

Soluzioni alla Seconda provetta di Sistemi Operativi per la Robotica - 28/5/2009

1 (1pt) Quali di queste affermazioni sono vere?

RTAI risiede staticamente nel kernel di linux

I task di RTAI non condividono lo spazio di indirizzamento con altri task

La schedulazione in RTAI non può essere aperiodica

RTAI offre la schedulazione EDF

V F
V F
V F
V F

2 (4pt) Nel modello robotico 'uniciclo ideale' lo Spazio delle Configurazioni è rappresentato dal vettore $\mathbf{q}=[x \ y \ \phi]^T$ dove l'apice T significa vettore trasposto. Se $[s \ j]$ sono la velocità lineare e angolare del robot (speed, jog), e $d\mathbf{q}/dt= G(\mathbf{q}) [s \ j]^T$, scrivere la forma della matrice $G(\mathbf{q})$ di 3 righe e 2 colonne.

Soluzione

Il modello cinematico e' espresso dalla seguente equazione, da cui si evince la forma della matrice $G(\mathbf{q})$:

$$\frac{d \mathbf{q}}{d t} = \frac{d}{d t} \begin{vmatrix} x \\ y \\ \theta \end{vmatrix} = \begin{vmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{vmatrix} \begin{vmatrix} s \\ j \end{vmatrix}$$

3 (2pt) Descrivere brevemente le caratteristiche dei modelli cognitivi reattivi, deliberativi, subsumption.

Soluzione

I modelli cognitivi reattivi sono relativi ad un funzionamento del robot in cui le percezioni agiscono sugli attuatori secondo una reazione immediata. Chiamati anche 'modelli sense-act' (non pensare, reagisci), sono modelli senza memoria, senza conoscenza del mondo circostante, senza apprendimento. Strutturati secondo una deconposizione orrizzontale.

I modelli cognitivi deliberativi sono relativi ad un funzionamento del robot in cui le percezioni agiscono sugli attuatori secondo una azione che segue un raginamento. Chiamati anche 'modelli sense-plan-act' (pensa e poi reagisci), possono avere memoria, conoscenza del mondo circostante, apprendimento. Strutturati secondo una deconposizione orrizzontale.

I modelli subsumption sono relativi ad un funzionamento del robot in cui le percezioni agiscono sugli attuatori secondo una reazione immediata. Rappresentano un modo di implementare i modelli reattivi, nel quale un modello inibisce i modelli sottostanti a seconda delle percezioni. Strutturati secondo una deconposizione verticale.

4 (5pt) Questo è il main del programma randwalk.cc visto nella seconda parte del corso.

```
int main(int argc, char** argv)
{
    int randint;    int randcount = 0;    int avoidcount = 0;    bool obs = false;
    parse_args(argc,argv);
    LaserProxy *lp = NULL;    SonarProxy *sp = NULL;

    PlayerClient robot(gHostname, gPort); Position2dProxy pp(&robot, gIndex);
    sp = new SonarProxy (&robot, gIndex); pp.SetMotorEnable (true);
    double newturnrate=0.0f, newspeed=0.0f;
    for(;;)
    {
        robot.Read();
        obs = ((sp->GetScan (2) < minfrontdistance) ||
              (sp->GetScan (3) < minfrontdistance) ||
```

```

        (sp->GetScan (4) < minfrontdistance) ||
        (sp->GetScan (5) < minfrontdistance) );
if(obs) {
    newspeed=avoidspeed;
    if(sp->GetScan(1)+sp->GetScan(15)<sp->GetScan(7)+sp->GetScan(8))
        newturnrate = -turnrate; else newturnrate = turnrate;
} else
{
    avoidcount = 0;
    newspeed = speed;
    if(!randcount) //rotazione random ogni 2 secondi
    {
        randint = rand() % 41 - 20;//numero random tra -20 e 20
        newturnrate = dtor(randint);
        randcount = 20;
    }
    randcount--;
}
pp.SetSpeed(newspeed, newturnrate);
}
}

```

Tenendo conto che i sensori [2][3][4][5] sono frontali, [10][11][12][13] sono posteriori, [1][15] sono a sinistra e [7][8] a destra del robot, e un angolo di jog negativo sterza a destra e positivo sterza a sinistra, rispondere alle seguenti domande. Come si può modificare il programma per avere i seguenti comportamenti?

