Machine Intelligence

Lecture 1: Introduction to MI and Agents

Thomas Dyhre Nielsen

Aalborg University

Course Organization

Teacher

Thomas Dyhre Nielsen, tdn@cs.aau.dk, office 1.2.34

Literature

D. Poole and A. Mackworth: *Artificial Intelligence. Foundations of Computational Agents* (2nd edition)

`http://artint.info/`

Course Homepage

Can be found under Moodle

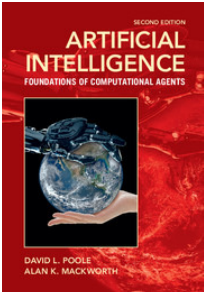
Times

- Wednesdays, 8.15-12.00: Exercises 8.15-10.00, Lecture 10.15-12.00
- Extended exercise sessions: Thursdays 12.30-16.15.

Exam

Exam in January

Why this book?



Several reasons, but:

> *We decided that it is better to clearly explain the foundations upon which more sophisticated techniques can be built, rather than present these more sophisticated techniques. This means that a larger gap may exist between what is covered in this book and the frontier of science. But it also means that the student will have a better foundation to understand current and future research.*

Tentative course overview

Topics:

- Introduction
- Problem solving as search
- Constrained satisfaction problems
- Logic-based knowledge representation
- Representing domains endowed with uncertainty.
- Bayesian networks
- Machine learning
- Planning
- Reinforcement learning
- Multi-agent systems

Learning goals

After having followed the course you should

- have knowledge about basic techniques within the field of machine intelligence and computational agents.
- be able to apply key machine intelligence techniques to a specific problem domain.
- be able to reason about computational agents that can operate in domains of varying complexity.

Learning goals

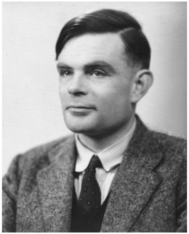
After having followed the course you should

- have knowledge about basic techniques within the field of machine intelligence and computational agents.
- be able to apply key machine intelligence techniques to a specific problem domain.
- be able to reason about computational agents that can operate in domains of varying complexity.

. . . and be well-prepared for the aMI course and the machine intelligence specialization!

AI

The Turing Test



A.M. Turing: “Computing Machinery and Intelligence”, Mind Vol.59 (1950). Proposes an *imitation game*, which (slightly modified) has become known as the *Turing test*:

An *interrogator* is connected via one terminal to a real person, and by another terminal to a computer (both in another room). The interrogator does not know which terminal is connected to the machine. He or she can perform on both terminals a (natural language) dialogue with whatever is at the other end of the line. The machine passes the Turing test, if the interrogator is not able to identify, which terminal is connected to the machine.

$\Rightarrow$ The Turing test tests observable behavior, not cognitive processes!

Achievements: Deep Blue

- 30 IBM RS/6000 processors
- 480 custom chess processors
- able to examine 200 million moves per second
- database of 700.000 grandmaster games
- endgame database (covering all 5 piece positions)

Results:

1996:	Kasparov 4	Deep Blue 2
1997:	Kasparov 2.5	Deep Blue 3.5

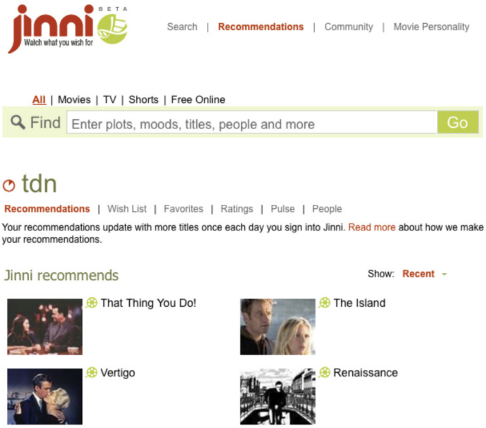
... two decades later

AlphaGo was developed by Google Deepmind to play Go. In March 2016 it payed a five game match against Lee Sedol with a final score of 4-1 in favor of AlphaGo.

- AlphaGo combines Monte Carlo tree search with (deep) artificial neural networks.

