

729G78 Artificiell intelligens

Introduktion

Robin Keskisärkkä

HCS/IDA

Introduktion

- Kursöversikt
- Mål med AI
- Historia
- Intelligent agenter

Kursöversikt

- Intelligenta agenter
 - 1 Fö, 3 Lab + 1 reserv
- Problemlösning och sökning
 - 2 Fö, 4 Lab + 1 reserv
- Kunskapsrepresentation och logik
 - 6 Fö, 1 Le, 4 Lab + 1 reserv
- Planering
 - 1 Fö, 2 Lab
- Probabilistisk logik
 - 2 Fö, 1 Le, 2 Lab
- Maskininlärning, ANN
 - 3 Fö, 3 + 2 Lab + 2 reserv

Kursens organisation

- Föreläsningar (Robin)
- Lektioner (Robin)
- Laborationer
 - Erik Holgersson
 - Fredrik Fernlund
 - Gustaf Söderholm
 - Javad Saeedizade
 - Mio Cajander

Tidsplan

- 30 h föreläsning
- 6 h lektion
- ca 120 h egen tid för inläsning av teori
- 42 h schemalagda laborationer
 - Laborationsförberedande video
- ca 150 h av egen tid för laborationerna, mycket beroende på hur bra programmeringskursen gick

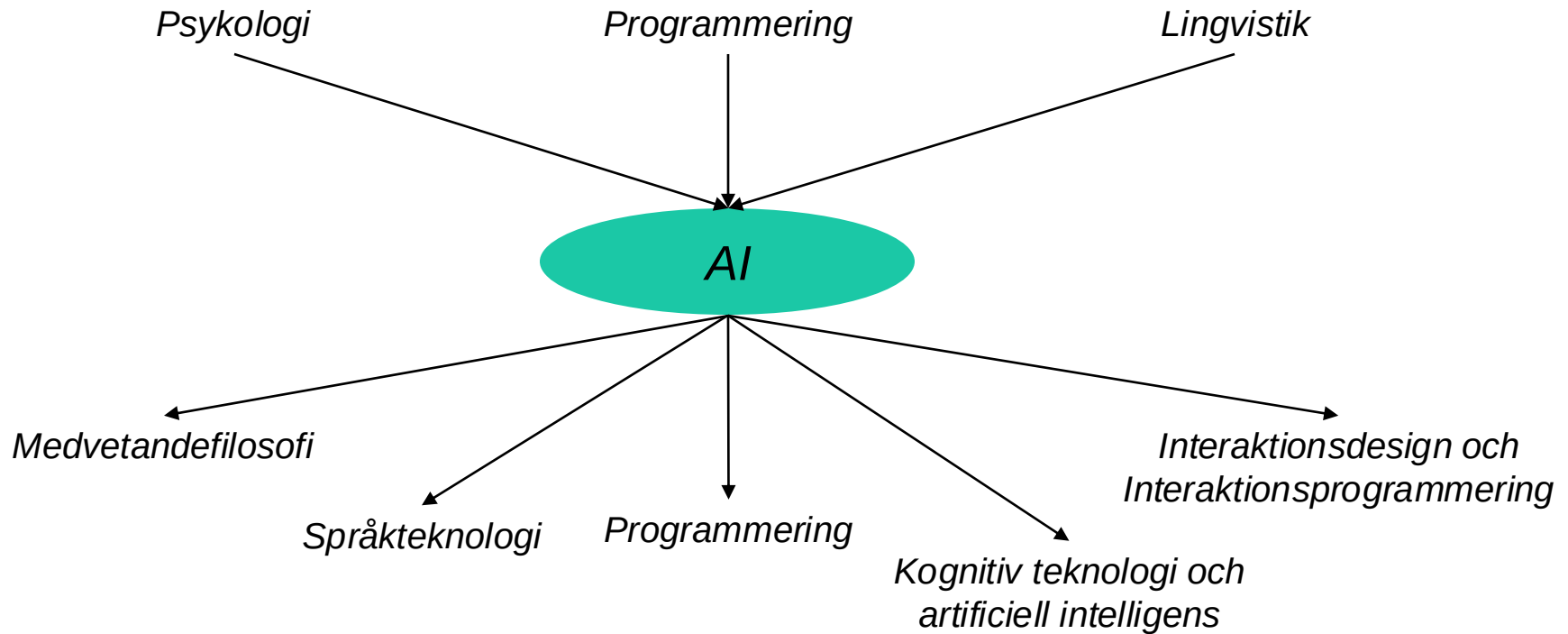
Kursens examination

- Tenta, G/VG (4 hp)
 - Laborationer, G/VG (5 hp)
 - VG på kurs om VG på tenta *eller* VG på labbar
 - Deadlines
-
- Misstänkt fusk anmäls alltid till disciplinnämnden

Kursvärdering 2023

- Helhetsbetyg på kurs: 4,27
 - Svarsfrekvens ca 42%
 - Relevant kurs
- Föreläsningarna går ibland för snabbt fram
- Förtydligande av labbinstruktioner
- Snabbare feedback på inlämnade labbar
- Lektioner kring resolution och naturlig deduktion

