which book).  
*Optional:* when book B1 is returned, then the book B is given to the requester.

*TOLD (REQUESTER, REQUEST (REQUESTER, B)) :- TRUST (REQUESTER)*  
*REQUEST (REQUESTER, B) :- once (check)*  
*check :- Borrowed (REQUESTER, Book), AskToReturn (REQUESTER, Book)*  
*check :- NotBorrowed (REQUESTER, B), loanA (B)*

*RETURNED (REQUESTER, B) :- once (check1)*

*check1 :- requestP (REQUESTER, B), loanA (B)*

## ROBOT MANUFACTURE

AT\_SHOP\_E(BAKED, X) :- once (check\_Location(X))

check\_Location(X) :- Location(Robot, X), Location(House, Y), Buy(Cake)

Buy(Cake) :- Location(Robot, Y), BayP(cake)

## TEMPERATURE

Temperature(E(+)) :- once (check(+))

check(+):- high(+), sogniA

check(+):- low(+), accendiA

high(+):-  $T \geq \text{threshold}$

low(+):-  $T < \text{threshold}$

WANT\_TO\_GO\_OUT :- bored.

WANT\_TO\_GO\_OUT T :- Temperature\_outP(T),  $T > 0$ , go\_outA