Recommender systems

Systems for recommending items that users are likely to find interesting



Recommmending items to users

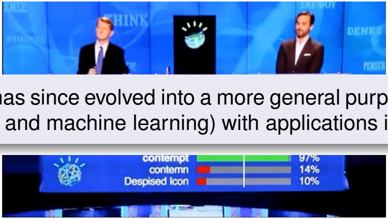
In 2011, Watson beat Brad Rutter, the biggest all-time money winner on Jeopardy!, and Ken Jennings, the record holder for the longest championship streak (75 days)

- Application of natural language processing, information retrieval, knowledge representation and reasoning, and machine learning
- Made up of a cluster of 90 IBM Power 750 servers with a total of 2880 POWER7 processor cores and 16 Terabytes of RAM. Each Power 750 server uses a 3.5 GHz POWER7 eight core processor, with four threads per core.

Taken from Wikipedia, August 25th, 2011

Jeopardy!: Watson

In 2011, Watson beat Brad Rutter, the biggest all-time money winner on Jeopardy!, and Ken Jennings, the record holder for the longest championship streak (75 days)

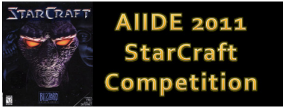


New developments Watson has since evolved into a more general purpose cognitive platform (using natural language processing and machine learning) with applications in e.g. health care.

- Application of natural language processing, information retrieval, knowledge representation and reasoning, and machine learning
- Made up of a cluster of 90 IBM Power 750 servers with a total of 2880 POWER7 processor cores and 16 Terabytes of RAM. Each Power 750 server uses a 3.5 GHz POWER7 eight core processor, with four threads per core.

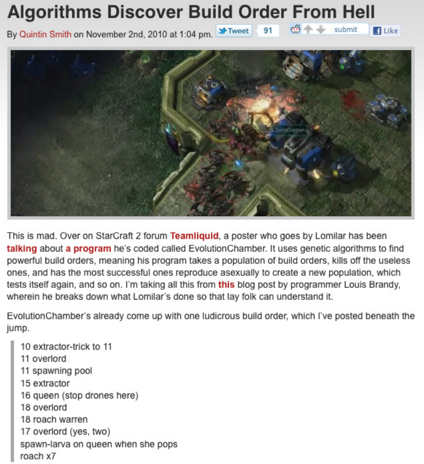
Taken from Wikipedia, August 25th, 2011

AI in computer games (e.g. Starcraft)



http://eis.ucsc.edu/StarCraftAICompetition

AI in computer games (e.g. Starcraft)



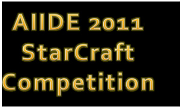
Algorithms Discover Build Order From Hell

By Quintin Smith on November 2nd, 2010 at 1:04 pm. Tweet 91 submit Like

This is mad. Over on StarCraft 2 forum **Teamliquid**, a poster who goes by Lomilar has been **talking** about **a program** he's coded called EvolutionChamber. It uses genetic algorithms to find powerful build orders, meaning his program takes a population of build orders, kills off the useless ones, and has the most successful ones reproduce asexually to create a new population, which tests itself again, and so on. I'm taking all this from **this** blog post by programmer Louis Brandy, wherein he breaks down what Lomilar's done so that lay folk can understand it.

EvolutionChamber's already come up with one ludicrous build order, which I've posted beneath the jump.

> 10 extractor-trick to 11
> 11 overlord
> 11 spawning pool
> 15 extractor
> 16 queen (stop drones here)
> 18 overlord
> 18 roach warren
> 17 overlord (yes, two)
> spawn-larva on queen when she pops
> roach x7



`http://eis.ucsc.edu/StarCraftAICompetition`

Computer games

The TrueSkill ranking system is a skill based ranking system for Xbox Live developed at Microsoft Research.

https://research.microsoft.com/en-us/projects/trueskill/

Troubleshooting

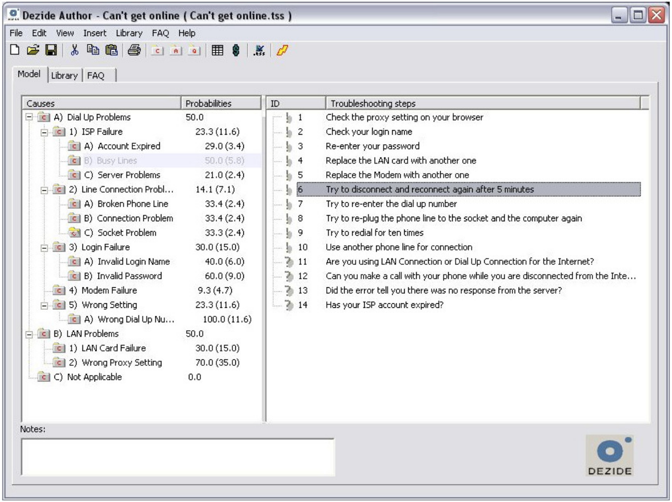
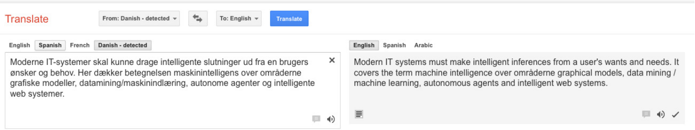