Förkunskaper och påbyggnad



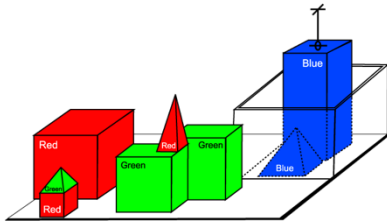
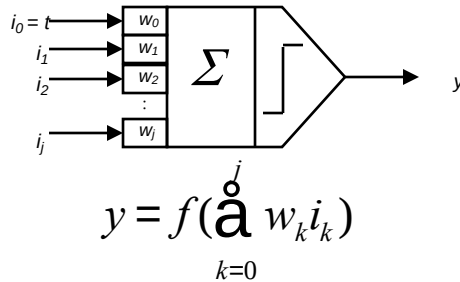
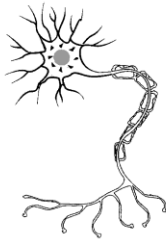
Vad är AI?

- "Artificial Intelligence is the study of how to make computers do things at which, at the moment, people are better." (Elaine Rich)
- "Artificial Intelligence is the study of mental faculties through the use of computational models." (Eugene Charniak & Drew McDermott)
 - Kognitionsvetenskap
- "Understand and build intelligent entities." (Russel & Norvig)
 - Handla rationellt
 - Ska vara autonomt och mätbart

Mål med AI system



Historia



SHRDLU (1968–1970)

...

Person: Why did you do that?

Computer: BECAUSE YOU ASKED ME TO.

Person: How did you do it?

Computer: BY PUTTING A LARGE RED BLOCK ON THE TABLE ; THEN LETTING GO OF IT ;
THEN PUTTING A LARGE GREEN CUBE ON IT ; THEN LETTING GO OF THAT CUBE ;
THEN PUTTING THE RED CUBE ON THAT CUBE; THEN LETTING GO OF THAT CUBE.

...

- Förhistoria: 1943 neuronmodell
 - Kopplade ihop kunskap om hjärnans uppbyggnad med neurala nät och logik
- 1952–69 stora förväntningar
 - Dartmouth-konferensen (1956), General Problem Solver (GPS), mikrovärldar (ex. blocks world), perceptroner som kan tränas

Historia (forts.)

- 1966–74 bakslag

- Ingen kunskap, ex Eliza, maskinöversättning
Engelska $\Rightarrow$ Ryska $\Rightarrow$ Engelska

The spirit is willing but the flesh is weak $\Rightarrow$ Дух готов, но плоть слаба
Дух готов, но плоть слаба $\Rightarrow$ The vodka is good but the meat is rotten

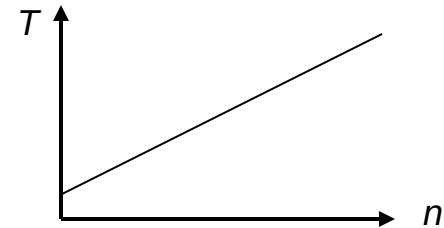
- Fundamentala begränsningar, ex perceptroner kan inte klara XOR
- Komplexitetsproblem

Komplexitetsanalys

- Uppskatta komplexiteten hos ett program

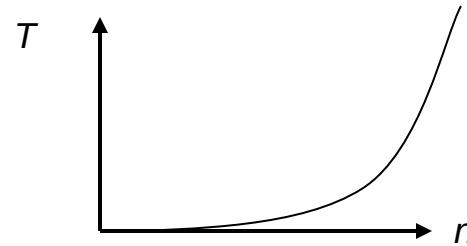
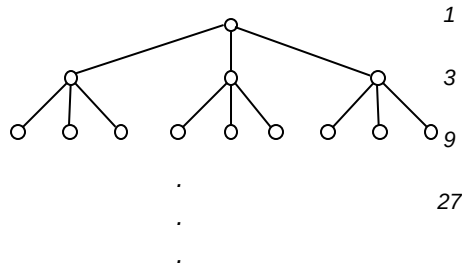
```
def summa(lista):  
    if lista==[]:  
        return 0  
    else:  
        return lista[0] + summa(lista[1:])
```

- Antal operationer:
 - (1 test + 1 addition) * varv + 1 sluttest
- Tidskomplexitet:
 - $T(n) = 2n + 1$, där n är längden på listan



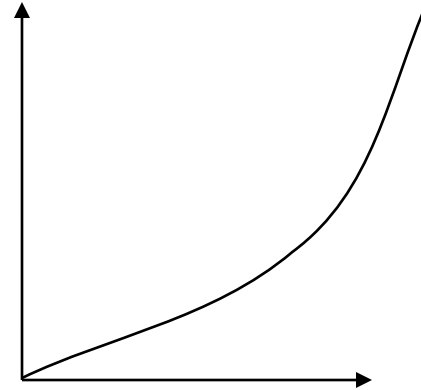
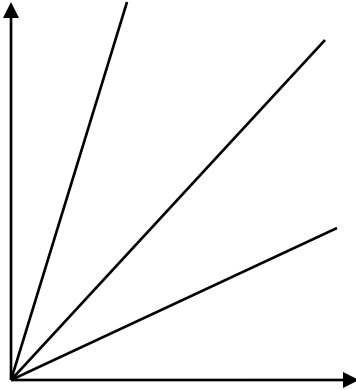
Komplexitetsanalys (forts.)

