Matematikken i kunstig intelligens
Opgaver om koordinerende robotter

Thomas Bolander

2. juni 2018

Vejledning til opgaver

- Opgave 1 kan eventuelt springes over, hvis man har mindre tid. De resterende opgaver kan løses uafhængigt af opgave 1.

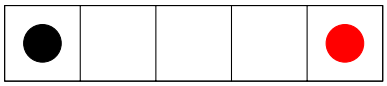
Opgave 1 (Tilstandsrum)

Når man laver kunstig intelligens til rutefinding, lagerrobotter, satelitter, brætspil, computerspil og en lang række andre anvendelser, prøver man ofte at få en fornemmelse af hvor svært problemet er ved at kigge på størrelsen af **tilstandsrummet**. Tilstandsrummet er hvor mange forskellige tilstande et system kan være i, og man bruger ofte teknikker indenfor det område af matematikken der hedder **kombinatorik** for at udregne det.

Betragt f.eks. et helt trivielt 1-personers spil, hvor den sorte brik skal flyttes fra den ene ende af brættet til den anden:

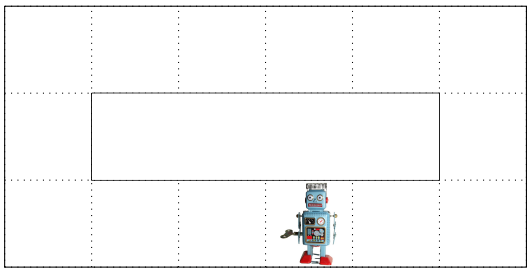
Der er 5 felter som brikken kan stå på, og derfor bliver størrelsen af tilstandsrummet 5.

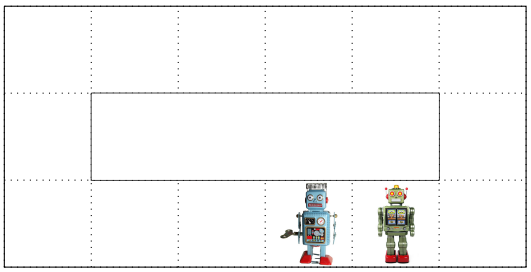
a) Betragt et spil med 2 brikker (vist nedenfor), hvor brikkerne kan flyttes rundt på alle felter uafhængigt af hinanden, undtagen at der ikke må være 2 brikker på samme felt. Hvad er da størrelsen af tilstandsrummet? Hvordan ændrer størrelsen af tilstandsrummet sig hvis den røde brik ikke kan springe over den sorte, det vil sige, den røde brik altid er til højre for den sorte? Og hvordan ændrer størrelsen af tilstandsrummet sig hvis begge brikker er sorte?



Grunden til størrelsen af tilstandsrummet er relevant for kunstig intelligens er, at det giver en øvre grænse på hvor mange muligheder den kunstige intelligens skal tænke igennem for at løse sit problem. Lad os prøve at tage brætspil som eksempel. I kryds-og-bolle er størrelsen af tilstandsrummet 5478, hvilket gør det utroligt nemt at lave et computerprogram som kan spille kryds-og-bolle optimalt. I skak er størrelsen af tilstandsrummet 10^{17}, og allerede af dette kan man få en fornemmelse af at det er meget sværere at lave et computerprogram som er god til at spille skak (hvilket det ganske rigtigt er). I brætspillet Go er størrelsen af tilstandsrummet 10^{172}, hvilket giver en indikation af at det er endnu sværere (hvilket det også netop er). Det er derfor ikke overraskende at det allerede i 1997 lykkedes at lave et computerprogram som kunne blive verdensmester i skak (IBMs Deep Blue), men først sidste år, i 2017, lykkedes det at lave et computerprogram som blev verdensmester i Go (Google DeepMinds AlphaGo).

b) Også hvis vi vil programmere robotter til at kunne løse navigationsproblemer, må vi se på størrelsen af det tilstandsrum de befinder sig i. For hvert af nedenstående eksempler skal du finde størrelsen af tilstandsrummet. Vi antager at der kun kan være 1 robot på hvert felt.

