

---

*INF5390 – Kunstig intelligens*

*Øving 2 - Gjennomgang*

Roar Fjellheim

# Øving 2.1

---

## *Agents That Plan (INF5390-AI-06)*

En ape skal hente en klase bananer som henger i taket. Det er en skyvbar boks som apen kan klatre opp på og som er høy nok til at han da når bananene. Initielt er apen i posisjon *A*, bananene i *B* og boksen i *C*. Apens aksjoner er *Go* fra et sted til et annet, *Push* en gjenstand fra et sted til et annet, *ClimbUp/ClimbDown* en boks og *Grasp/UnGrasp* en gjenstand.

- Skriv ned de 6 aksjonsskjemaene i PDDL.
- Bruk PDDL-notasjon for å skrive ned initiell tilstand.
- Skriv ned en aksjonssekvens som slutter med at bananene er på gulvet i *A*.

## 2.1a: Aksjonsskjemaer i PDDL

---

- *Action*(ACTION: *Go*( $x, y$ ),  
PRECOND: *At*(*Monkey*,  $x$ ),  
EFFECT: *At*(*Monkey*,  $y$ )  $\wedge$   $\neg$ (*At*(*Monkey*,  $x$ )))
- *Action*(ACTION: *Push*( $b, x, y$ ),  
PRECOND: *At*(*Monkey*,  $x$ )  $\wedge$  *Pushable*( $b$ ),  
EFFECT: *At*( $b, y$ )  $\wedge$  *At*(*Monkey*,  $y$ )  $\wedge$   $\neg$ *At*( $b, x$ )  $\wedge$   
 $\neg$ *At*(*Monkey*,  $x$ ))
- *Action*(ACTION: *ClimbUp*( $bo$ ),  
PRECOND: *At*(*Monkey*,  $x$ )  $\wedge$  *At*( $bo, x$ )  $\wedge$  *Climbable*( $bo$ ),  
EFFECT: *On*(*Monkey*,  $bo$ ))

## 2.1a: Aksjonsskjemaer i PDDL (cont.)

---

- *Action*(ACTION: *Grasp*(*ba*),  
PRECOND:  $At(Monkey, x) \wedge At(bo, x) \wedge On(Monkey, bo)$ ,  
EFFECT:  $Have(Monkey, ba)$ )
- *Action*(ACTION: *ClimbDown*(*bo*),  
PRECOND:  $On(Monkey, bo)$ ,  
EFFECT:  $\neg On(Monkey, bo)$ )
- *Action*(ACTION: *UnGrasp*(*ba*),  
PRECOND:  $Have(Monkey, ba)$ ,  
EFFECT:  $\neg Have(Monkey, ba)$ )

## 2.1b: Initiell tilstand

---

*At(Monkey,A)  $\wedge$*

*At(Bananas,B)  $\wedge$*

*At(Box,C)  $\wedge$*

*Pushable(Box)  $\wedge$*

*Climbable(Box)*

## 2.1c: Aksjonssekvens

---

GOAL:  $At(Monkey, A) \wedge$   
 $At(Bananas, A) \wedge$   
 $\neg Have(Monkey, Bananas)$

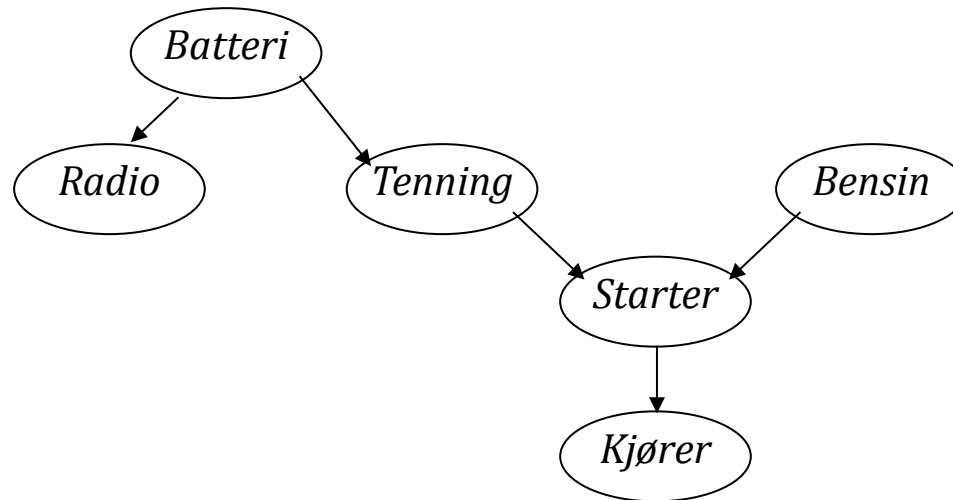
PLAN:  $Go(Monkey, A, C),$   
 $Push(Box, C, B),$   
 $Climb(Box),$   
 $Grasp(Bananas),$   
 $ClimbDown(Box),$   
 $Go(Monkey, B, A),$   
 $UnGrasp(Bananas)$

## Øving 2.2

---

### *Agents That Reason Under Uncertainty (INF5390-AI-08)*

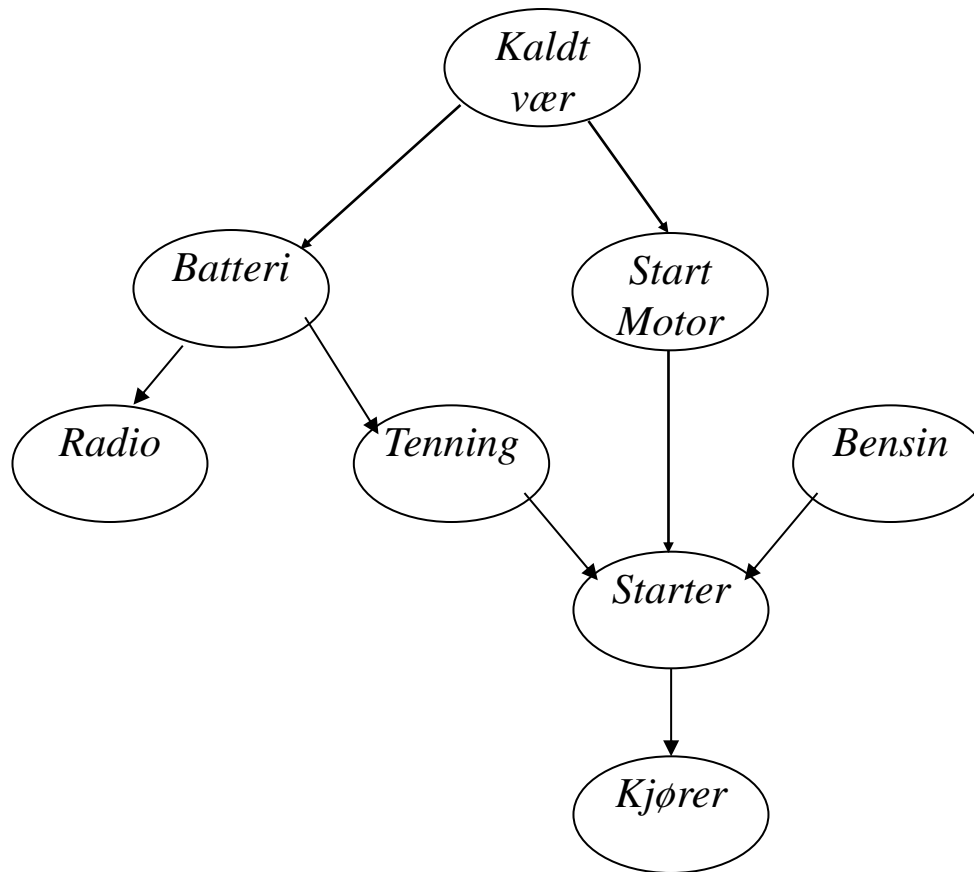
Anta at vi har et Bayesiansk nettverk for problemet at en bil ikke vil starte, se figur. Hver variabel er Boolsk og verdien *true* betyr at vedkommende funksjon fungerer som den skal.



- Utvid nettverket med de Boolske variablene *Kaldt vær* og *StartMotor*.
- Beskriv i ord strukturen av CPT (Conditional Probability Table) for variabelen *Starter* (dvs. at motoren starter).

## 2.2a: Utvidet nettverk

---





## 2.2b: CPT for *Starter*

- CPT for *Starter* inneholder et sett av nødvendige betingelser som tilsammen er nesten tilstrekkelig for at motoren starter
- Dvs. at alle CPT entries er 0 unntatt for den linjen hvor alle betingelsene er *true*
- Den linjen vil være svært nær 1, men ikke helt 1 fordi det kan være ytterligere feilkilder

Tenning	StartMotor	Bensin	P(Starter)
F	F	F	0
F	F	T	0
F	T	T	0
T	T	T	0.999
T	T	F	0
.....			

## Øving 2.3

---

### *Agents That Learn (INF5390-AI-10)*

Anta at vi genererer et treningssett fra et beslutningstre og deretter anvender algoritme for læring av beslutningstre på treningssettet. Vil læringsalgoritmen returnere det korrekte treet når treningssettet går mot uendelig stort? Hvorfor, alternativt hvorfor ikke?

## 2.3: Identiske beslutningstrær?

---

- Algoritmen vil ikke nødvendigvis generere det “korrekte” (det samme) treet, men vil generere et tre som er logisk ekvivalent, gitt at metoden for å generere eksempler til slutt genererer alle mulige kombinasjoner av input attributter.
- To beslutningstrær definert over det samme sett av attributter og som stemmer overens for alle mulige eksempler, er logisk ekvivalente pr. definisjon.
- Men de to trærne kan ha forskjellig form fordi det er mange måter å representere en funksjon på. F.eks. med to attributter A og B kan vi ha et tre med A som rotnode og et annet med B som rotnode (forskjellig valg for splitting av eksemper).