

## Tentamen i Robotteknik MPR211, 12 mars 1999

Lärare: Rolf Berlin, ank 1286  
Per-Åke Jansson ank 1527

Tillåtna hjälpmedel: Typgodkända kalkylatorer och alla formelsamlingar.  
OBS bifogat finns några formler på sista sidan!

Betygslista anslås onsdag 24/3 på Robotlaboratoriets anslagstavla. Granskning sker hos Gunvor Johansson torsdag 25/3 kl 13-15 på inst. för prod teknik.

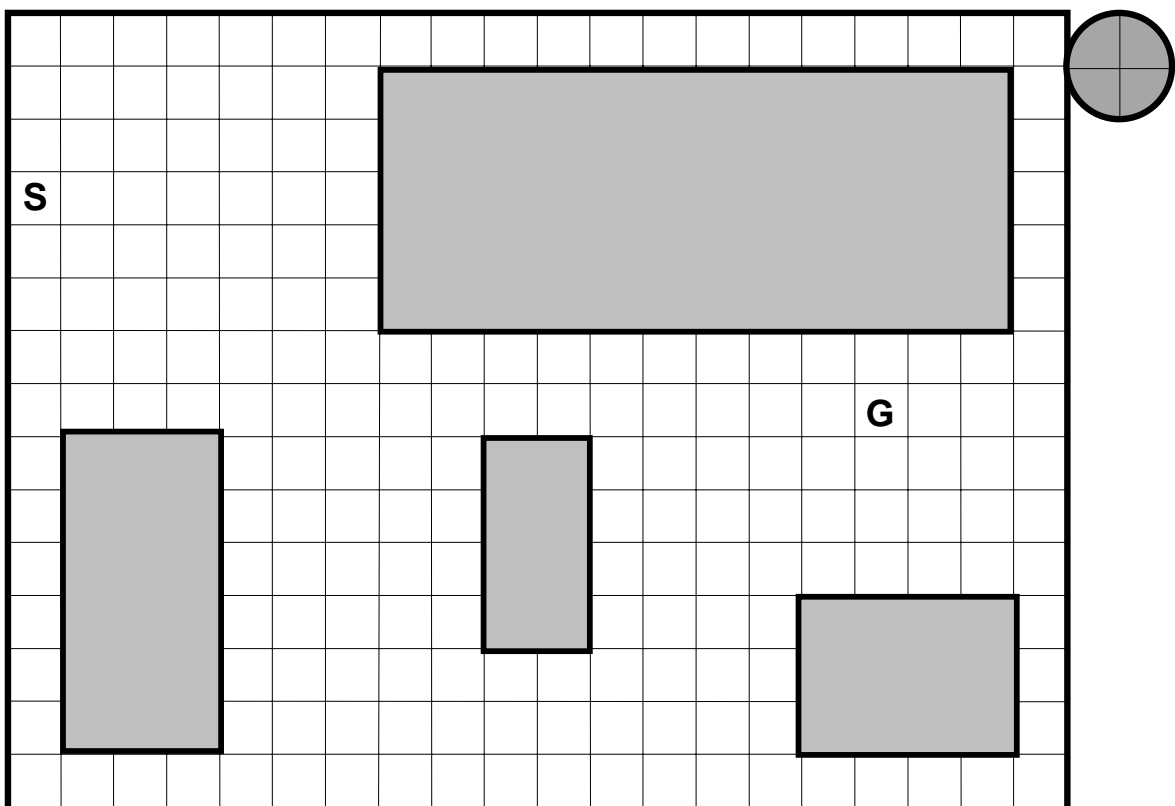
**Betygsgränser: 30-39p=betyg 3, 40-49p =betyg 4 50-60p=betyg 5**

- |    |  |    |
|----|--|----|
| 1a | Beskriv det karakteristiska för en telemanipulator och en industrirobot  | 2p |
| 1b | Vilka två brancher har varit mest drivande för att använda och utveckla robotteknik. Beskriv även orsaken till respektive branches behov.                              | 2p |
| 1c | Vilka två industrirobotapplikationer svarar för hälften av alla installerade robotar, motivera även varför de blivit så dominerande.                                   | 2p |
| 1d | Beskriv tre koordinatsystem som man använder vid programmering av industrirobotar. Motivera vid vilka tillfällen de olika koordinatsystemen är lämpligast att använda. | 3p |
| 1e | Vad betyder förkortningen tcp och varför används denna funktion.   | 1p |
| 2a | Förklara begreppen: programmeringsfel, absolutfel och repeterfel i samband med robotprogrammering, rita gärna en skiss.  | 3p |
| 2b | Nämn sex centrala krav på ett avancerat off-line programmerings system   | 3p |
| 2c | Beskriv hur en "harmonic drive" är uppbyggd, vad som är ingående och utgående axel samt hur den fungerar.  | 4p |



- 3a Förklara vad som menas med orden "**reactive multiagent robotic societies**" citatet är klippt ur en forskningsrapport om mobila robotar. 2p
- 3b I vilka tre huvudfunktioner delar man upp **NAVIGERING** av avancerade mobila robotar. Beskriv vad som ingår i de olika nivåerna 3p
- 3c **Expandera objekten** med hänsyn till den runda roboten med radie en ruta. Förklara varför man expanderar objekten.
- Utför därefter en **komplett banplanering** från start **S** till målet **G**. Använd en "8-connected distance transform", och "8-connected path finding algorithm" baserad på verkligt avstånd mellan rutacentrum. Rutorna är 10 enheter i kvadrat. 5p

**OBS det finns en kopia av figuren för inlämning längst bak!**



- 4.a Nämn 3 st personsäkerhetsfunktioner i ett ABB industrirobotsystem (S4, IRb 1400) med anknytning till on-line programmering? (1p)
- 4.b Antag att Ni programmerar en ABB IRb 1400/S4. Beskriv detaljerat vad följande RAPID-instruktion resulterar i för robotrörelse och ange även andra möjliga argument samt hur dessa inverkar?

**MoveL p1, v1000, z100, Tframe** (3p)

- 4.c Robotcellen för IBM7545-roboten i robotlabbet har modifierats genom att DUPLO-plattorna laddas och töms (mycket snabbt) i 2 separata stationer. Klarsignal för roboten att få bygga på någon av plattorna ges genom de 2 digitala ingångarna nr 1 och 2. Ingång 1 aktiveras (=1) när en platta är framatad i station1 och för station 2 aktiveras ingång 2. Skriv ett **optimerat** AML/E-program som ser till att en DUPLO-bit med en LEGO-bit ovanpå, monteras på varje DUPLO-platta. OBS! Frammatningen av plattorna sker slumpmässigt i stationerna. När roboten är klar med en platta ska tillhörande digitala utgång (utgång nr 1 för station 1 respektive utgång nr 2 för station 2) pulsas under 1 sekund (=1 under 1s). Utgångsläget vid programstart är att roboten står i sitt hemmaläge med DUPLO-gripdonet och väntar på att DUPLO-plattor ska matas fram i de för tillfället tomma stationerna. Till Er hjälp finns följande subrutiner redan definierade (behöver inte deklarerars): (6p)

Subrutin	Beskrivning
MONTDUPLO(VAR)	Hämtar och monterar sedan en DUPLO i station VAR, dvs anropet MONTDUPLO(2); innebär att roboten hämtar en DUPLO och monterar denna på DUPLO-plattan i station 2
MONTLEGO(VAR)	Hämtar och monterar sedan en LEGO i station VAR, dvs anropet MONTLEGO(2); innebär att roboten hämtar en LEGO och monterar denna på DUPLO-biten i station 2
DUPLOGRIP	Lämnar först gripdonet för LEGO och hämtar sedan gripdonet för DUPLO
LEGOGRIP	Lämnar först gripdonet för DUPLO och hämtar sedan gripdonet för LEGO

**Befintliga kommandon och syntax:**

<b>Kommando</b>	<b>Beskrivning</b>
WRITEO(DO,VAL);	Utgång DO sätts till värdet VAL
TESTI(DI,VAL,HOPP);	Kontrollerar om en ingång (DI) har värdet VAL, vilket medför hopp till HOPP. Har ingången inte värdet VAL exekveras nästa instruktion
HOPP::;	Markerar en hoppadress (label). Hopp får bara ske inom samma subrutin
BRANCH(HOPP);	Ovillkorligt hopp till hoppadressen HOPP
DELAY(TID);	Uppehåller exekveringen av programmet under TID sekunder
NAMN: SUBR(VAR,VAR1..);	Definierar en subrutin med namnet NAMN och variablerna med namnen VAR,VAR1.. blir lokala
NAMN(VAR,VAR1..);	Anropar subrutinen med namnet NAMN med variablerna VAR,VAR1..
END;	Markerar slutet på subrutin

**Programexempel:**

-- Först definieras huvudsbrutinen  
PROGEX:SUBR;

-- Därefter kommer subrutiner  
SUB:SUBR(VAR);  
LEGOGRIP;  
MONTDUPLO(VAR);  
DUPLOGRIP;  
END;

-- Sist kommer huvudprogrammet

```

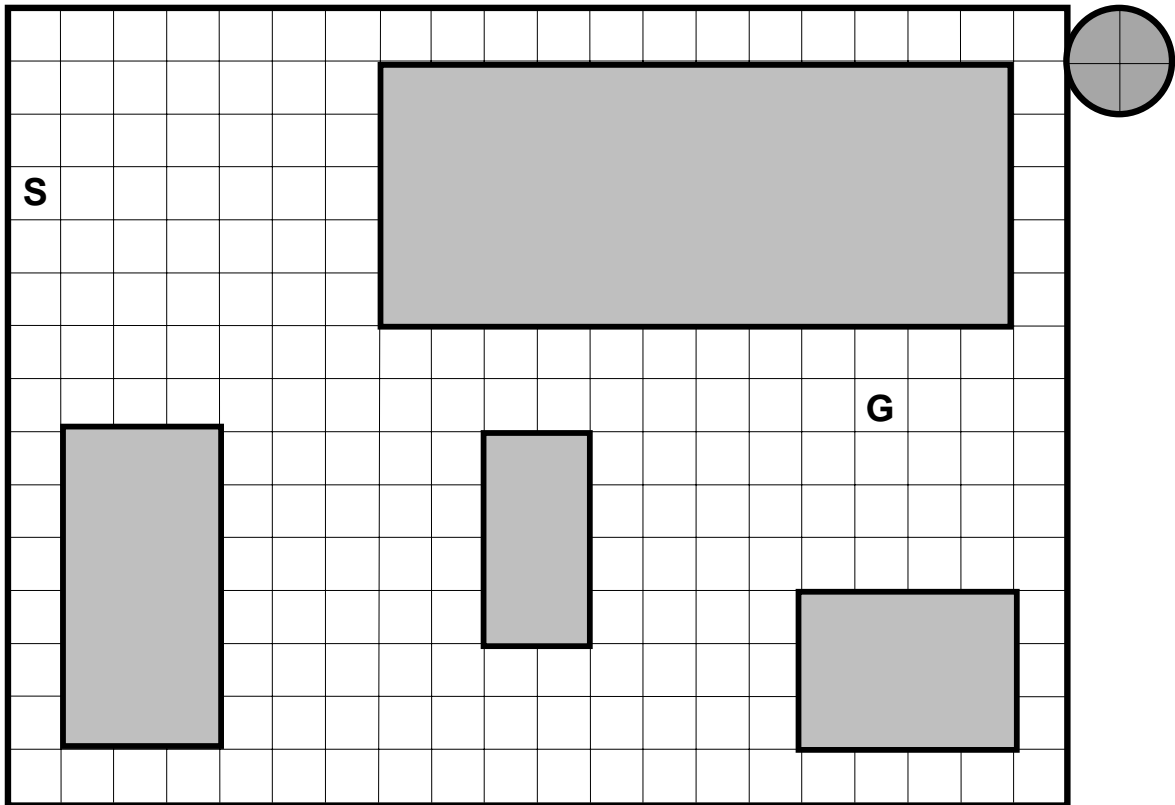
DELAY(2);
TESTI(1,1,SLUT);
BRANCH(SLUT);
SUB(1);
SUB(2);
MONTLEGO(1);
MONTLEGO(2);
WRITEO(2,0);
SLUT::;
END;
```

5

6

## Svarsblankett till fråga 3c

Namn.....Personnummer.....



## Rotationsmatriser

$$\mathbf{Rot}(x, \theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix}$$

$$\mathbf{Rot}(y, \theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$$

$$\mathbf{Rot}(z, \theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$