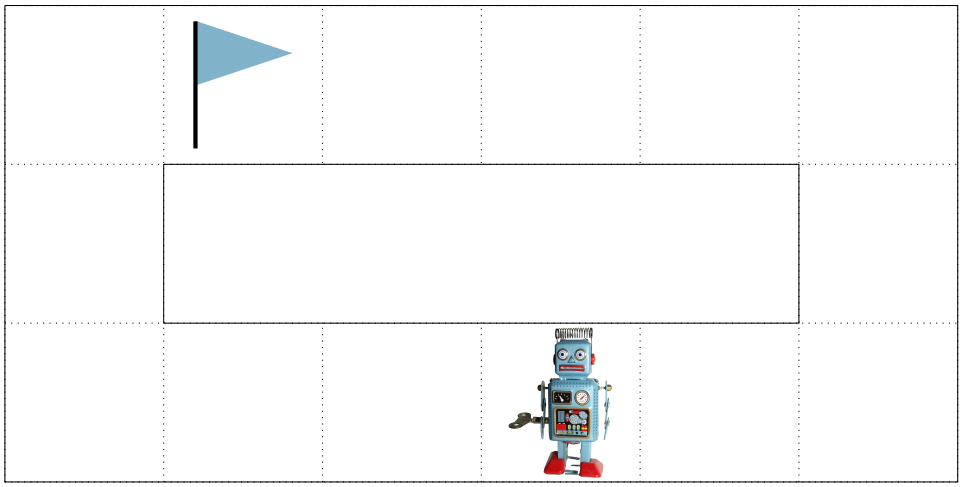




Opgave 2 (Korteste løsning)

En anden vigtig ting indenfor problemløsning med kunstig intelligens er at finde længden af en korteste løsning. Hvis det handler om rutefinding på en GPS eller i Google Maps svarer det til at finde den korteste rute fra A til B. Hvis det handler om at styre en robot, vil man ofte måle længden af en løsning i antallet af operationer som robotten skal udføre.

Betragt følgende eksempel:



Robotten skal navigere hen til det blå flag, og den kan kun flytte ét felt af gangen, enten op, ned, til venstre eller til højre. Når robotten flytter sig ét felt kalder vi det et **træk** ligesom i brætspil. En korteste løsning er i dette tilfælde den korteste vej til det blå flag målt i antal træk. På eksemplet ovenfor har den korteste løsning længden 6, da robotten skal foretage følgende træk:

1. venstre, 2. venstre, 3. venstre, 4. op, 5. op, 6. højre.

Der er også en anden løsning: at gå mod uret rundt. Men den løsning er længere:

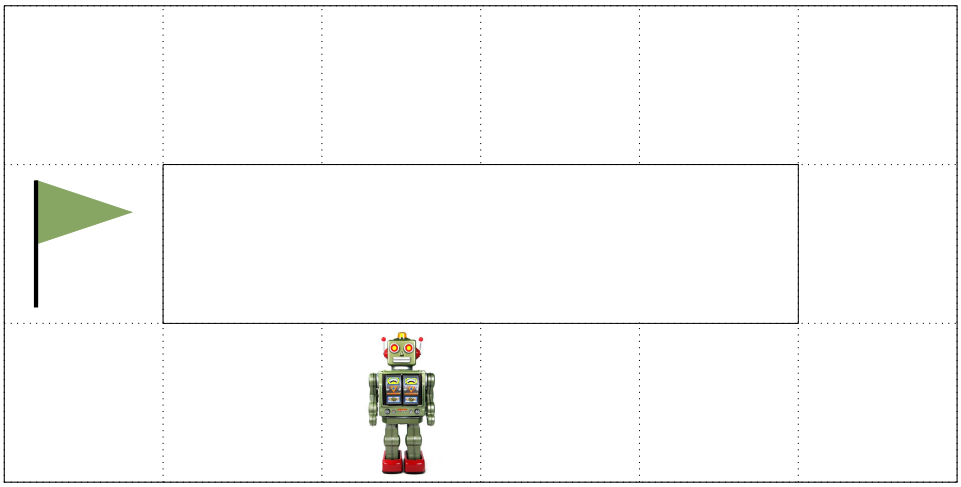
1. højre, 2. højre, 3. op, 4. op, 5. venstre, 6. venstre, 7. venstre, 8. venstre.

Faktisk er der uendeligt mange løsninger, for robotten kan altid flytte sig frem og tilbage mellem to felter så længe den nu gider inden den til slut går hen til flaget.