Figure borrowed from Dezide's homepage (`www.deczide.com`)

Achievements: Automatic Translation

Google translate:



Achievement: DARPA grand challenge 2005

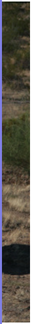
Competition for autonomous vehicles: navigate 132 miles through desert terrain (route specified by approx. 3000 “waypoints”). 5 out of 23 vehicles completed the task. Winner: *Stanley* of Stanford Racing Team in 6h 53m (19.2 mph).

- 7 Pentium M computers
- Sensors: 4 laser range finders, 1 radar system, 1 stereo camera pair, 1 monocular vision system, GPS, inertial measurement unit, wheel speed.

Achievement: DARPA grand challenge 2005

C ... rt terrain (route specified
b ... Winner: *Stanley* of
S

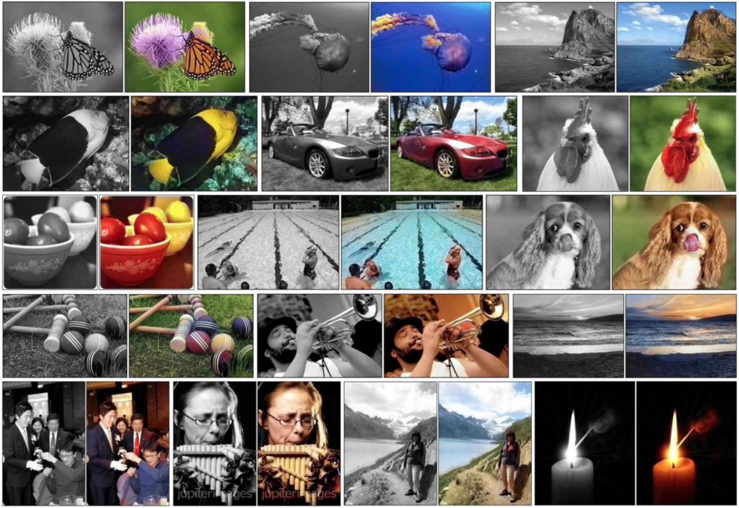
June 2016: approx. 2.8 mill. km. autonomous driving.



...ir, 1 monocular vision

Image analysis/processing

Colorization of images



Zhang, Isola, Efros. Colorful Image Colorization. In ECCV, 2016.

Image captioning

"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."

"boy is doing backflip on wakeboard."

"girl in pink dress is jumping in air."

"black and white dog jumps over bar."

"young girl in pink shirt is swinging on swing."

"man in blue wetsuit is surfing on wave."

Andrej Karpathy, Li Fei-Fei, Deep Visual-Semantic Alignments for Generating Image Descriptions, CVPR 2015

Other application areas

- Medical diagnosis and advisory systems
- Information processing and filtering,
- Display of information for time-critical decisions
- Spam filtering
- Optical character recognition
- Profiling/Credit scoring: profiling customers
- Bioinformatics
- Real estate: Prediction of house prices
- Computer networks: intrusion detection
- Alert and monitoring systems
- Speech recognition
- Face recognition, image annotation
- Action recognition (in video sequences)
- ...

Turing Test (Loebner Competition)

- Loebner Competition: (Non-scientific) competition for computer systems performing under Turing Test conditions.
- `http://www.pandorabots.com/pandora/talk?botid=f5d922d97e345aa1`

Online help

- Combine natural language interface with expert knowledge.
- Restricted Domain (Geography): CHAT-80 (1982)
- Broad Domain (“all factual knowledge”): Wolfram Alpha
  `http://www.wolframalpha.com/`

A "computational knowledge engine" ?