- Asymptotisk analys
 - Uppskatta vad som kan hända för stora värden på n
 - Ordo-notationen: $O(n)$
- Ex 1. $T(n) = 2n + 1$ ger: $O(n)$ *(polynomiska problem)*
- Ex 2. $T(n) = n^4 + 600n$ ger $O(n^4)$
- Ex 3. Sökträd med förgreningsfaktor b och sökdjup d *(exponentiellt problem)*
ger $O(b^d)$



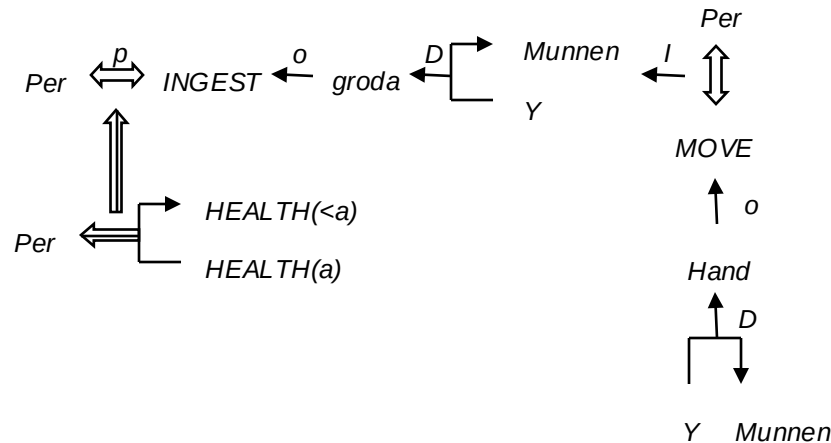
Komplexitetsanalys (forts.)

- “Enkla” problem är polynomiska (P)
Ex. $O(\log n)$, $O(n)$, $O(n^{100})$
- Svåra problem är icke-polynomiska (NP)
Ex. $O(e^n)$

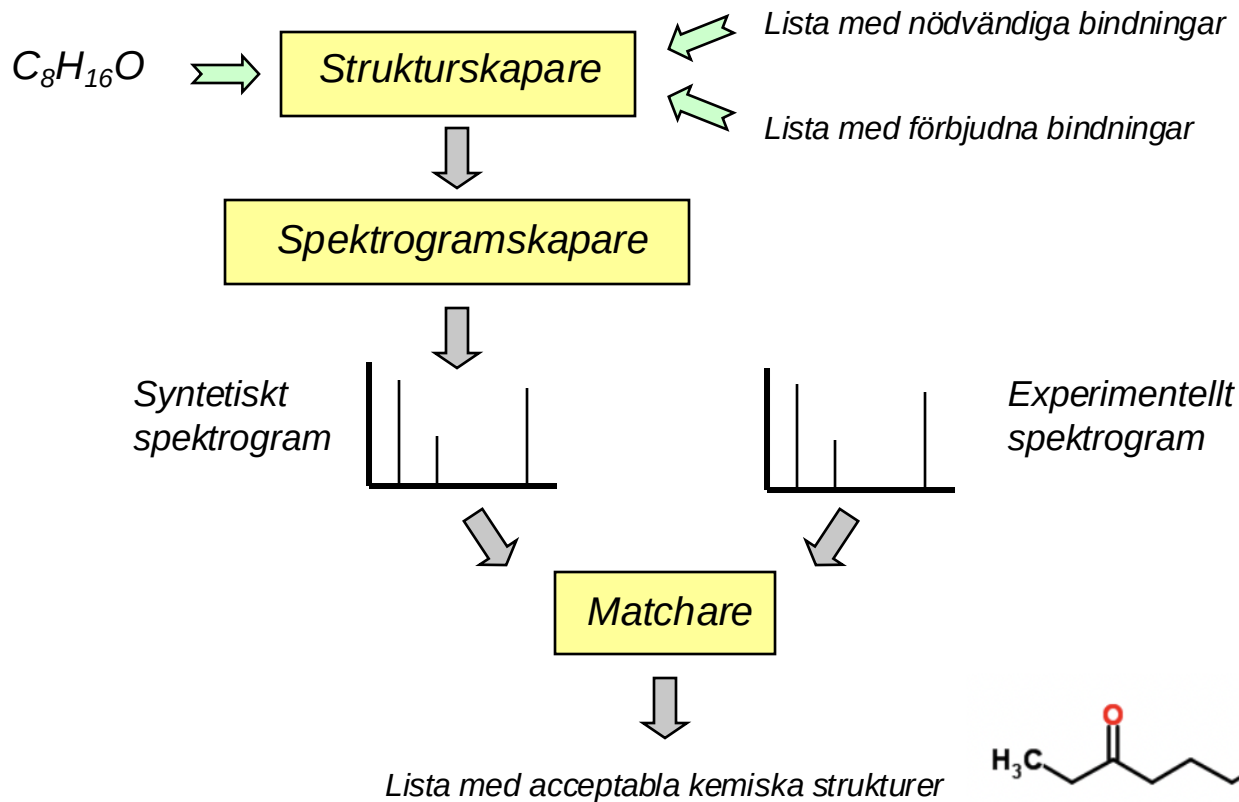


Historia (forts.)