Løsninger som disse er i virkeligheden en slags simple computerprogrammer, som de kunne se ud på en robot: De fortæller robotten præcist hvad den skal gøre i hvilken rækkefølge. Et sådant program er ikke i sig selv kunstig intelligens, det bliver først kunstig intelligens

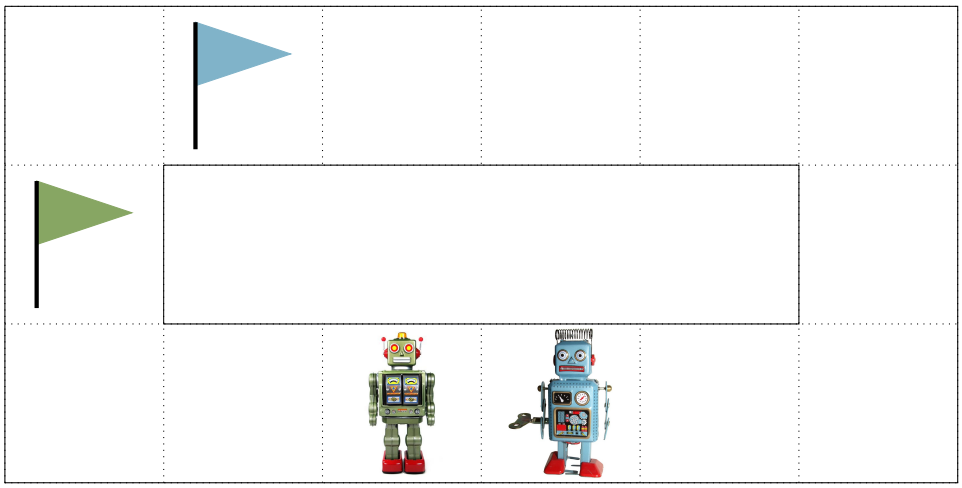
når robotten selv finder løsningerne. Det område af kunstig intelligens som handler om at få robotter eller anden kunstig intelligens til selv at finde løsninger til problemer kaldes **automatiseret planlægning**. Det bygger i vid udstrækning på forskellige områder af diskret matematik, især kombinatorik, algoritmik, induktion, rekursion og logik. Der er lidt diskret matematik i gymnasiet, især på A-niveau, men ellers er det noget man først for alvor ser på universitetet.

a) Betragt følgende eksempel, hvor den grønne robot skal hen til det grønne flag:



Find en korteste løsning, skriv den op på samme form som ovenfor, og angiv dens længde.

Når der er mere end én robot bliver tingene en smule mere kompliceret. Vi vælger her at antage at kun én robot kan bevæge sig ad gangen, men de behøver ikke nødvendigvis bevæge sig turbaseret (skiftevis grøn og blå) som i almindelige brætspil. Det kan f.eks. være at den ene robot først bevæger sig halvdelen af vejen, så bevæger den anden sig i mål, og til slut bevæger den første sig også i mål. Når vi snakker om en løsning skal vi derfor specificere hvem der bevæger sig i hvert træk. Hvis grøn skal bevæge sig op i det 5. træk kan vi f.eks. skrive det som '5. grøn: op'. Betragt følgende eksempel:

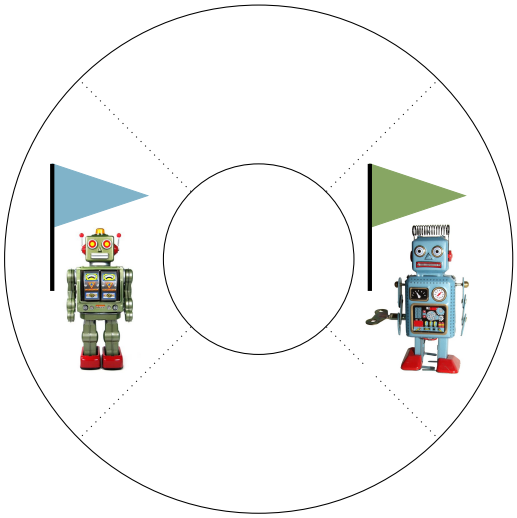


Her er en korteste løsning følgende:

1. blå: højre, 2. blå: højre, 3. blå: op, 4. blå: op, 5. blå: venstre, 6. blå: venstre, 7. blå: venstre, 8. blå: venstre, 9. grøn: højre, 10. grøn: højre, 11: grøn: op.