1. *avoidspeed* è settato a velocità negativa (indietreggia quando c'è un ostacolo frontale). In questo caso il robot indietreggia per 0.1 secondi (si noti che ogni ciclo `for(;;)` avviene ogni 100 ms) e poi sterza per 100 ms di un angolo casuale.
2. mentre indietreggia il robot si accorge di ostacoli posteriori ad esso e, nel caso, savanza per 0.1 secondi e poi sterza per 100 ms di un angolo casuale
3. la rotazione random ogni due secondi viene sostituita da una velocità lineare costante

Soluzione

Bisogna tener presente che `SetSpeed` serve solo a controllare gli attuatori; non implica ritardi. Il ritardo di 100 ms è introdotto dal metodo `robot.Read()`. Detto questo, un modo per esaudire alle richieste è descritto dal seguente codice, dove il controllo sugli ostacoli posteriori è realizzato dopo la sterzata casuale:

```

int main(int argc, char** argv)
{
    int randint;    int randcount = 0;    int avoidcount = 0;
    bool obs = false;    parse_args(argc,argv);
    LaserProxy *lp = NULL;    SonarProxy *sp = NULL;

    PlayerClient robot(gHostname, gPort);
    Position2dProxy pp(&robot, gIndex); sp = new SonarProxy (&robot, gIndex);
    pp.SetMotorEnable (true);
    double newturnrate=0.0f, newspeed=0.0f;
    for(;;)
    {
        robot.Read();
        obs = (
            (sp->GetScan (2) < minfrontdistance) ||
            (sp->GetScan (3) < minfrontdistance) ||
            (sp->GetScan (4) < minfrontdistance)
            || (sp->GetScan (5) < minfrontdistance)
        );
        if(obs) //se ostacolo frontale

```

```

{
  pp.SetSpeed(-4,0);
  robot.Read();//indietreggia 100ms
  randint = rand() % 81 - 40;
  newturnrate = dtor(randint);
  pp.SetSpeed(0,newturnrate);
  robot.Read();//ruota random 100ms
  obs = (
    (sp->GetScan (10) < minpostdistance) ||
    (sp->GetScan (11) < minpostdistance) ||
    (sp->GetScan (12) < minpostdistance)
    || (sp->GetScan (13) < minpostdistance)
  );
  if(obs ) //se ostacolo posteriore
  {
    pp.SetSpeed(4,0);
    robot.Read();//avanza 100ms
    randint = rand() % 81 - 40;
    newturnrate = dtor(randint);
    pp.SetSpeed(0,newturnrate);
    robot.Read();
  }
  else pp.SetSpeed(1,0);//avanzamento lineare
}
}

```

5 (4pt) Siano A e B due insiemi fuzzy e C, D, F tre insiemi fuzzy risultanti dai seguenti operatori: $C=A \cup B$, $D=A \cap B$, $F=\overline{A \cup B}$ dove con la barretta superiore rappresenta il complemento. Rispondere alle seguenti due domande:

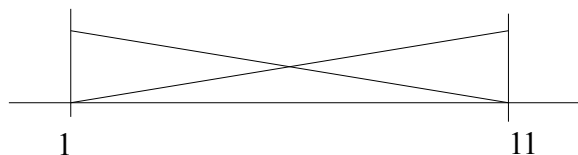
a. Sia dato un universo di discorso discreto $X=\{1, 3, 5, 7, 9, 11\}$ e siano A e B due insiemi fuzzy definiti su X così fatti:

$$A=\{0.1/1, 0.2/3, 0.3/5, 0.4/7, 0.5/9, 0.6/11\}$$

$$B=\{0.9/1, 0.8/3, 0.7/5, 0.6/7, 0.5/9, 0.4/11\}$$

Quanto valgono C, D, F? Verificare che $\overline{A \cup B} = \overline{A} \cap \overline{B}$ (DeMorgan per gli insiemi fuzzy su universo discreto).

b.Sia dato un universo di discorso continuo $X=[1...11]$ e siano A e B due insiemi fuzzy definiti su X così fatti:



Rappresentare i tre insiemi C, D, F. Verificare che $\overline{A \cup B} = \overline{A} \cap \overline{B}$ (DeMorgan per gli insiemi fuzzy su universo continuo).