- 1969–86 kunskapsbaserade system (1980–88 expertsystem i industrin)
 - ex. DENDRAL (organisk kemi), MYCIN (bakterieinfektioner), ...
 - Conceptual Dependencies (CD)
 - Grupperade till primitiva handlingar, ex. PTRANS = fysisk överföring (ex. flytta ett objekt), ATRANS = överföring av abstrakt relation (ex. ger eller får)
 - Handlingar och roller kopplade till dessa används vid tolkning

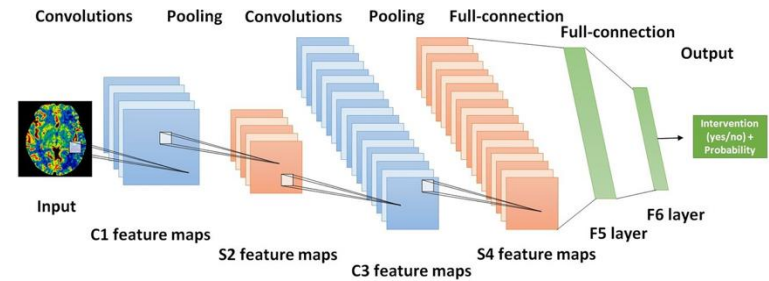


Dendral (Stanford)



Historia (forts.)

- 1986- Neurala nät
- 1987- Mogen vetenskap
 - Sannolikhet
 - Maskininlärning
 - Verkliga problem
- 2001- Stora datamängder
 - Miljarder istället för miljoner exempel
- 2011- Djupa neurala nät
 - Återkopplade



Google

Symbolisk vs subsymbolisk kognition

Symbolisk

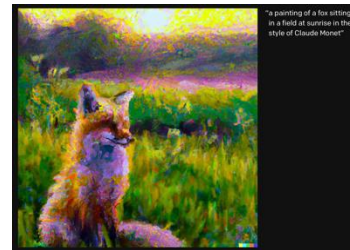
- Kunskapen representerad som symboler, explicit
 - Objekt och relationer
- Logik, planer, ontologier
- Kunskap genom att koda
- Inferens genom sökning i kunskapsbasen

Subsymbolisk

- Kunskapen representerad subsymboliskt, implicit
 - Vikter i ett nätverk
- Vektorer, matriser
- Kunskap genom inläring
- Inferens genom aktivering

Tillämpningar

- Spel, ex IBM Deep Blue, AlphaGo
- Data- och textanalys
- Autonom planering och kontroll, ex självkörande bilar
- Rekommendationssystem
- Bildigenkänning och bildskapande
- Naturligt språk, maskinöversättning, taligenkänning



Intelligenta agenter

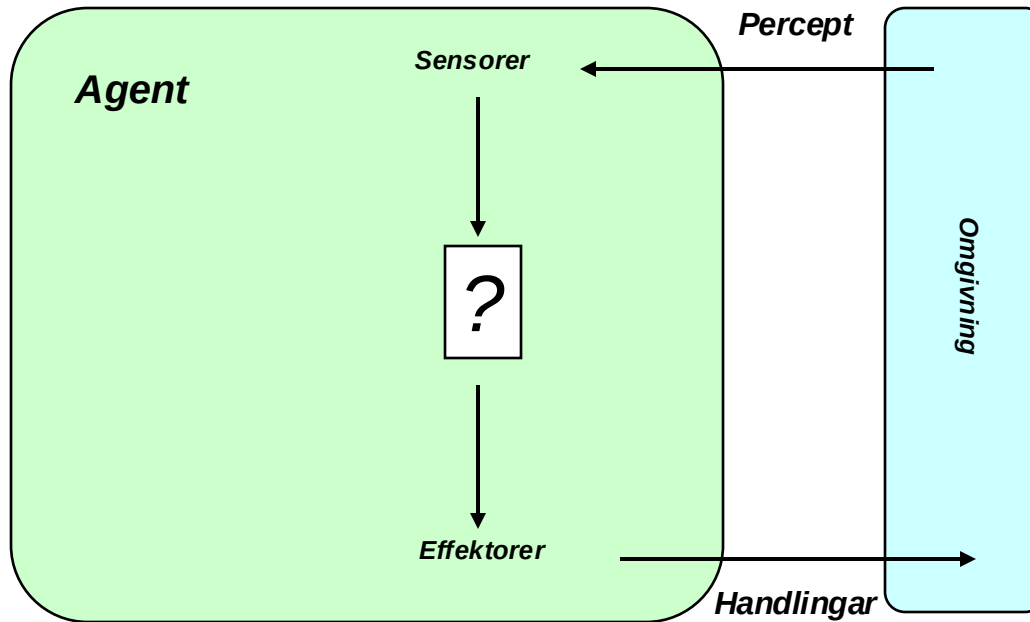
Intelligenta agenter

- Vad är en intelligent agent?
- Typer av intelligenta agenter
 - Enkel reflexstyrd
 - Modellbaserad
 - Målstyrd
 - Nyttostyrd
 - Agent som lär sig

Intelligent agent

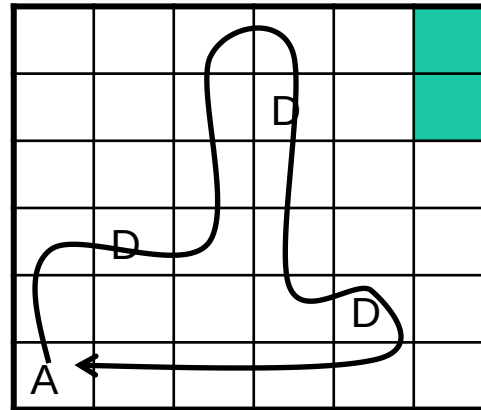
- Rationell
 - Handla så bra som möjligt
 - Mäta agentens utförande
 - Mappa från varseblivningssekvenser till handlingar
 - Vad agenten vet om omgivningen
 - Varseblivningshistoria
 - Handlingar
- Autonom
 - Agentens handlingar beror bara på agentens kunskap samt vad den varseblir

Intelligent agent



Ex. Dammsugaragent

Agenttyp	Utförandemått	Omgivning	Handlingar	Sensorer
Dammsugar-agent	Städa rent och gå hem, effektivt	Rum, damm, hinder	Suga, framåt, vänster, höger, stäng av	Väggsensor, dammsensor, hemmasensor



Andra exempel på agenter

Agenttyp	Utförandemått	Omgivning	Effektorer	Sensorer
Medicinsk diagnos	Frisk patient min. kostnad	Patient, sjukhus, personal	Ställa frågor, tester, diagnoser, behandlingar	Symptom, patientsvar, datainmatning
Satellitbildanalys	Riktig kategorisering	Bilder från satellit	Visa kategorisering	Bildpunkter
Plockrobot	Antal delar i rätt låda	Rullband med delar, lådor	Robotarm	Kamera, robotsensorer
Raffinaderi-kontroll	Maximal renhet, avkastning, säkerhet	Raffinaderi, operatörer	Ventiler, pumpar, displayer, värmare	Temperatur, tryck, kemiska sensorer
Interaktiv lärare	Maximera studentens poäng	Studenter	Visa övningar, föreslå, rätta	Tangentbord

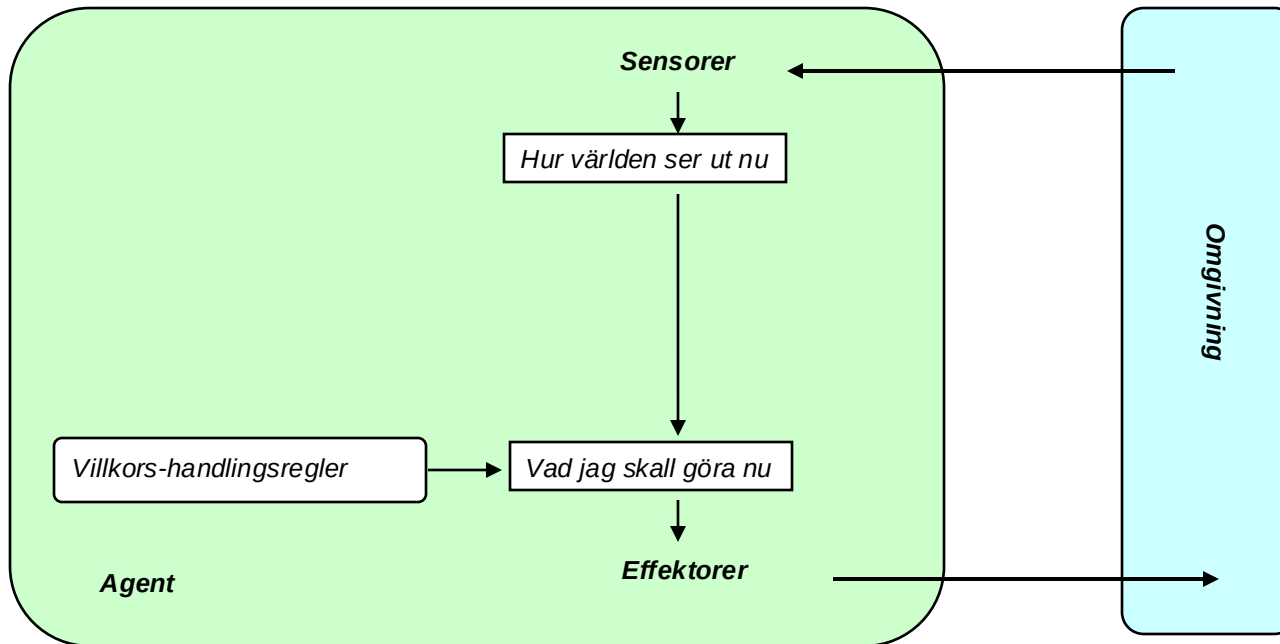
Egenskaper hos omgivningar

- Helt observerbar
 - Sensorerna ger allt som behövs för att välja handling
- Deterministisk
 - Nästa tillstånd beror bara av agentens handling och nuvarande tillstånd
- Episodisk
 - Handling beror inte av vad som hänt tidigare
- Statisk
 - Omgivningen ändras inte medans agenten fattar beslut
- Diskret
 - Tid och tillstånd diskreta
- En agent
- Partiellt observerbar
 - Sensorerna ger inte allt. Kan också vara icke-observerbar
- Stokastisk
 - Kan inte vara säker på utfallet av en handling
- Sekvensiell
 - Beslut om val av handling i ett tillstånd kan påverka framtida val
- Dynamisk/Semidynamisk
 - Omgivningen kan ändras medans agenten fattar beslut
- Kontinuerlig
 - Tillstånd och tid kontinuerliga
- Multiagent

Exempel på omgivningar

Omgivning	Observerbar	Determ.	Episodisk	Statisk	Diskret	Agenter
Schack	Helt	Determ.	Sekventiell	Semi	Diskret	Multi
Taxikörning	Partiellt	Stokastisk	Sekventiell	Dynamisk	Kontinuerlig	Multi
Plockrobot	Partiell	Stokastisk	Episodisk	Dynamisk	Kontinuerlig	En agent
Instruktör	Partiell	Stokastisk	Sekventiell	Dynamisk	Diskret	Multi
Bildanalys	Helt	Determ.	Episodisk	Semi	Kontinuerlig	En agent
Poker	Partiell	Stokastisk	Sekventiell	Statisk	Diskret	Multi
Raffinaderi-kontroll	Partiell	Stokastisk	Sekventiell	Dynamisk	Diskret	Multi

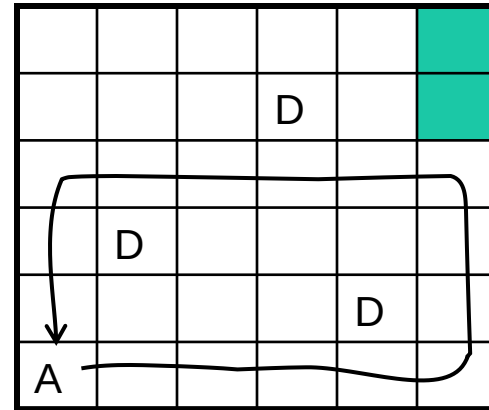
Enkel reflexstyrd agent



Enkel reflexstyrd agent

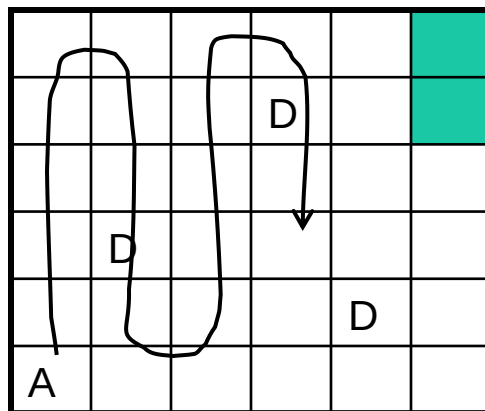
```
rules = {  
    "dirt": "suck",  
    "bump": "turn-left",  
    ... }
```

```
def reflexAgent(percept):  
    state = interpretInput(percept)  
    rule = ruleMatch(state, rules)  
    action = ruleAction(rule)  
    return action
```

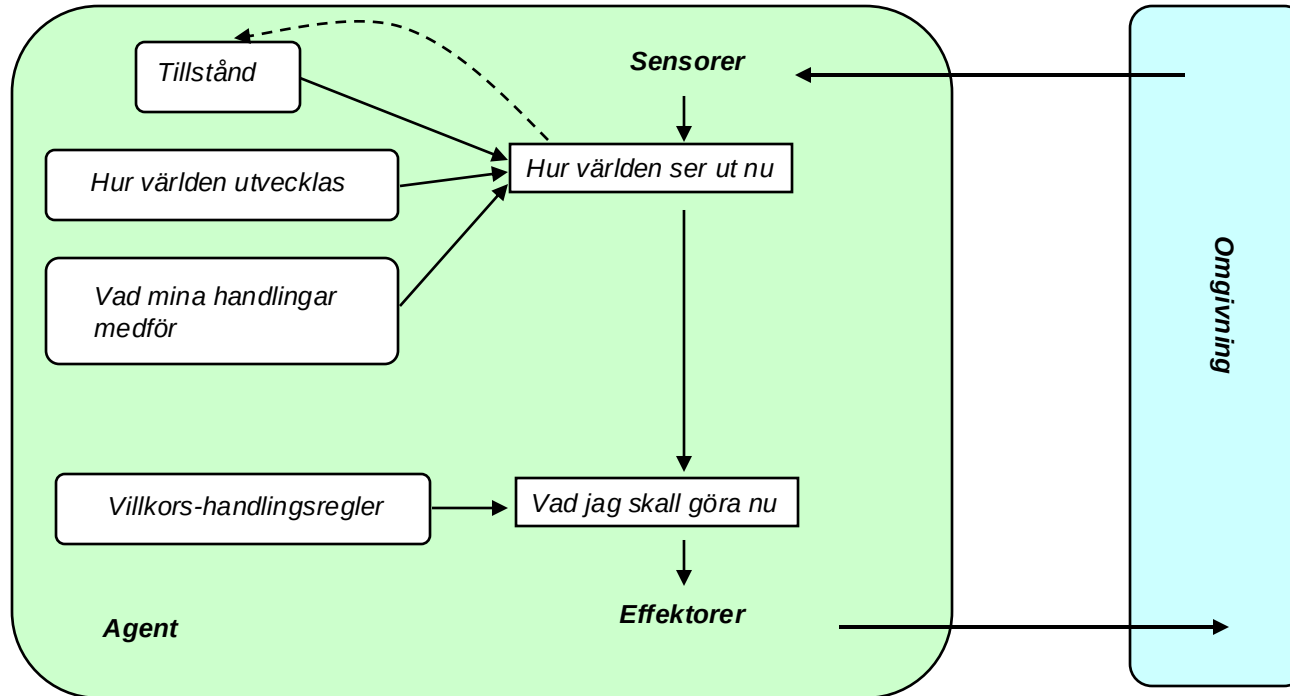


Problem

- Agenten kan inte gå systematiskt



Modellbaserad reflexstyrd agent



Modellbaserad reflexstyrd agent

```
rules = { ... }
```

```
state = [ ... ]
```

```
def reflexAgentWithState(percept):
```

```
    state = updateState(state, action, percept)
```

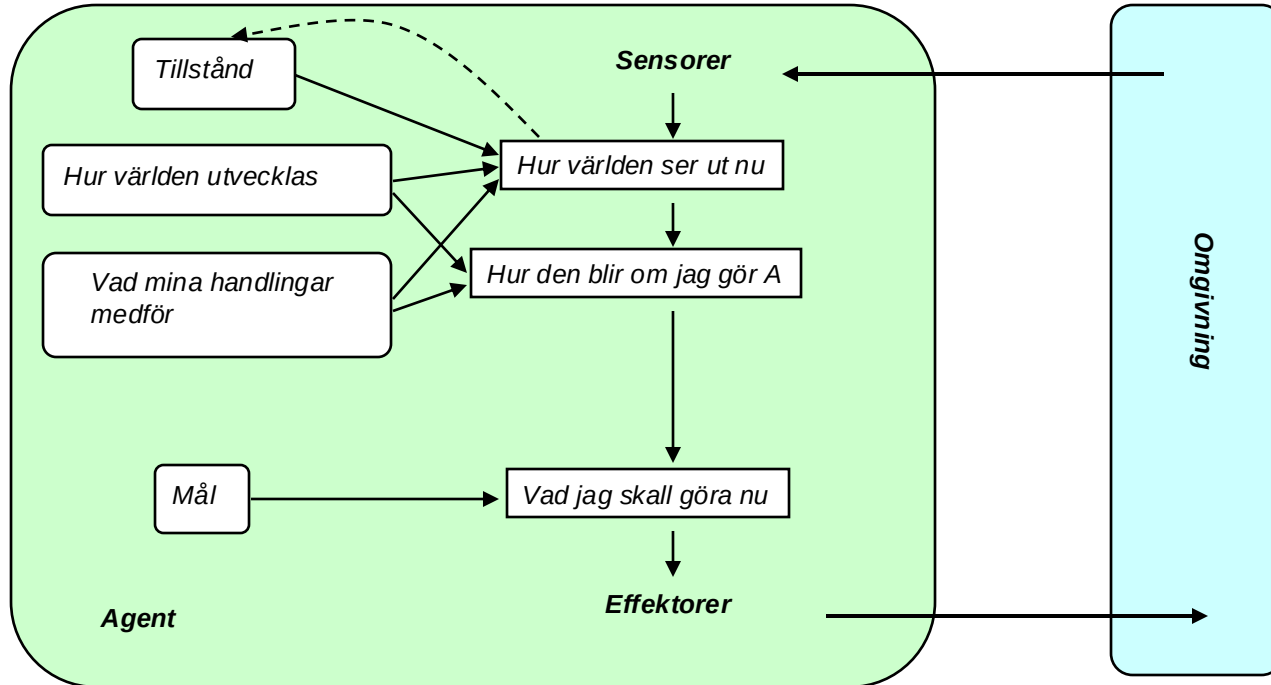
```
    rule = ruleMatch(state, rules)
```

```
    action = ruleAction(rule)
```

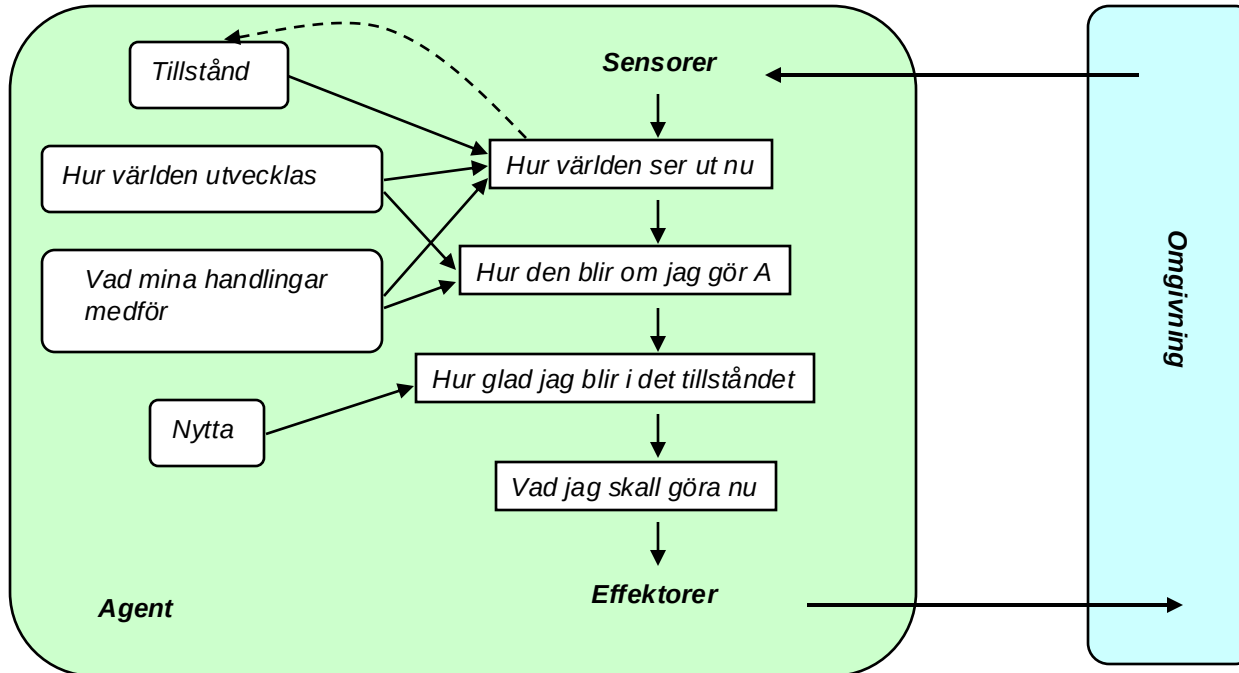
```
    return action
```



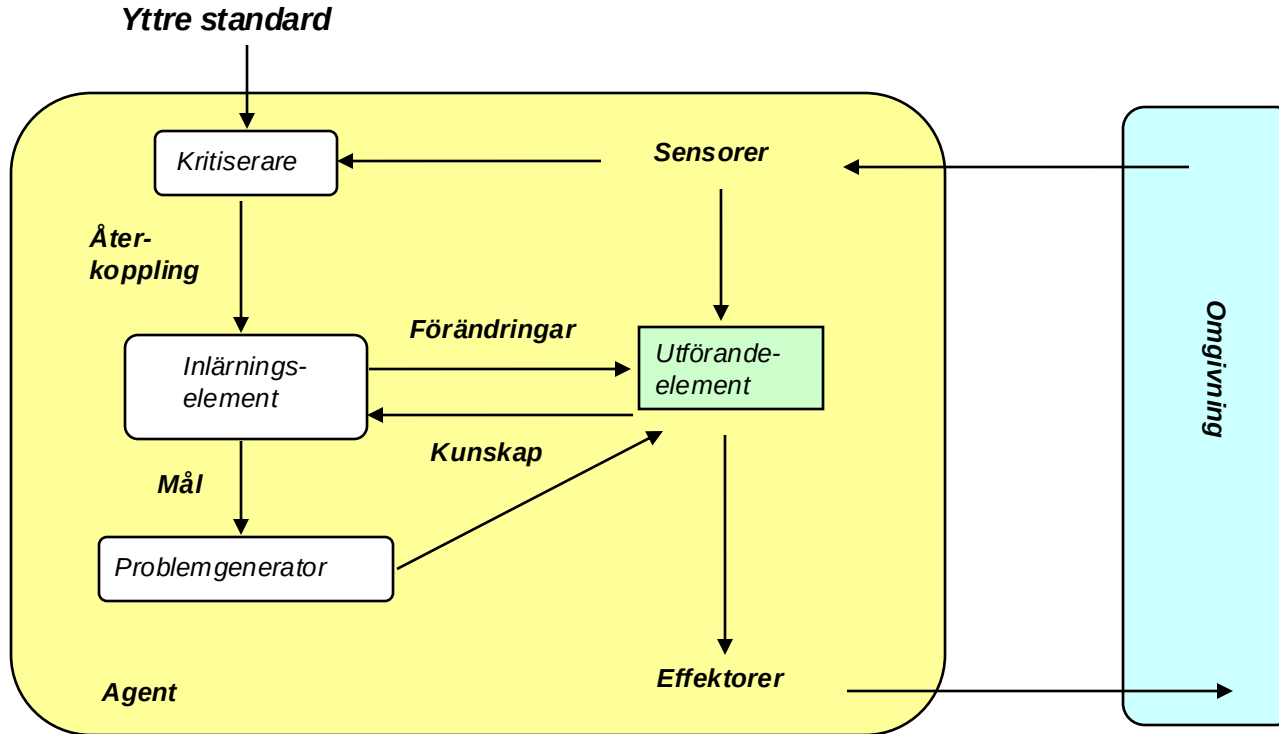
Målstyrd agent



Nyttobaserad agent



Agent som lär sig



Sammanfattning

- Olika mål med AI
- Komplexitetsanalys
- Symbolisk vs subsymbolisk kognition
- Intelligent agenter
 - Enkel reflexstyrd
 - Modellbaserad
 - Målstyrd
 - Nyttobaserad
 - Agent som lär sig
- Egenskaper hos omgivningen
 - Observerbarhet
 - Statisk eller deterministisk
 - Sekventiell eller episodisk
 - Dynamisk eller statisk
 - Kontinuerlig eller diskret
 - En eller flera agenter