Længden er 11. Bemærk at længden af en korteste løsning med 2 robotter **ikke** er lig med summen af en korteste løsning med kun den blå robot (som vi ovenfor fandt til 6) og en korteste løsning med kun den grønne robot (som du fandt løsningen til i spørgsmål a).

b) Betragt nu følgende eksempel, hvor begge robotter igen skal til deres respektive målflag:



Her er der kun 4 felter, og robotterne skal bytte plads. Hvert træk er enten at flytte ét felt med uret eller ét felt mod uret. Vi kalder disse træk "drej med uret" og "drej mod uret". Find en korteste løsning, skriv den op på samme form som ovenfor, og angiv dens længde.

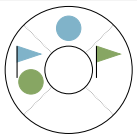
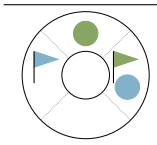
Opgave 3 (Strategier)

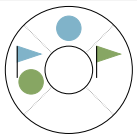
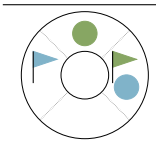
Ovenfor fandt vi løsninger, hvor **vi** bestemte hvad begge robotter skulle gøre. Men normalt vil det jo være robotterne selv der hver især skal beslutte hvad de vil gøre. Og så er det ikke længere nok at skrive en løsning op som en række af træk, for man ved jo ikke nødvendigvis hvad den anden robot har tænkt sig at gøre. I eksemplet ovenfor, hvor robotterne skulle bytte plads, kan det være at robotterne har planlagt at bevæge sig hver sin vej rundt, og så vil de støde sammen (eller den ene robot vil mislykkes).

For at løse sådanne problemer laver man en anden type af løsning som kaldes en **strategi**. En strategi er ikke en simpel række af træk, men en matematisk funktion, som til hver mulig tilstand knytter det træk som robotten skal foretage i den tilstand. Man kan illustrere en strategi som en tabel, hvor der for hver mulig tilstand er angivet hvad robotten skal gøre i tilstanden. Hver robot har så sin egen strategi. I eksemplet ovenfor kan den grønne robot lave strategien vist til venstre på figur 1, hvor vi har byttet de to robotter ud med farvede cirkler, for at gøre det tydeligere. Den første linje viser f.eks. at at den grønne robot har planlagt at starte med at dreje med uret. Hvis det lykkes for den blå robot at bevæge sig mod uret inden den grønne laver sit første træk, vil den grønne i stedet bevæge sig mod uret. Det ses af tredje linje i strategien nedenfor.

a) Udfyld resten af strategien for den blå robot i figur 1. Tænk over hvad der vil give mest mening for den blå robot at gøre i hver tilstand. Målet er stadig at både den blå og grønne robot ender på det felt hvor deres eget flag er.

b) Du skal nu afprøve strategierne fra spørgsmål a. Klip de to cirkler ud af figur 2 og brug dem som robotter. Sæt dem til at starte med på hinandens flag. Kast så med en mønt for at finde ud af hvem der skal foretage et træk. Hvis du slår plat, skal den grønne