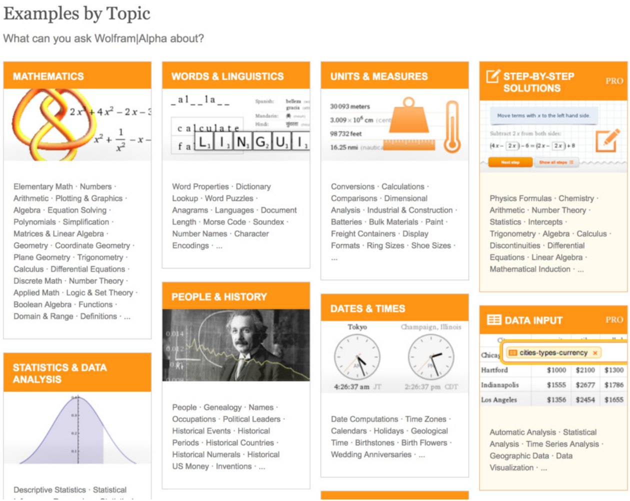
All one needs to be able to do is to take questions people ask in natural language, and represent them in a precise form that fits into the computations one can do.

[...]

I wasn't at all sure it was going to work. But I'm happy to say that with a mixture of many clever algorithms and heuristics, lots of linguistic discovery and linguistic curation, and what probably amount to some serious theoretical breakthroughs, we're actually managing to make it work. [Stephen Wolfram, `http://blog.wolfram.com`*, March 5, 2009]*

Think about that for a minute. It computes the answers. Wolfram Alpha doesn't simply contain huge amounts of manually entered pairs of questions and answers, nor does it search for answers in a database of facts. Instead, it understands and then computes answers to certain kinds of questions. [Nova Spivack on `twine.com`*]*

Some queries to try:

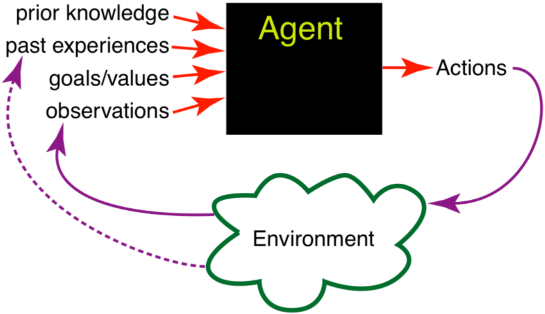


The Agent view of AI

AI and Agents

AI is the field that studies the synthesis and analysis of computational agents that act intelligently. [PM, p.3]

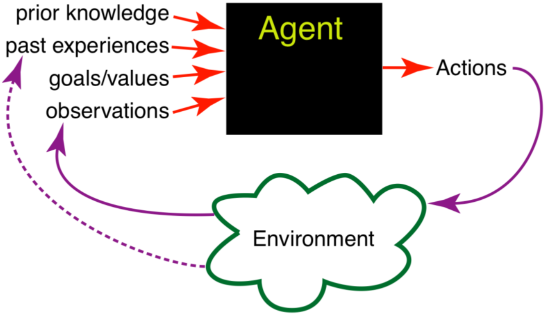
A coupling of perception, reasoning, and acting comprises an agent. [PM, p.10]



AI and Agents

AI is the field that studies the synthesis and analysis of computational agents that act intelligently. [PM, p.3]

A coupling of perception, reasoning, and acting comprises an agent. [PM, p.10]

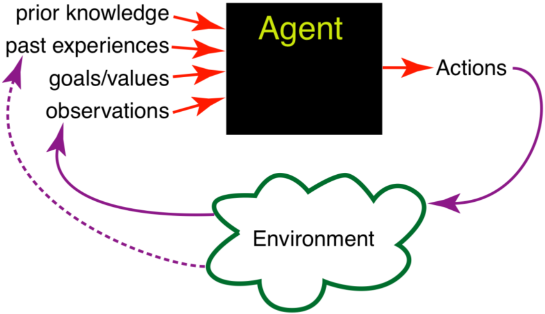


Some special flavors:

- Autonomous agents
- Intelligent agents
- Software agents
- Multi-agent systems

AI is the field that studies the synthesis and analysis of computational agents that act intelligently. [PM, p.3]

A coupling of perception, reasoning, and acting comprises an agent. [PM, p.10]

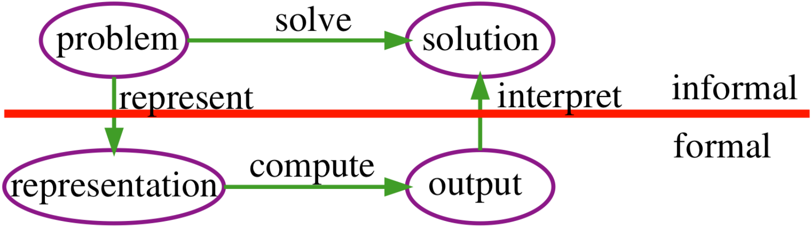


Some special flavors:

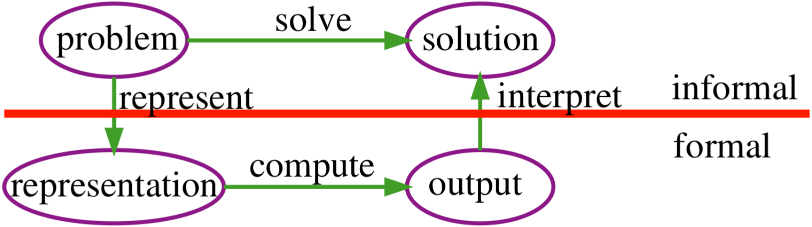
- Autonomous agents
- Intelligent agents
- Software agents
- Multi-agent systems

What is *not* an agent?

- perception $\sim$ input
- reasoning $\sim$ computation
- acting $\sim$ output
- $\rightsquigarrow$ "agent" a design metaphor, not a strict technical concept


problem
solve
solution
represent
interpret
informal
formal
representation
compute
output

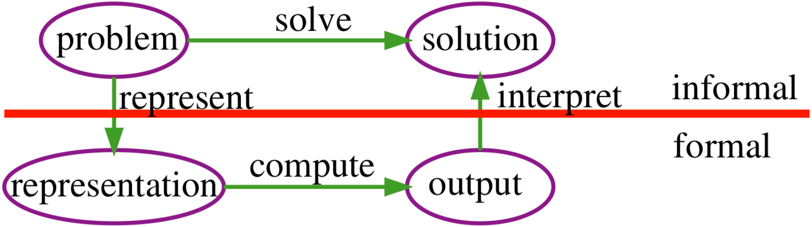
Represent and Compute



A *representation* should be:

- Sufficiently rich to encode the required knowledge.
- Be “close” to the problem.
- Amenable to efficient computation.
- Able to be acquired from people, data, or experience.

Represent and Compute



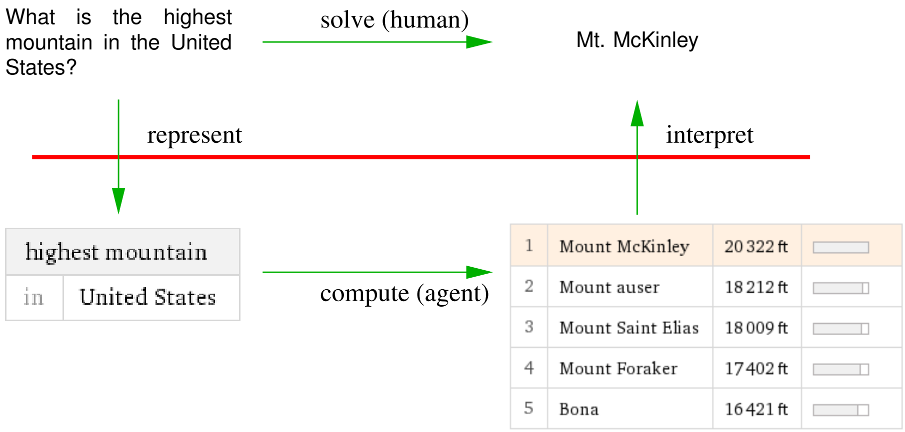
A *representation* should be:

- Sufficiently rich to encode the required knowledge.
- Be "close" to the problem.
- Amenable to efficient computation.
- Able to be acquired from people, data, or experience.

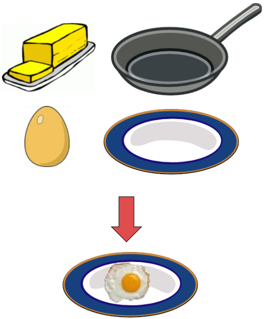
Questions to be considered:

- What is a solution and how good should it be (optimal, satisficing, approximately, probable)?
- How can the problem be represented?
- How can an output be computed (what properties should a solution have)?

Represent and Compute



1	Mount McKinley	20 322 ft	
2	Mount auser	18 212 ft	
3	Mount Saint Elias	18 009 ft	
4	Mount Foraker	17 402 ft	
5	Bona	16 421 ft	



solve (human)

- Heat Butter in Pan
- Break Egg into Pan
- Fry at moderate heat
- Put egg on plate

represent

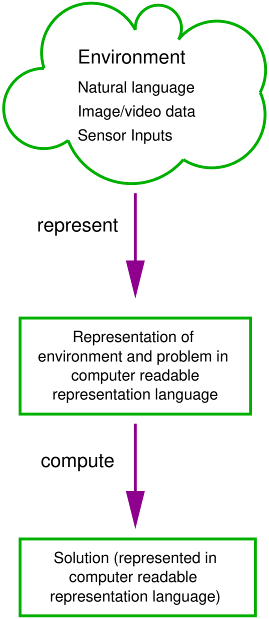
interpret

010010110000110101
011010001110011000
011010100010010011
101101000111101011

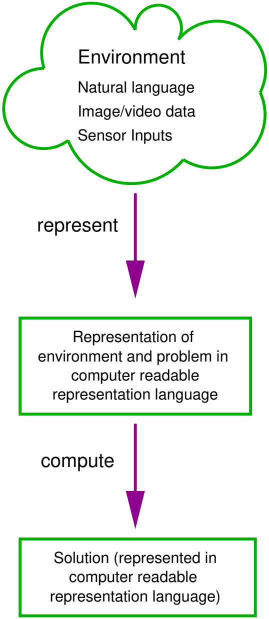
compute (agent)

011100010011101010011
001010010010001110010
110100100100001011100
111010010011101010011

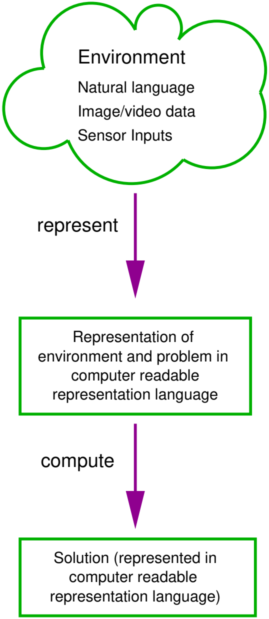
Problem Formalization



Problem Formalization



2. What formal representation languages can be used?

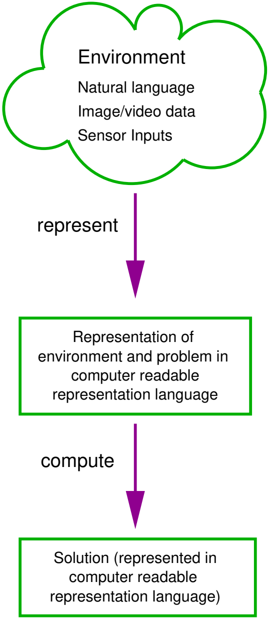


1. How to interpret input/observations from environment?

2. What formal representation languages can be used?

3. How to solve problems in the given representation language?

Problem Formalization



1. How to interpret input/observations from environment?

Natural Language Processing
Computer Vision
...

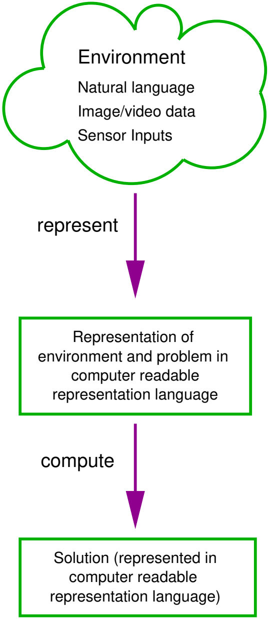
2. What formal representation languages can be used?

Knowledge Representation

3. How to solve problems in the given representation language?

Problem Solving
Automated Reasoning

Problem Formalization



1. How to interpret input/observations from environment? — *Natural Language Processing*, *Computer Vision*, ...

2. What formal representation languages can be used? — *Knowledge Representation*

3. How to solve problems in the given representation language? — *Problem Solving*, *Automated Reasoning*

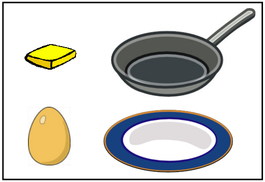
In this course the focus is on **2.** and **3.**!

Representing the problem

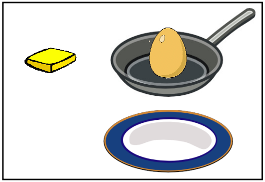
Levels of Representation

We consider 3 representation schemes

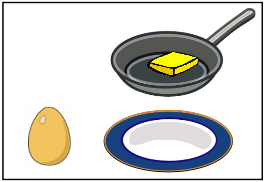
- State based
- Feature based
- Relational



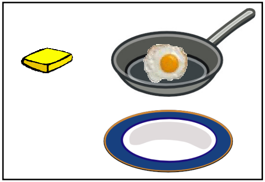
State 01



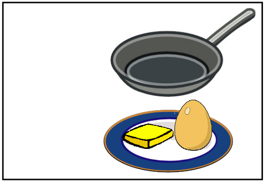
State 04



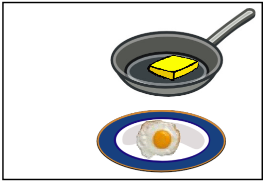
State 08



State 12

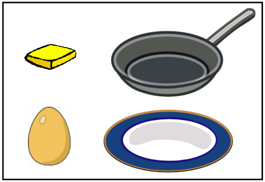


State 14

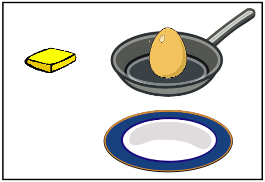


State 18

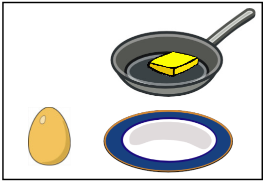
Feature based



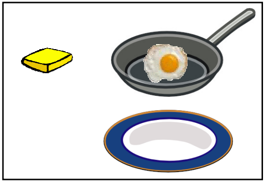
egg=whole,
butter_in=table,
egg_in=table



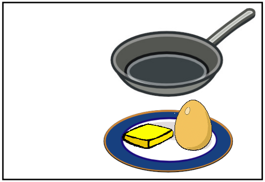
egg=whole,
butter_in=table,
egg_in=pan



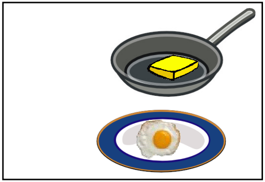
egg=whole,
butter_in=pan,
egg_in=table



egg=broken,
butter_in=table,
egg_in=pan



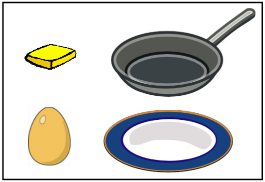
egg=whole,
butter_in=plate,
egg_in=plate



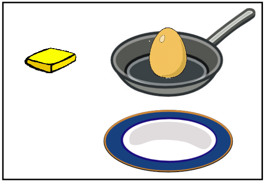
egg=broken,
butter_in=pan,
egg_in=plate

30 binary features represent $2^{30} = 1.073.741.824$ states.

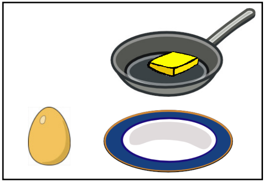
Relational



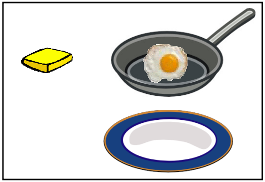
state(egg,whole),
in(butter,table),
in(egg,table)



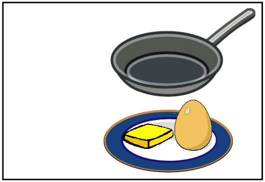
state(egg,whole),
in(butter,table),
in(egg,pan)



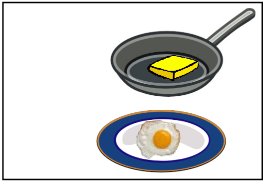
state(egg,whole),in(butter,pan),
in(egg,table)



state(egg,broken),
in(butter,table),
in(egg,pan)



state(egg,whole),
in(butter,plate),
in(egg,plate)

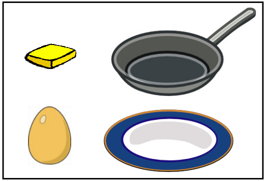


state(egg,broken),
in(butter,pan),
in(egg,plate)

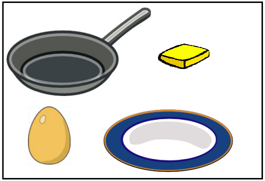
1 binary relation and 100 individuals give $100^2 = 10.000$ boolean features, or 2^{10000} states.

Levels of Detail

Do we need to distinguish states



egg=*whole*,
butter_in=*table*,
egg_in=*table*,
butter_position_to_pan=left



egg=*whole*,
butter_in=*table*,
egg_in=*pan*,
butter_position_to_pan=right

- Not at “recipe level”
- Yes at robot control level: “*move arm to left, grab butter, ...*”
- $\rightsquigarrow$ may need hierarchical description of state space to reason at different levels of abstraction.

Other dimensions of complexity

Modularity

- Flat, modular, hierarchical

Other dimensions of complexity

Modularity

- Flat, modular, hierarchical

Planning horizon

- Non-planning, finite, indefinite, infinite

Other dimensions of complexity

Modularity

- Flat, modular, hierarchical

Planning horizon

- Non-planning, finite, indefinite, infinite

Domain uncertainty

- Fully observable vs. partially observable world.
- Deterministic vs. stochastic.

Other dimensions of complexity

Modularity

- Flat, modular, hierarchical

Planning horizon

- Non-planning, finite, indefinite, infinite

Domain uncertainty

- Fully observable vs. partially observable world.
- Deterministic vs. stochastic.

Preferences

- Achievement goals, maintenance goals.
- Complex ordinal or cardinal preferences.

Other dimensions of complexity

Modularity

- Flat, modular, hierarchical

Planning horizon

- Non-planning, finite, indefinite, infinite

Domain uncertainty

- Fully observable vs. partially observable world.
- Deterministic vs. stochastic.

Preferences

- Achievement goals, maintenance goals.
- Complex ordinal or cardinal preferences.

Number of agents

- Single agent or multiple agents

Other dimensions of complexity

Modularity

- Flat, modular, hierarchical

Planning horizon

- Non-planning, finite, indefinite, infinite

Domain uncertainty

- Fully observable vs. partially observable world.
- Deterministic vs. stochastic.

Preferences

- Achievement goals, maintenance goals.
- Complex ordinal or cardinal preferences.

Number of agents

- Single agent or multiple agents

Learning

- Knowledge is given at design time or learned from experience.

Other dimensions of complexity

Modularity

- Flat, modular, hierarchical

Planning horizon

- Non-planning, finite, indefinite, infinite

Domain uncertainty

- Fully observable vs. partially observable world.
- Deterministic vs. stochastic.

Preferences

- Achievement goals, maintenance goals.
- Complex ordinal or cardinal preferences.

Number of agents

- Single agent or multiple agents

Learning

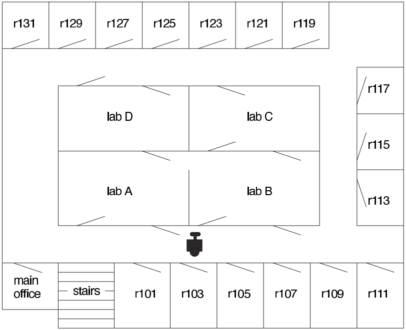
- Knowledge is given at design time or learned from experience.

Computational limits

- Perfect or bounded rationality

Prototypical Applications

Autonomous Delivery Robot



Inputs:

- Prior knowledge, past experience, goals, observations

Outputs:

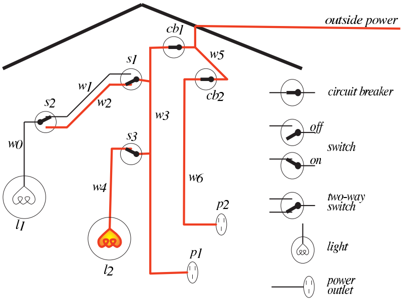
- Motor control, speech, video display

Complexity issues:

- Hierarchical decomposition
- Planning horizon
- Goals
- Uncertainty
- . . .

- Robot can move, pick up and put down objects
- Receives commands (in natural language)
- Can deliver packages, mail, coffee, . . .
- Must interpret commands, develop and execute plans for action

Diagnostic Assistant



Inputs:

- Prior knowledge, past experience, goals, observations

Outputs:

- Recommendations on treatments and tests

Complexity issues:

- Sensing/effect uncertainty
- Knowledge representation
- Goal specification

- Advise a human about system, e.g. diagnose patient, troubleshoot electrical system, automobile, etc.
- Substitute of human expert

Trading Agent

Characteristics

- Automatically buy/sell goods for user/company (possibly at auction)
- Determine good strategy to procure necessary goods in time at best price

Inputs

- Prior knowledge about goods, past experience, preferences, observations

Output

- Proposals to the user

Tentative course overview

Topics:

- Introduction
- Problem solving as search
- Constrained satisfaction problems
- Logic-based knowledge representation
- Representing domains endowed with uncertainty.
- Bayesian networks
- Machine learning
- Planning
- Reinforcement learning
- Multi-agent systems