Soluzione

Per l'universo discreto:

$$A=\{0.1/1, 0.2/3, 0.3/5, 0.4/7, 0.5/9, 0.6/11\}$$

$$B=\{0.9/1, 0.8/3, 0.7/5, 0.6/7, 0.5/9, 0.4/11\}$$

quindi:

$$\overline{A}=\{0.9/1, 0.8/3, 0.7/5, 0.6/7, 0.5/9, 0.4/11\}$$

$$\overline{B}=\{0.1/1, 0.2/3, 0.3/5, 0.4/7, 0.5/9, 0.6/11\}$$

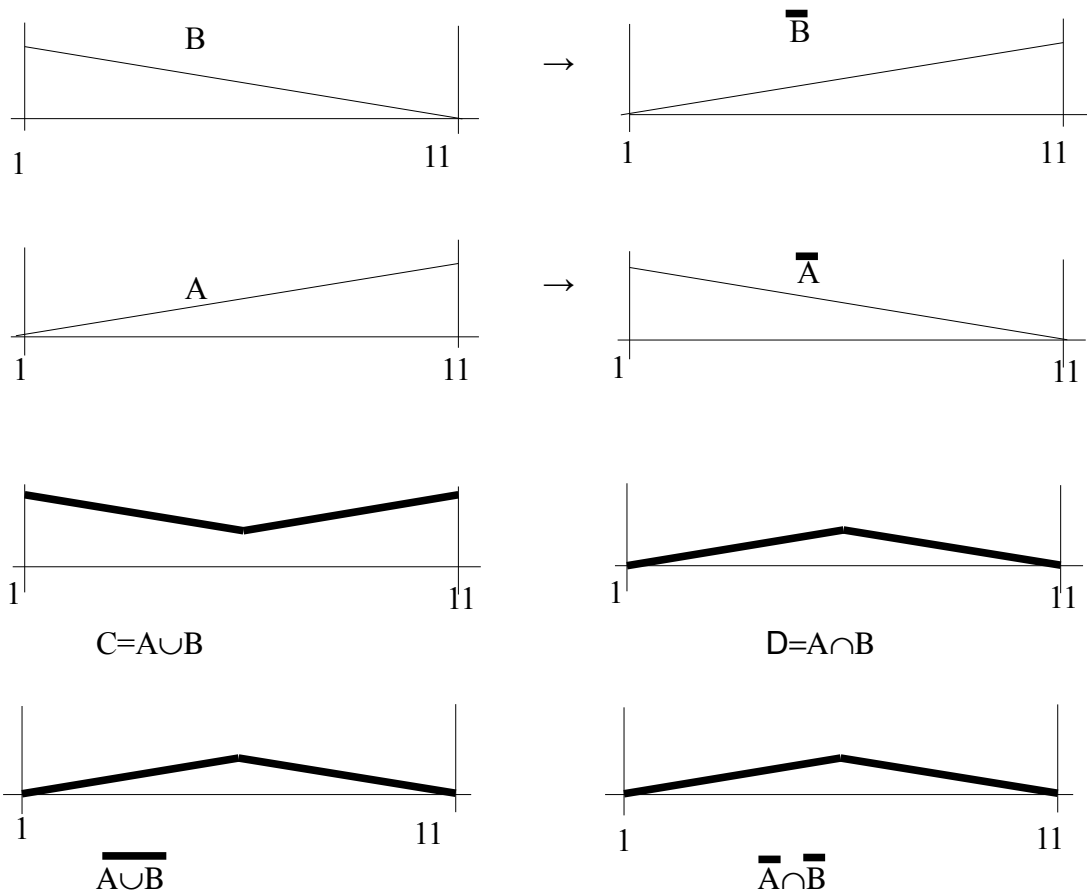
$$C=A \cup B=\{0.9/1, 0.8/3, 0.7/5, 0.6/7, 0.5/9, 0.6/11\}$$

$$D=A\cap B=\{0.1/1, 0.2/3, 0.3/5, 0.4/7, 0.5/9, 0.4/11\}$$

$$F=\overline{A\cup B}=\{0.1/1, 0.2/3, 0.3/5, 0.4/7, 0.5/9, 0.4/11\}$$

$$\overline{A\cap B}=\{0.1/1, 0.2/3, 0.3/5, 0.4/7, 0.5/9, 0.4/11\}=\overline{A\cup B}$$

Per l'universo continuo:



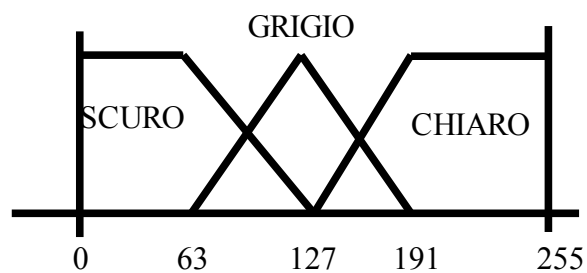
6 (4pt) Un semplice sistema fuzzy per regolare il contrasto di una immagine è il seguente

R1: IF (z is SCURO) THEN (v is PIUSCURO)

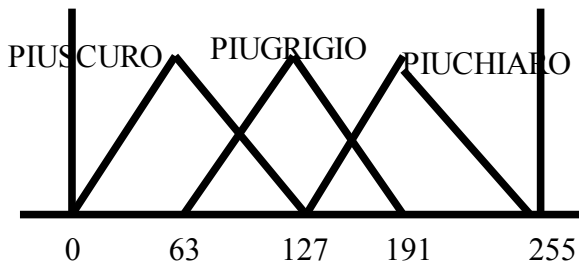
R2: IF (z is GRIGIO) THEN (v is GRIGIO)

R3: IF (z is CHIARO) THEN (v is PIUCHIARO)

dove z e v sono le intensità di un pixel di ingresso e di uscita rispettivamente. Gli insiemi fuzzy di ingresso sono i seguenti:



e quelli di uscita sono:



Calcolare l'intensità del pixel d'uscita se il pixel di ingresso è pari a 50, 95 e 159.

Soluzione

Intanto vediamo che i punti di intersezione tra le pendenze diagonali sono 95 e 159.

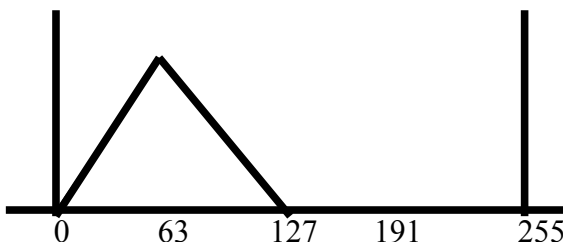
Se $z=50$ il grado di appartenenza all'insieme SCURO è 1, a GRIGIO è 0 e a CHIARO è 0

se $z=95$ il grado di appartenenza all'insieme SCURO è 0.5, a GRIGIO è 0.5 e a CHIARO è 0

se $z=159$ il grado di appartenenza all'insieme SCURO è 0, a GRIGIO è 0.5 e a CHIARO è 0.5

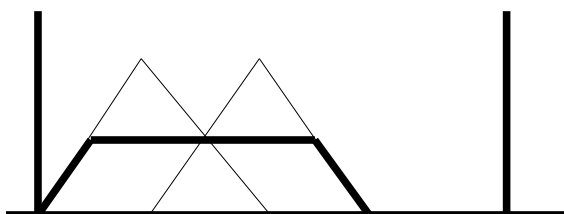
Applichiamo Mandami.

Se $z=50$, v appartiene all'insieme fuzzy



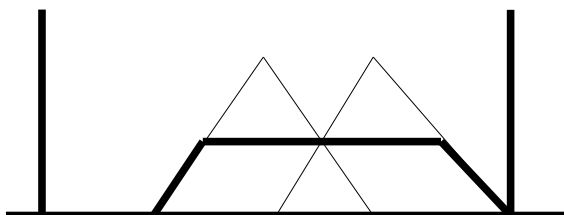
che ha baricentro pari a 63.

Se $z=95$, v appartiene all'insieme fuzzy



che ha baricentro pari a 95

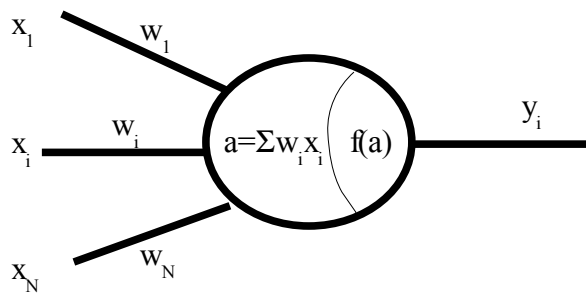
Infine, se $z=159$, v appartiene all'insieme fuzzy



con baricentro pari a 159

Si noti che i baricentri si possono ricavare semplicemente dalla simmetria degli insiemi.

7 (4pt) Una rete neurale usa neuroni artificiali con N ingressi:



La Funzione di attivazione sia $f(a) = \frac{2}{3} a^{\frac{1}{3}}$.

Derivare le equazioni di addestramento del neurone artificiale (aggiornamento dei pesi) usando il metodo del gradiente e definendo la funzione d'errore nel seguente modo: $E = \frac{1}{2} (d-y)^2$ dove d è la funzione desiderata e y l'uscita.

Si ricorda che il metodo del gradiente è un algoritmo iterativo: $w^{k+1} = w^k - \mu \partial E / \partial w^k$

Soluzione

La derivata parziale dell'errore E , $E = \frac{(d-y)^2}{2}$, rispetto al peso generico w_i è data dalla:

$$\frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial y} \frac{\partial y}{\partial w_i} = -(d-y) \frac{\partial y}{\partial w_i} = -\delta \frac{\partial y}{\partial w_i}$$

dove $y = \frac{2}{3} a^{\frac{1}{3}}$ e $a = \sum w_i x_i$. Quindi $\frac{\partial y}{\partial w_i} = \frac{\partial y}{\partial a} \frac{\partial a}{\partial w_i} = a^{-\frac{2}{3}} x_i = \left(\frac{3}{2} y\right)^{\frac{1}{3}} x_i$.

In definitiva $\frac{\partial E}{\partial w_i} = -\delta \frac{\partial y}{\partial w_i} = -\delta \left(\frac{3}{2} y\right)^{\frac{1}{3}} x_i$ e quindi $w_i^{k+1} = w_i^k + \mu \delta \left(\frac{3}{2} y\right)^{\frac{1}{3}} x_i$

8 (4pt) Si ricorda che il robot cartesiano CART ha dei sensori, in particolare P, che percepisce che nella casella diacente c'è una persona. Con P! Percepisce che nella casella stessa c'è una persona. CART, ad un certo punto della sua esplorazione, accumula le seguenti conoscenze, espresse sotto forma di regole R1 e R2.:

R1: $\sim P_{2,2} \rightarrow \sim P_{2,1} \wedge \sim P_{!2,2}$

R2: $P_{2,1}$

Si dimostri che da queste regole CART conclude che $P_{2,2}$ è vera, cioè che c'è una persona nelle celle adiacenti a (2,2).

Nota: si ricordano alcune proprietà della logica proposizionale

- contrapposizione:
 $A \rightarrow B$ equivale a $\sim B \rightarrow \sim A$
- la proprietà distributiva vale anche tra implicazioni e connettori logici:
 $A \rightarrow (B \wedge C)$ equivale a $(A \rightarrow B) \wedge (A \rightarrow C)$

Soluzione

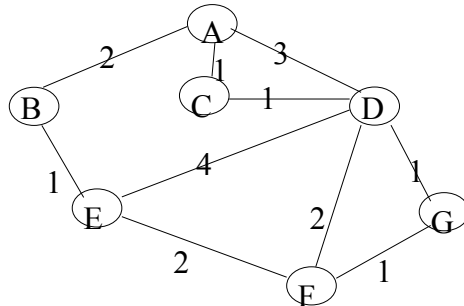
Un modo semplice per dimostrare la tesi è il seguente:

Proprietà distributiva dalla R1: $(\sim P_{2,2} \rightarrow \sim P_{2,1}) \wedge (\sim P_{2,2} \rightarrow \sim P_{!2,2})$

Disgiunzione (devono essere vere entrambe): ($\sim P_{2,2} \rightarrow \sim P_{2,1}$) e per contrapposizione:

($P_{2,1} \rightarrow P_{2,2}$). Dalla R2 si deduce $P_{2,2}$ cioè c'è una persona in una cella adiacente a (2,2).

9 (4pt) Sia dato il seguente grafo di visibilità che rappresenta i possibili percorsi tra il robot posto in A e la posizione finale posta in F. Sugli archi è visualizzato il costo del percorso cioè la distanza tra nodi.



Determinare con l'algoritmo di Dijkstra il cammino ottimo tra il nodo A e il nodo F, elencandone i passi.

Soluzione

Parto dal nodo A. Etichetto i nodi adiacenti: B con (A,2), C con (A,1) e D con (A,3).

Di questi scelgo C e etichetto l'adiacente D con (C,2) che sovrascrive la precedente etichetta perché minore di (A,3).

Tra B e D scelgo ad esempio B e da questo etichetto E con (B,3). Tra E e D scelgo D e poi F con (D,4) e G con (D,3). A questo punto tutti i nodi sono etichettati. Il percorso minimo tra A e F è quindi A - C - D - F con peso 4 e tra A e D è A - C - D con peso 2.

Se avessi scelto D, al passo precedente, avrei etichettato E con (D,6), F con (D,4) e G con (D,3). Dei nodi etichettato scelgo B con (A,2) e da questo rietichetto E con (B,3). Tra E e G scelgo ad esempio E ma vedo che non posso più rietichettare nodi. Anche da G vedo che non posso più rietichettare.

Alla fine le etichette sono: B con (A,2), C con (A,1), D con (C,2), E con (B,3), F con (D,4) e G con (D,3). Da queste vedo il percorso minimo ad A da tutti i nodi.