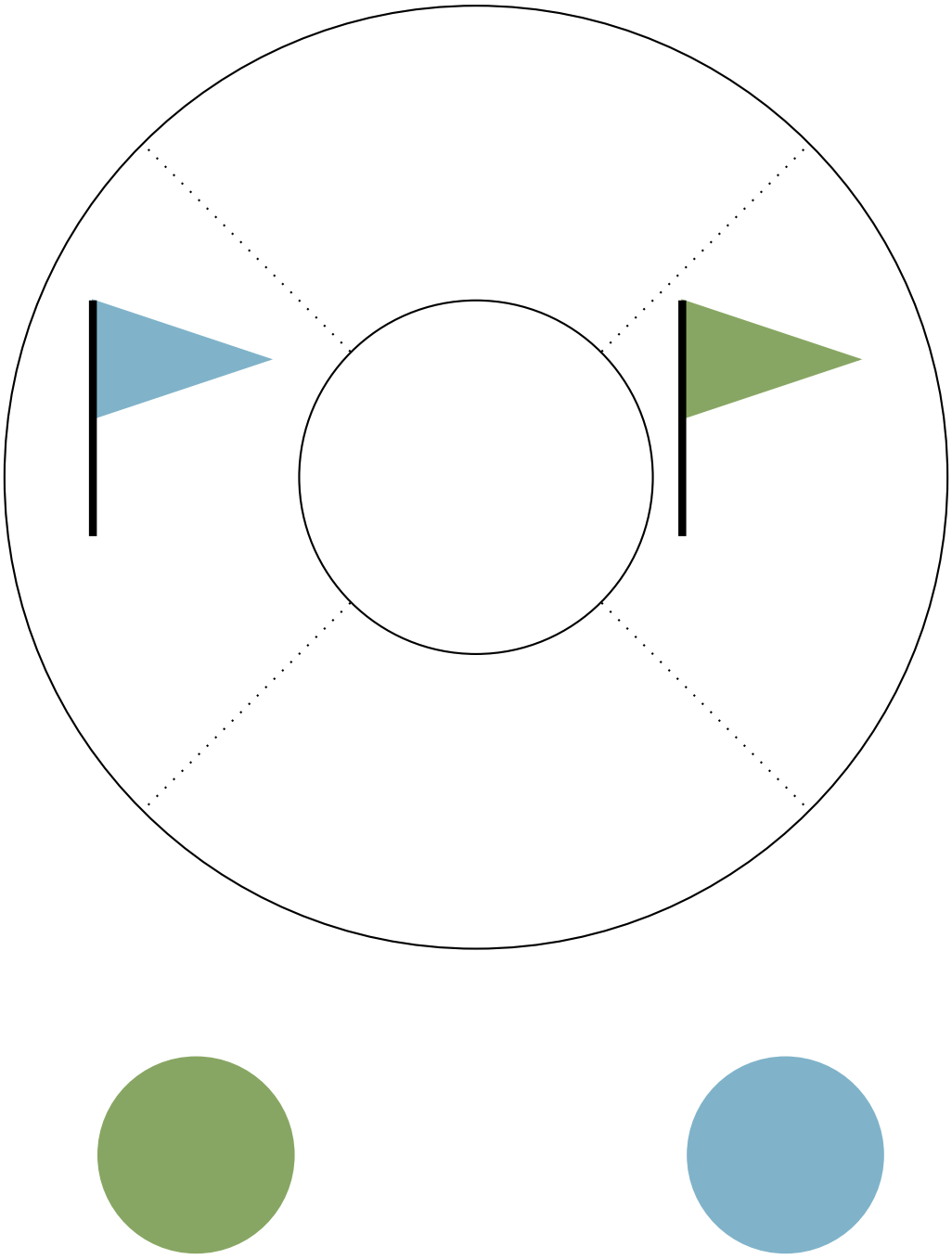
	drej med uret
	drej med uret
	drej mod uret
	vent
	drej med uret
	drej med uret
	stop
	stop
	vent
	drej mod uret
	drej mod uret

	drej mod uret
	
	
	
	
	stop
	
	
	
	
	stop

Figur 1: Strategier for den grønne og blå robot.

bevæge sig, ellers den blå. Den som skal bevæge sig, skal bevæge sig ifølge strategierne i tabellen i figur 1. Efter første træk kaster du mønten igen, og bruger igen dens udfald til at bestemme hvem der skal rykke. Sådan fortsætter du, med ét møntkast per træk. Så snart du når til en tilstand hvor den ene robots træk hedder "stop" skal du ikke længere kaste mønten, men blot følge strategien for den anden robot, indtil den også når til "stop". Forhåbentlig ender det så med at de to robotter ender hvor de skal og ikke støder ind i hinanden undervejs. Hvis de ikke ender hvor de skal, eller hvis de støder ind i hinanden, er der noget galt med stratagierne, og du må prøve at rette din strategi for blå. Prøv at spille spillet nogen gange og se hvor mange træk der bliver brugt i alt hver gang (at vente tæller også som et træk).

I sidste spørgsmål ovenfor viser du kun at strategierne virker ved at afprøve dem nogen gange. Normalt vil man forsøge at give et matematisk bevis for at en strategi er korrekt, og måske også at det er den bedste (den som giver de korteste løsninger). I kunstig intelligens prøver man ofte at give et matematisk bevis for at en bestemt metode til at lægge strategier altid giver en korrekt strategi (eller altid den bedst mulige). Dette kan af og til være ret udfordrende, især i det tilfælde hvor robotterne ikke engang nødvendigvis ved hvor de andre robotter skal hen. Det er stadig et aktivt forskningsområde. Anvendelserne er naturligvis ikke kun robotter på spilleplader, men generelt robotter der skal koordinere deres handlinger i forhold til hinanden, f.eks. når de gerne vil samarbejde eller hjælpe hinanden.



Figur 2: