

# Ch01. AI Introduction

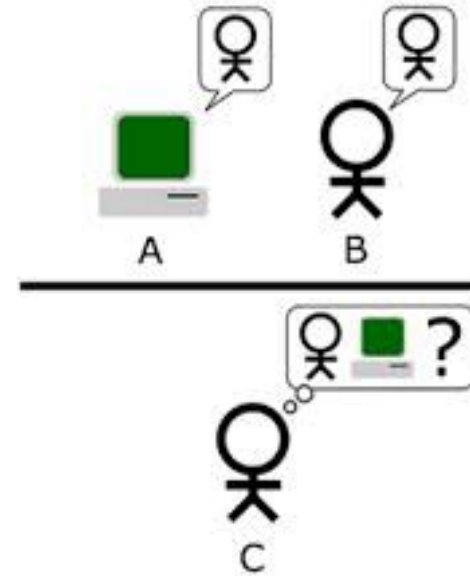
Hwanjo Yu

POSTECH

<http://hwanjoyu.org>

# The Turing Test, 1950

“Can machines think?”



Q: Please write me a sonnet on the subject of the Forth Bridge.

A: Count me out on this one. I never could write poetry.

Q: Add 34957 to 70764.

A: (Pause about 30 seconds and then give as answer) 105621.

*What can computer do better than Human?*

# Computer Vision

stone wall [ 0.95, [web](#) ]



dishwasher [ 0.91, [web](#) ]



car show [ 0.99, [web](#) ]



tractor [ 0.91, [web](#) ]



tractor [ 0.91, [web](#) ]



tractor [ 0.94, [web](#) ]



# Autonomous Driving

## Look – no driver

### Video camera

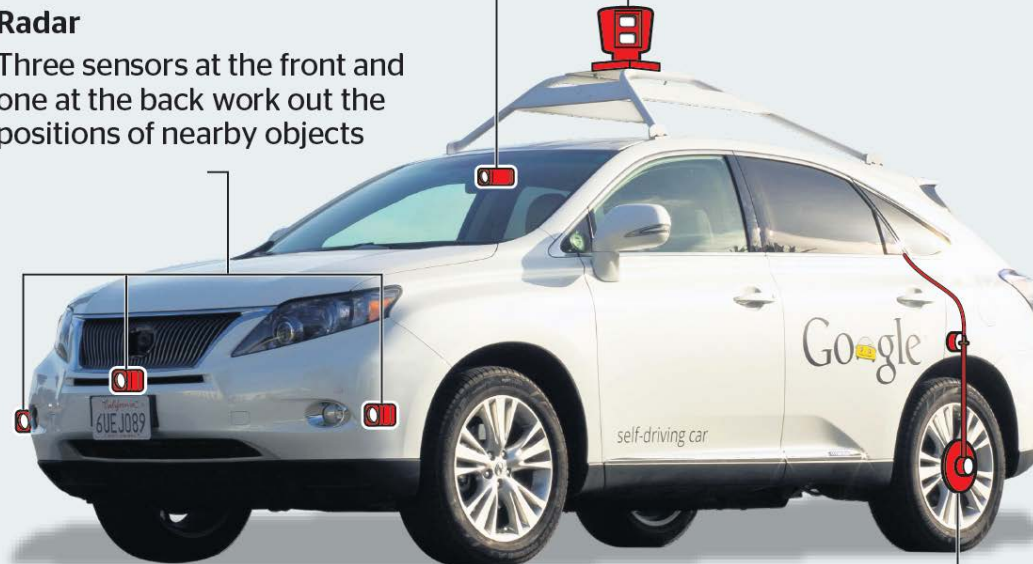
Detects traffic lights, oncoming vehicles and other obstacles

### Lidar

A rotating sensor on the roof scans 200ft in all directions to create a 3D map of its surroundings

### Radar

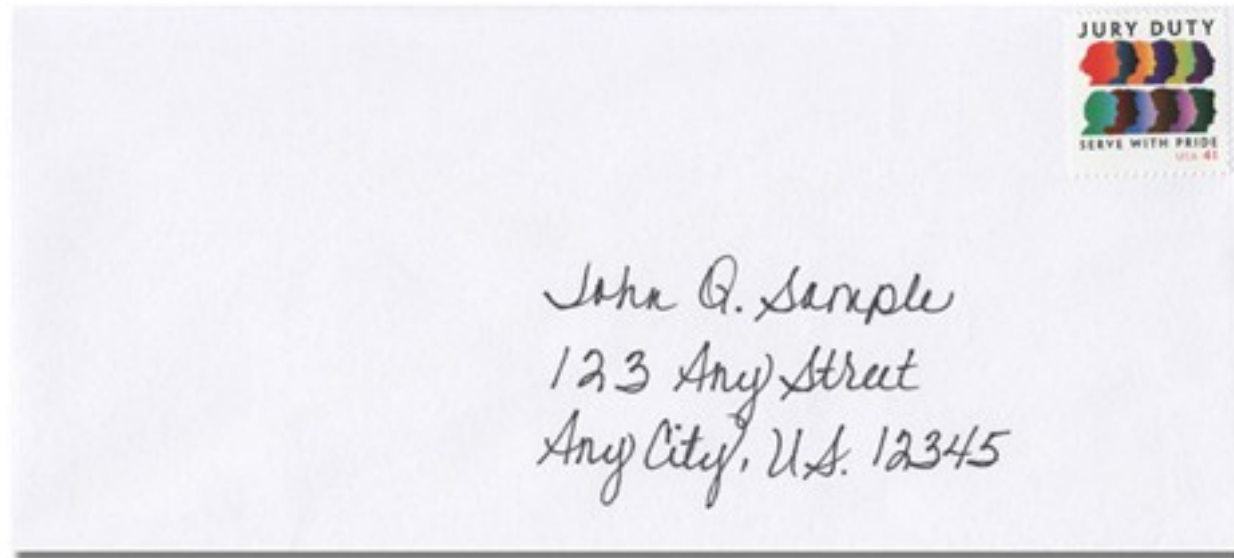
Three sensors at the front and one at the back work out the positions of nearby objects



### Position estimator

A sensor on the left rear wheel measures the car's movements so that its position can be mapped with accuracy

# Handwriting Recognition



# Sentiment Analysis

*“This movie should have NEVER been made. From the poorly done animation, to the beyond bad acting. I am not sure at what point the people behind this movie said "Ok, looks good! Lets do it!" I was in awe of how truly horrid this movie was.”*

**Positive or Negative ?**

# Machine Translations

The screenshot displays the Google Translate web interface. At the top, a navigation bar includes links to '+You', 'Search', 'Images', 'Maps', 'Play', 'YouTube', 'News', 'Gmail', 'Drive', 'Calendar', and 'More'. The Google logo is on the left, and a 'SIGN IN' button is on the right. Below the navigation bar, the 'Translate' section features a 'From: French - detected' dropdown, a swap button, a 'To: English' dropdown, and a 'Translate' button. The main content area is split into two columns. The left column, labeled 'French - detected', contains the French text: 'Le premier ministre a lancé une autre piste – sans l'expliquer et beaucoup des experts présents à la conférence environnementale n'ont pu le faire - : la mobilisation d'une partie des gains financiers perçus sur le parc nucléaire français. "Pendant toute la durée de vie restante de nos centrales, et tout en assurant une sécurité maximale, a déclaré Jean-Marc Ayrault, notre parc nucléaire sera mis à contribution sans rupture d'approvisionnement".' The right column, labeled 'English', contains the translated text: 'The Prime Minister has launched another track - without explaining and many experts at the environmental conference could not do -: the mobilization of some of the financial gains earned on the French nuclear fleet. "Throughout the remaining life of our plants, and while ensuring maximum security, said Jean-Marc Ayrault, our nuclear fleet will be involved without supply disruption."' Both columns have a star icon and a speech bubble icon at the bottom right. At the bottom of the interface, there are links for 'Turn off instant translation', 'About Google Translate', 'Mobile', 'Privacy', 'Help', and 'Send feedback'.

+You Search Images Maps Play YouTube News Gmail Drive Calendar More

Google

SIGN IN

Translate

From: French - detected To: English Translate

English Spanish French French - detected

Le premier ministre a lancé une autre piste – sans l'expliquer et beaucoup des experts présents à la conférence environnementale n'ont pu le faire - : la mobilisation d'une partie des gains financiers perçus sur le parc nucléaire français. "Pendant toute la durée de vie restante de nos centrales, et tout en assurant une sécurité maximale, a déclaré Jean-Marc Ayrault, notre parc nucléaire sera mis à contribution sans rupture d'approvisionnement".

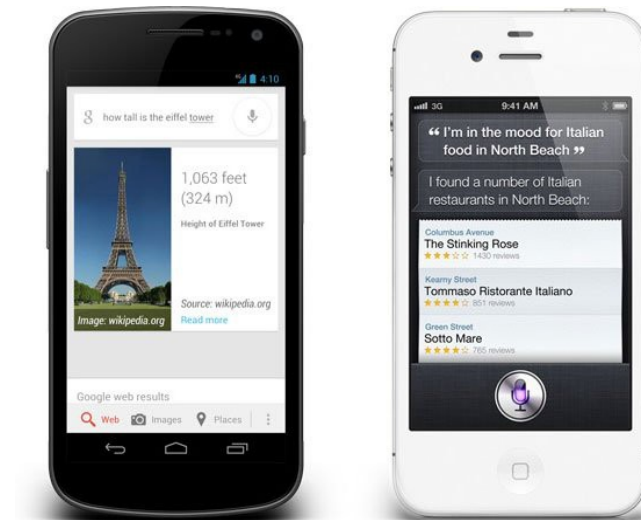
English Spanish Arabic

The Prime Minister has launched another track - without explaining and many experts at the environmental conference could not do -: the mobilization of some of the financial gains earned on the French nuclear fleet. "Throughout the remaining life of our plants, and while ensuring maximum security, said Jean-Marc Ayrault, our nuclear fleet will be involved without supply disruption."

Turn off instant translation About Google Translate Mobile Privacy Help Send feedback



# Virtual Assistants



# Dialog System

2016 ten breakthrough technology from MIT technology review

Chat bot

Tutor robot

...

# Humans versus Machines



1997: Deep Blue (chess)



2011: IBM Watson (Jeopardy!)



2016: AlphaGo

## Stephen Hawking warns artificial intelligence could end mankind

By Rory Cellan-Jones  
Technology correspondent

2 December 2014 | Technology



Stephen Hawking: "Humans, who are limited by slow biological evolution, couldn't compete and would be superseded"

BUSINESS

## The Dawn of the Age of Artificial Intelligence

Reasons to cheer the rise of the machines



### Elon Musk: AI Is Going to Happen. Let's Prepare For It

"It's definitely going to happen. So if it's going to happen, what's the best way for it to happen?"



BY TESS TOWNSEND Staff reporter, Inc.com @Tess\_Townsend



1.1k SHARES

2 COMMENTS



IMAGE: Getty Images

Elon Musk has emerged as a leading voice in speaking out on the potential dangers of artificial intelligence, going so far as to call it the "biggest existential threat" to

NICK BOSTROM

## SUPERINTELLIGENCE

Paths, Dangers, Strategies



### TODAY'S MUST READS

1 / Speaking Habits That Make You Sound, Like, Totally Unprofessional

How Playing the Long Game Made Elizabeth Holmes a Billionaire

5 Holy Knickknacks to Celebrate Pope Francis's Visit

Inside the Mind of Facebook's Sheryl Sandberg

Take a Video Tour of Facebook's Frank Gehry-Designed New York City Office

HIT THE ROAD

# AI Winter (AI 암흑기)

- 1956: Dartmouth workshop, John McCarthy coined "AI"
- 1960: checkers playing program, Logical Theorist
- 1966: ALPAC report cuts off funding for translation
- 1974: Lighthill report cuts off funding in UK
- 1970-80s: expert systems (XCON, MYCIN) in industry
- 1980s: Fifth-Generation Computer System (Japan); Strategic Computing Initiative (DARPA)
- 1987: collapse of Lisp market, government funding cut
- 1990-: rise of machine learning
- 2010s: heavy industry investment in deep learning
- ???

# Many AI Applications

...

Web search  
Speech recognition  
Handwriting recognition  
Machine translation  
Information extraction  
Document summarization  
Question answering  
Spelling correction  
Image recognition  
3D scene reconstruction  
Human activity recognition  
Autonomous driving  
Music information retrieval  
Automatic composition  
Social network analysis

...

...

Product recommendation  
Advertisement placement  
Smart-grid energy optimization  
Household robotics  
Robotic surgery  
Robot exploration  
Spam filtering  
Fraud detection  
Fault diagnostics  
AI for video games  
Financial trading  
Dynamic pricing  
Protein folding  
Medical diagnosis  
Medical imaging

...

# Characteristics of AI Tasks

**High societal impact** (affect billions of people)

**Diverse** (language, games, robotics)

**Complex** (really hard)

# Two sources of complexity

## Computational Complexity

- Most AI problems are NP-hard
- Go –  $361^{200}$  trajectories that a player would have to consider to play optimally.

## Information Complexity

- Translate a sentence
- Classify a bird from image



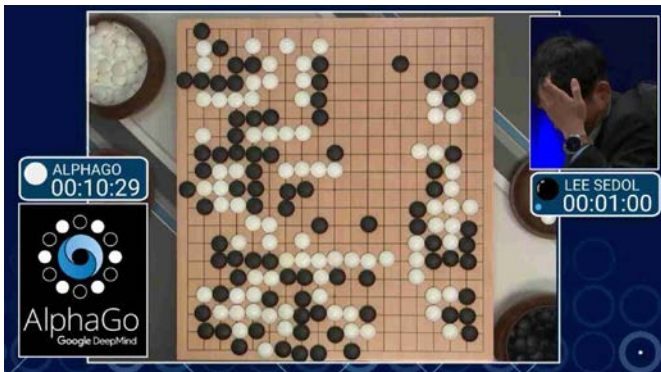
# Resources

Computation (time/memory)



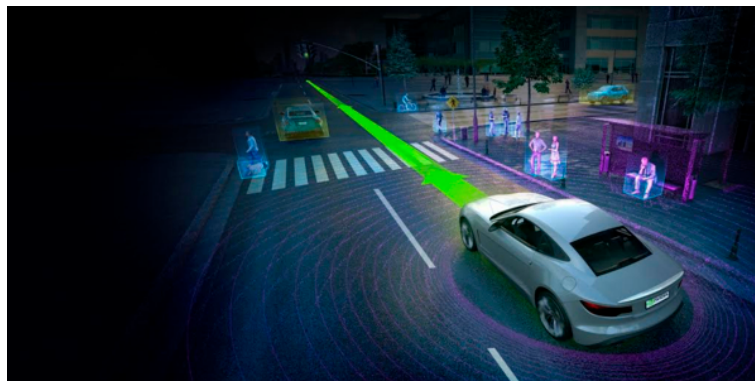
Information (data)





## AlphaGo (2016)

- 30 million training data
- Tensor Processing Unit
- Deep & Reinforcement Learning



## NVIDIA Self-Driving (2016)

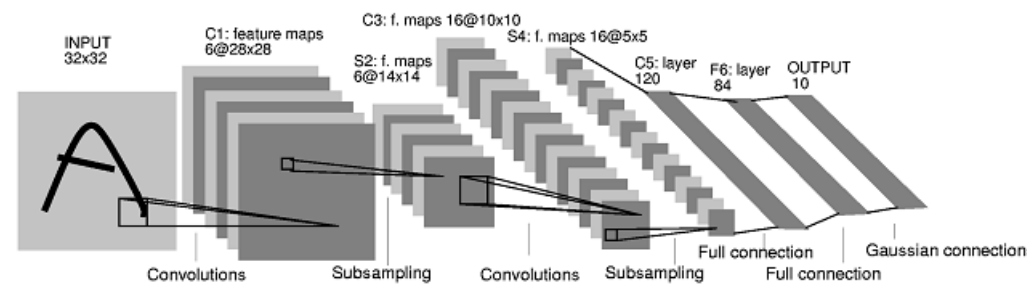
- Vision data by driving tens of thousands miles
- NVIDIA GPU, Deep Learning



## Google Translation

- Billions of translation data
- IBM's linguistic approach fails

# Big Data + Hardware + Machine Learning Algorithm



*How do we ~~solve~~ tackle these challenging AI tasks?*

Real-world task



```
Data structure for supporting uniform cost search.
class PriorityQueue:
    def __init__(self):
        self.DONE = 100000
        self.heap = []
        self.priorities = {} # Map from state to priority.

    # Insert (state) into the heap with priority (newPriority) if
    # (state) isn't in the heap or (newPriority) is smaller than the existing
    # priority.
    # Return whether the priority queue was updated.
    def update(self, state, newPriority):
        oldPriority = self.priorities.get(state)
        if oldPriority is None or newPriority < oldPriority:
            self.priorities[state] = newPriority
            heapq.heappush(self.heap, (newPriority, state))
            return True
        return False

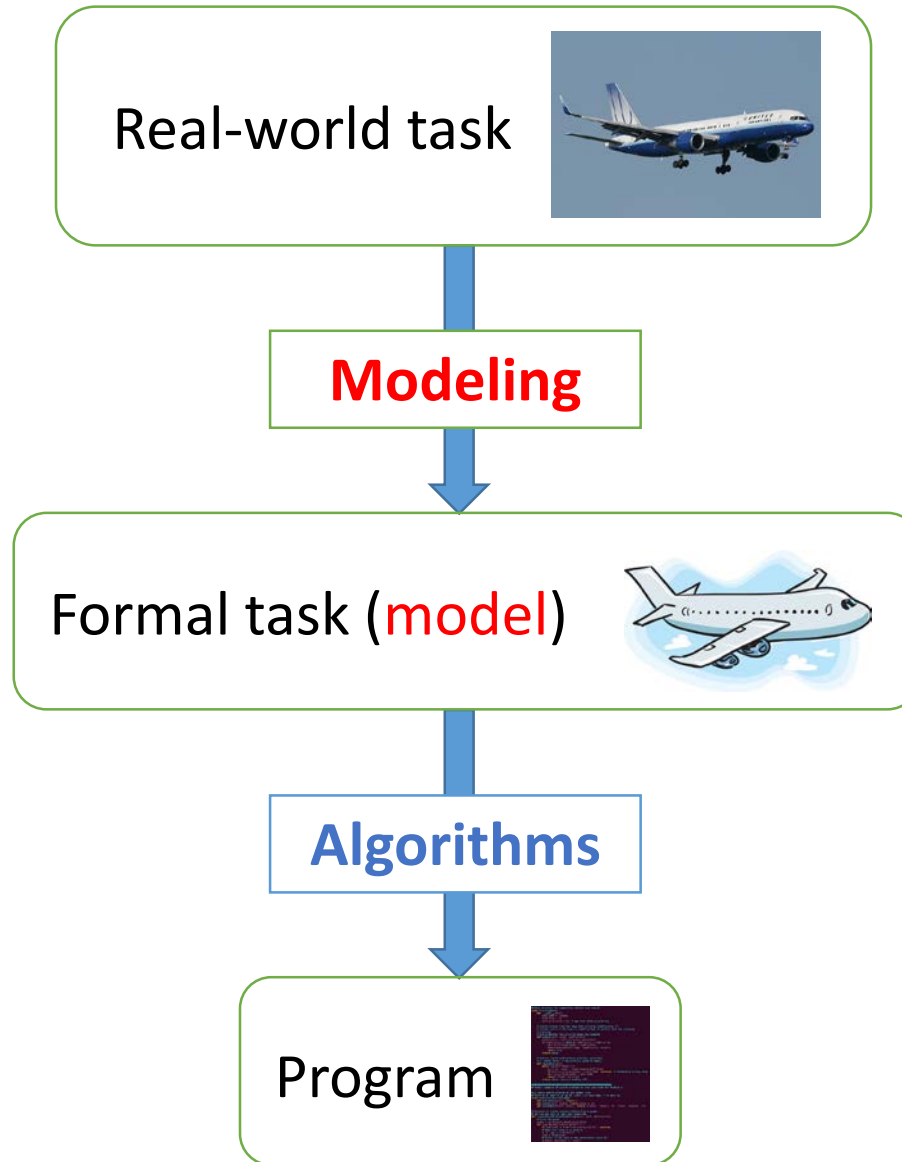
    # Return (state with minimum priority, priority)
    # or (None, None) if the priority queue is empty.
    def removeMin(self):
        while len(self.heap) > 0:
            priority, state = heapq.heappop(self.heap)
            if self.priorities[state] == self.DONE: continue # Outdated priority, skip
            self.priorities[state] = self.DONE
            return (state, priority)
        return (None, None) # Nothing left...

=====
# Simple examples of search problems to test your code for Problem 1.

# A simple search problem on the number line.
# Start at 0, want to go to 10, costs 1 to move down, 2 to move up.
class NumberLineSearchProblem:
    def startState(self): return 0
    def isGoal(self, state): return state == 10
    def actions(self, state): return [(('down', state-1, 1), ('up', state+1, 2))]

# Function to create search problems from a graph.
# You can use this to test your algorithm.
def createSearchProblemFromGraph(start, goal, description):
    # Parse the graph
    g = collections.defaultdict(list)
    for line in description.split("\n"):
        if len(line) == 0 or line.startswith('#'): continue
        # Split from state a to state b
        a, b, cost = line.split(" ")
        cost = float(cost)
        # Action is the same as the destination state (b).
        g[a].append((b, cost))

=====
```



# Algorithms (example)

- Formal task:
  - Input: list  $L = \{x_1, \dots, x_n\}$  and a function  $f : X \rightarrow \mathbb{R}$
  - Output:  $k$  highest-scoring elements
- Example ( $k = 2$ ):
  - $L : \quad A \quad B \quad C \quad D$
  - $f : \quad 3 \quad 2 \quad 7 \quad 1$
- Two algorithms:
  1. Scan through to find the largest, scan through again to find the second largest, etc.
  2. Sort  $L$  based on  $f$ , return first  $k$  elements

# Modeling (example)

- Real-world task:
  - Input: 20 billion web pages, a keyword query
  - Output: 10 most relevant web pages
- Modeling:
  - $L$  = list of web pages
  - $f(x) = 10 * \text{QueryMatch}(x) + 3 * \text{PageRank}(x)$
- Formal task:
  - Input: list  $L = \{x_1, \dots, x_n\}$  and a function  $f : X \rightarrow \mathbb{R}$
  - Output:  $k$  highest-scoring elements

# Modeling and algorithms

- Separate **what** to compute (**modeling**) from **how** to compute it (**algorithms**) => Advantage: division of labor
- What do we learn?
  - Type of models...
  - Art of modeling...
  - Developing Algorithms...



# Summary so far

- Applications of AI: high-impact, diverse
- Challenges: computational/information complexity
- Paradigm: modeling + algorithms

## Reflex

"Low-level intelligence"

"High-level intelligence"

# Sentiment Analysis

- Input: movie review

*“Shows moments of promise but ultimately succumbs to cliches and pat storytelling.”*

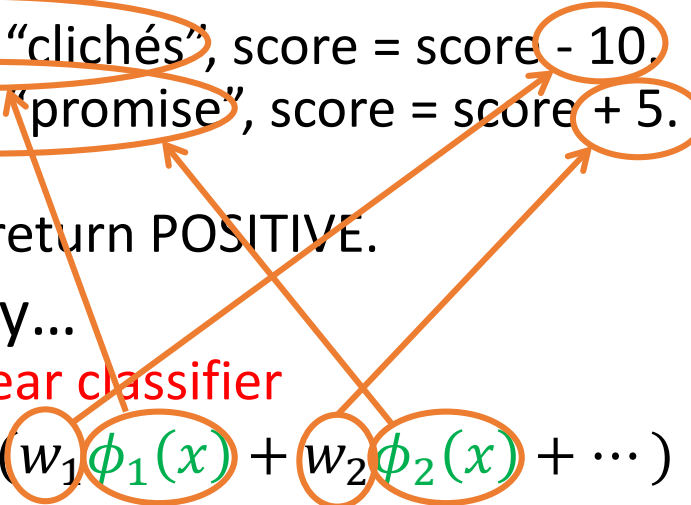
- Output: sentiment

*POSITIVE or NEGATIVE*

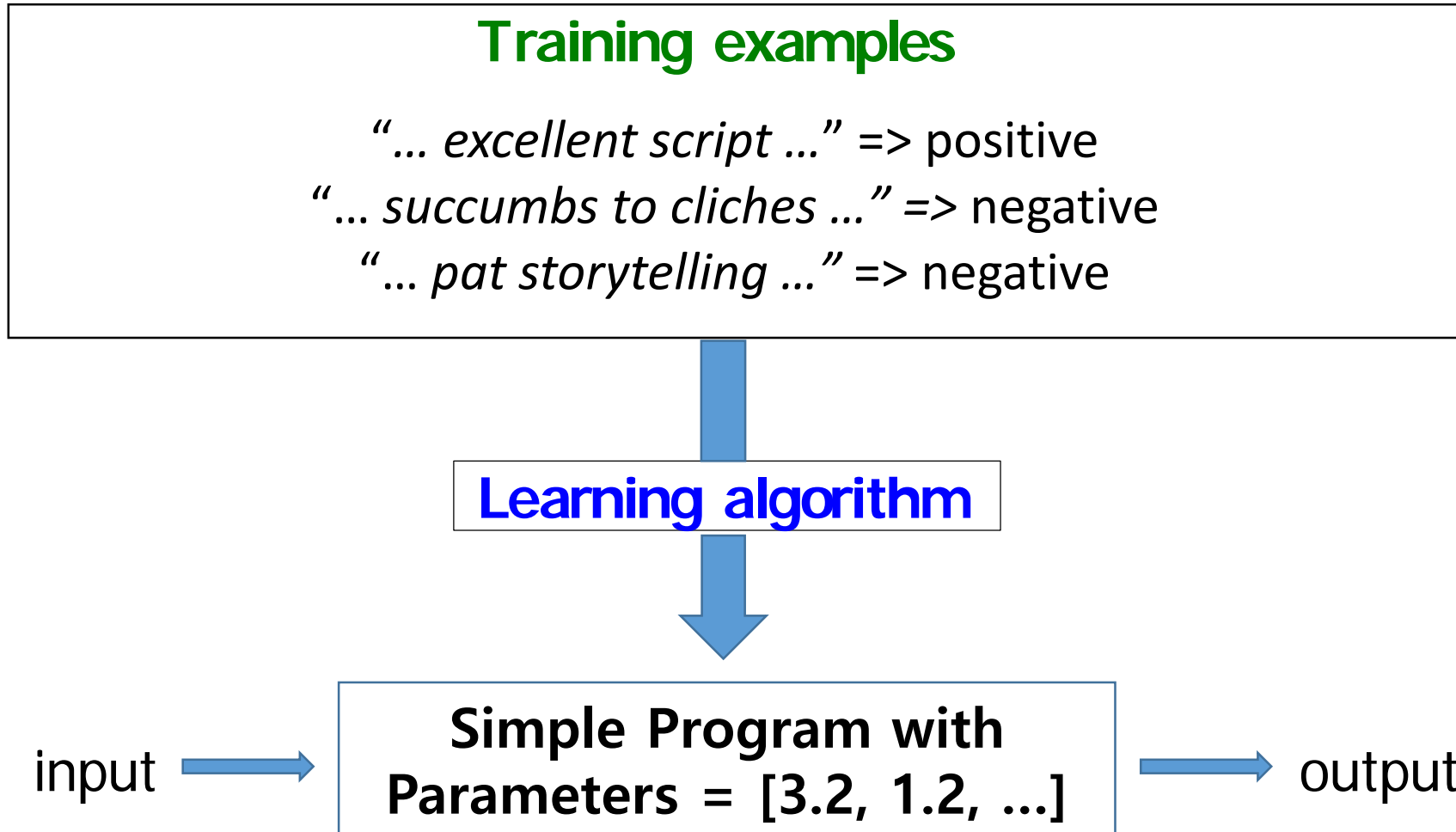
# Reflex-based Models

- Input:  $x$ , a document or sentences
- Output:  $f(x)$ , a simple function of  $x$
- Example: **model  $f$  is a set of simple rules**
  - If  $x$  contains “clichés”, return NEGATIVE.
  - If  $x$  contains “promise”, return POSITIVE.

# Reflex-based Models: Linear model

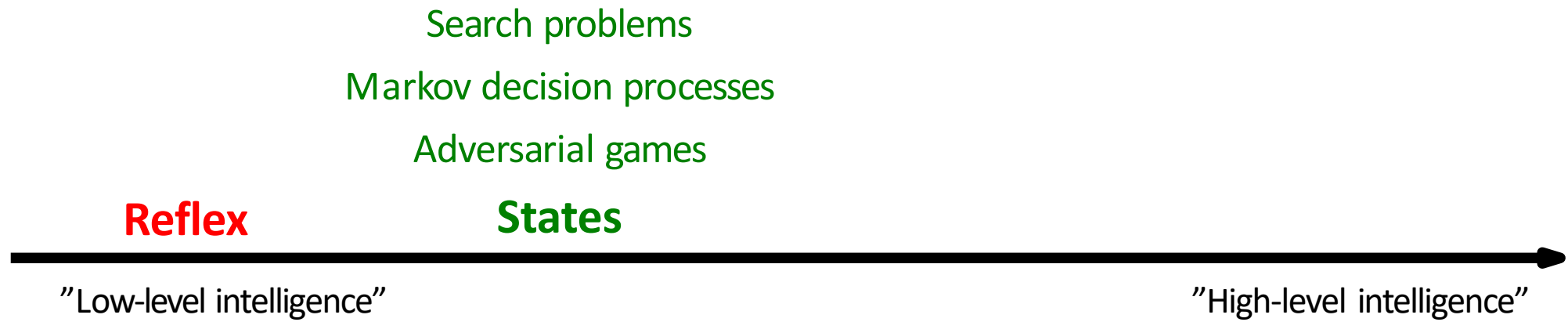
- Use scores to capture nuances...
  - Output  $f$  is determined based on scores
    - Set score = 0
    - If  $x$  contains “clichés”, score = score - 10.
    - If  $x$  contains “promise”, score = score + 5.
    - ...
    - If score > 0, return POSITIVE.
  - More generally...
    - Key idea: linear classifier
    - $f(x) = \text{sign}(w_1\phi_1(x) + w_2\phi_2(x) + \dots)$
    - $f(x) = \sigma(\mathbf{w} \cdot \phi(x))$
  - How about “not bad”?
- 
- The diagram consists of several orange circles and arrows. One circle is around the word 'promise' in the third bullet point. Another circle is around the symbol  $\phi_1(x)$  in the general formula. A third circle is around the symbol  $\phi_2(x)$  in the same formula. An arrow points from the 'promise' circle to the  $\phi_1(x)$  circle. Another arrow points from the  $\phi_2(x)$  circle to the 'promise' circle. There are also arrows pointing from the 'promise' circle to the 'score = score + 5' part of the second bullet point, and from the  $\phi_1(x)$  circle to the 'score = score - 10' part of the second bullet point.

# Machine learning approach



# Machine Learning

- **Key idea: generalization**
  - Learning algorithm maximizes accuracy on **training** examples.
  - But we only care about accuracy on future **test** examples.
  - How to **generalize** from training to test?





# Text Reconstruction

- Chinese is written without spaces:

是 什 4 意思

- Arabic omits (some) vowels:

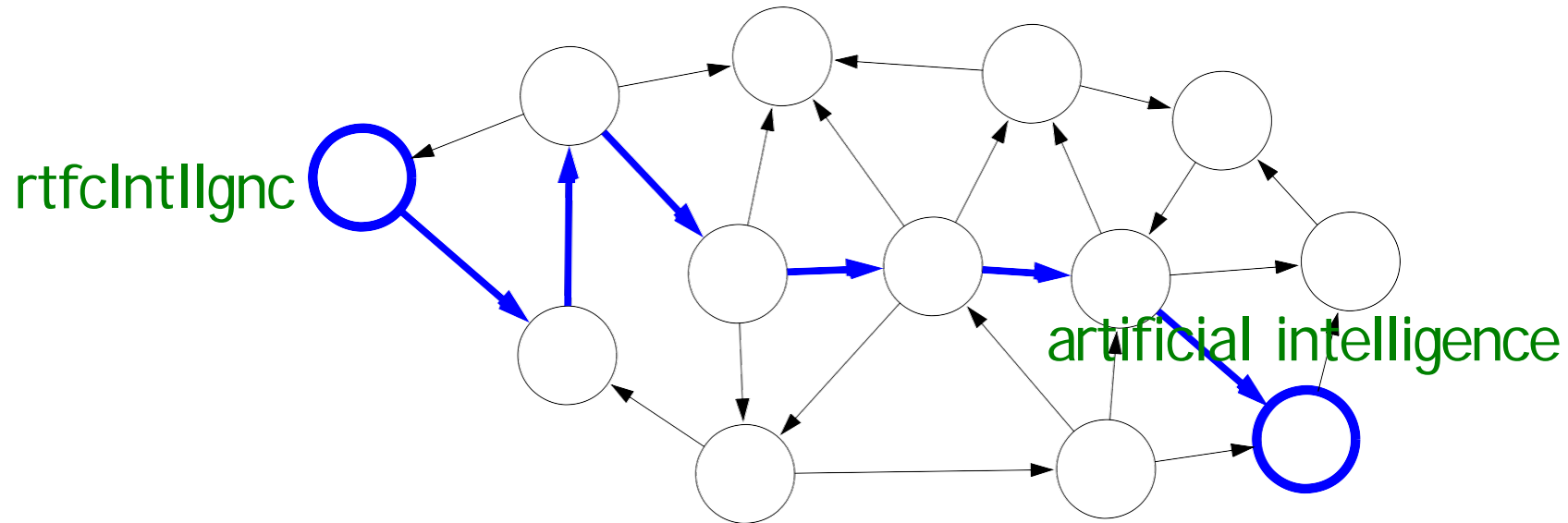
مَكْتَبَة

- Remove vowels and spaces from an English phrase:

rtfclntllgnc

# State-based Models

- Solutions are represented as paths through a graph



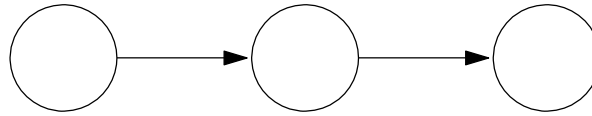
# State-based Models

## Key idea: state

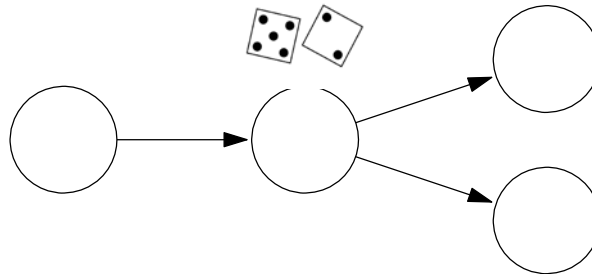
- A **state** captures all the relevant information about the past in order to act optimally in the future

# State-based Models

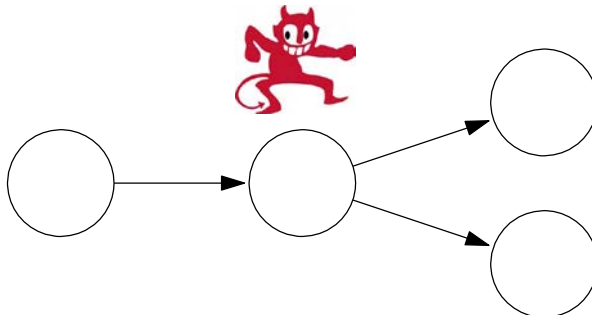
- Search problems: you control everything



- Markov decision processes: against nature (e.g., Blackjack)



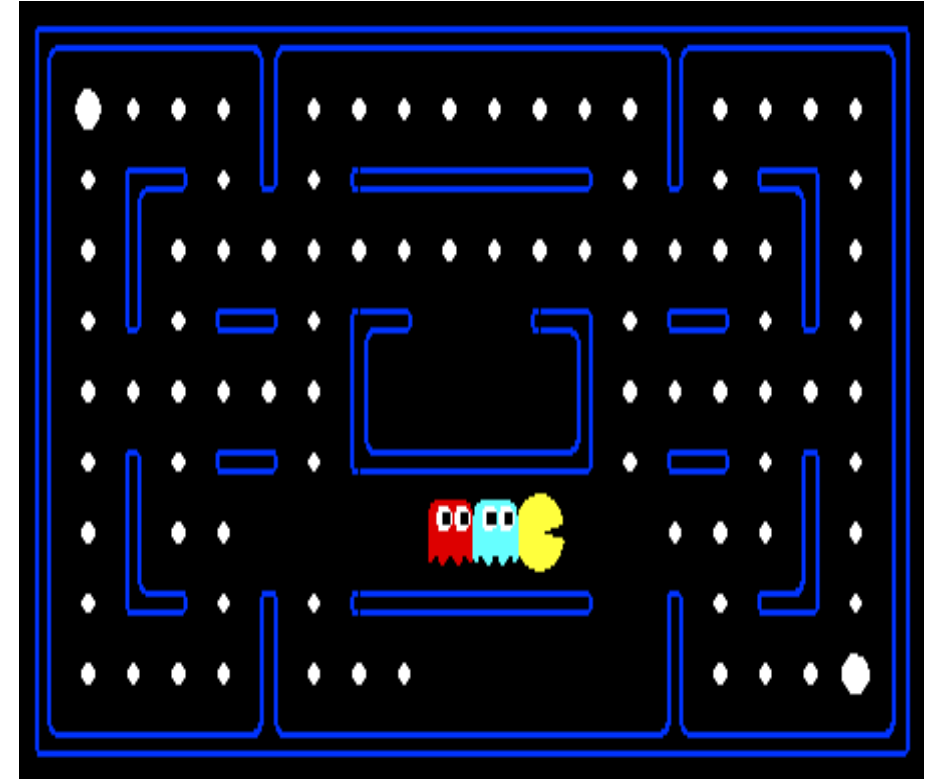
- Adversarial games: against opponent (e.g., chess)



# Pac-Man

What kind of model is appropriate for playing Pac-Man against ghosts that move into each valid adjacent square with equal probability?

1. Search problem
2. Markov decision process
3. Adversarial game

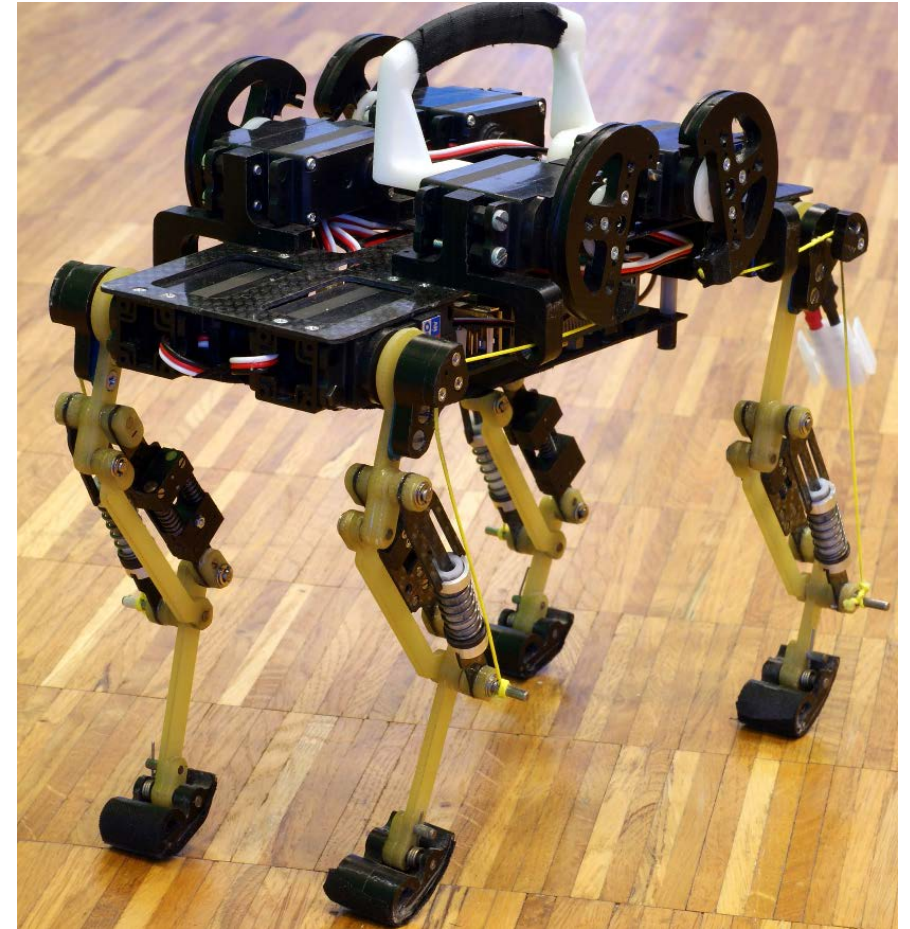


# Crawling robot

Goal: maximize distance travelled by robot

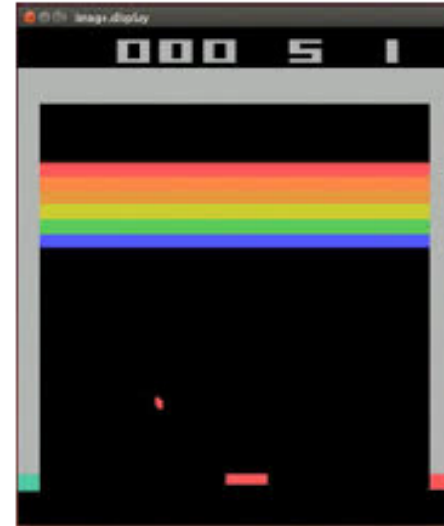
Markov decision process (MDP):

- States: positions (4 possibilities) for each of 2 servos
- Actions: choose a servo, move it up/down
- Transitions: move into new position (unknown)
- Rewards: distance travelled (unknown)



# Deep reinforcement learning

Playing Atari [Google DeepMind, 2013]:



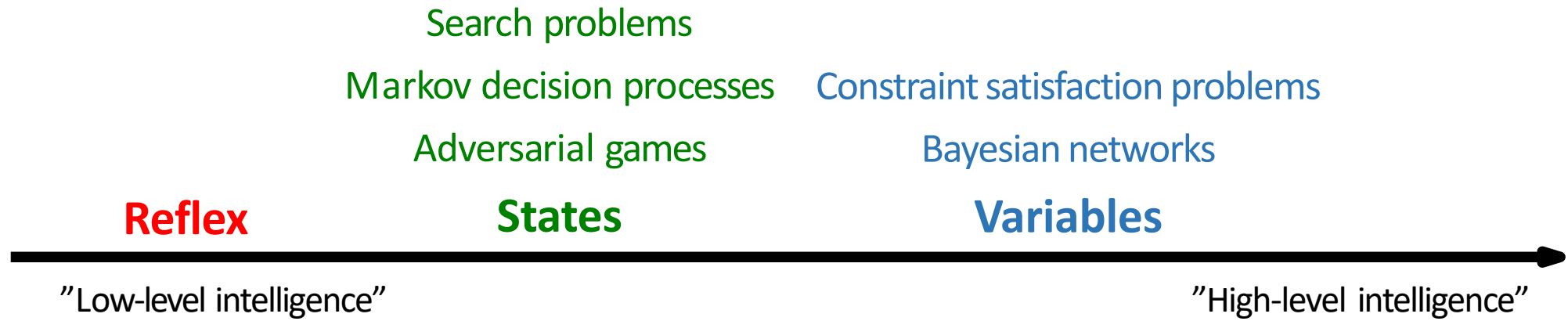
- Just use a neural network for  $\hat{Q}_{\text{opt}}(s, a)$
- Last 4 frames (images)  $\Rightarrow$  3-layer NN  $\Rightarrow$  keystroke
- $\epsilon$ -greedy, train over 10M frames with 1M replay memory
- <https://www.youtube.com/watch?v=V1eYniJ0Rnk>

# AlphaGo



- Supervised learning: on human games
- Reinforcement learning: on self-play games
- Evaluation function: convolutional neural network (value network)
- Policy: convolutional neural network (policy network)
- Monte Carlo Tree Search: search / lookahead





# Constraint Satisfaction Problem (CSP)



**Question:** how can we color each of the 7 provinces {red, green, blue} so that no two neighboring provinces have the same color?

# Map coloring



(one possible solution)

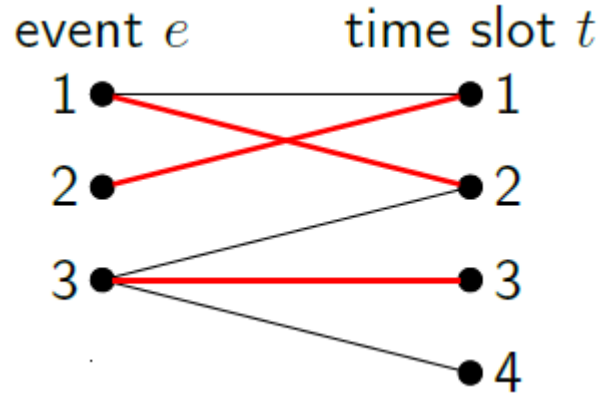
# Constraint Satisfaction Problem (CSP)

Three sculptures (A, B, C) are to be exhibited in rooms 1, 2 of an art gallery.

The exhibition must satisfy the following conditions:

- Sculptures A and B cannot be in the same room.
- Sculptures B and C must be in the same room.
- Room 2 can only hold one sculpture.

# Event scheduling

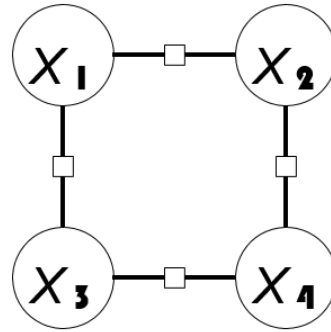


Setup:

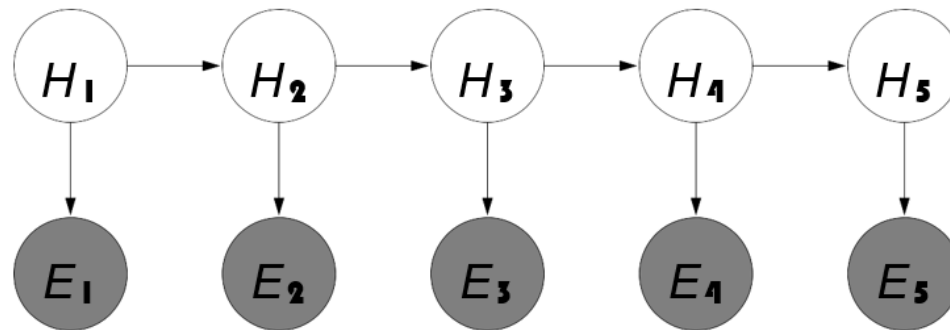
- Have  $E$  events and  $T$  time slots
- Each event  $e$  must be put in **exactly one** time slot
- Each time slot  $t$  can have **at most one** event
- Event  $e$  allowed in time slot  $t$  only if  $(e, t) \in A$

# Variable-based Models

- **Constraint satisfaction problem**: hard constraints (e.g., map coloring, scheduling)



- **Bayesian networks**: soft dependencies (e.g., tracking cars from sensors)



# Topic modeling

**Question:** given a text document, what topics is it about?

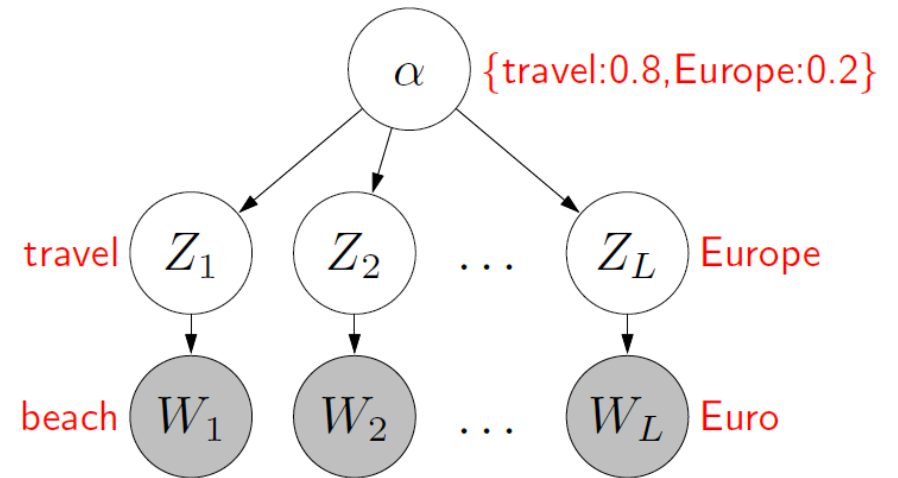
**Probabilistic program: latent Dirichlet allocation**

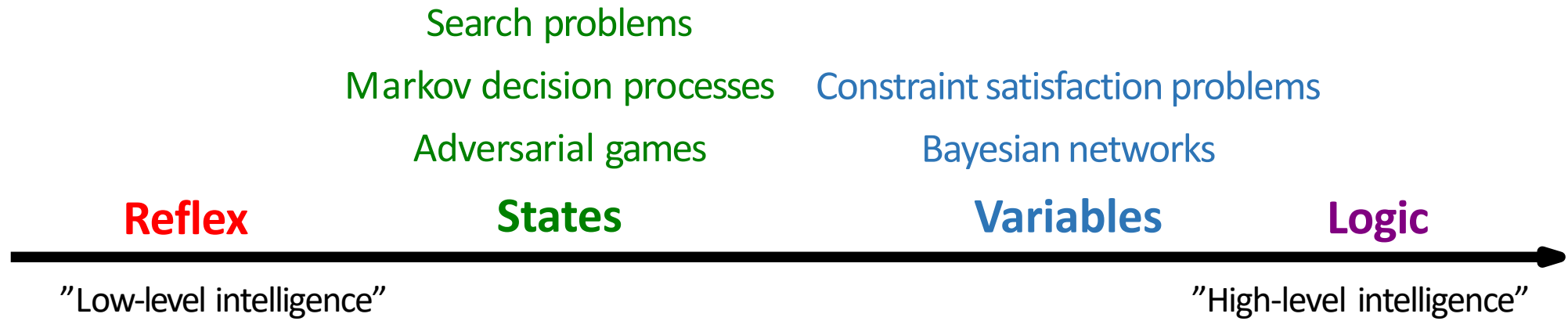
Generate a distribution over topics  $\alpha \in \mathbb{R}^K$

For each position  $i = 1, \dots, L$ :

Generate a topic  $Z_i \sim p(Z_i|\alpha)$

Generate word  $W_i \sim p(W_i|Z_i)$







# Question

You get extra credit if you write a paper and you solve the problems.

You didn't get extra credit, but you did solve the problems.

Did you write a paper?

Yes or No

# Knowledge representation and reasoning

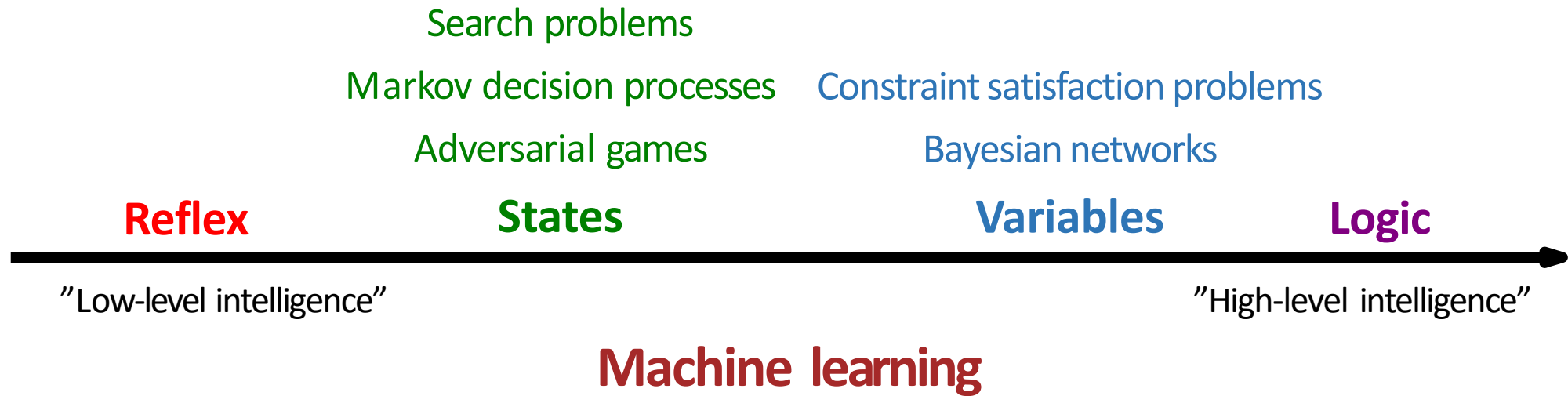
All students work hard.

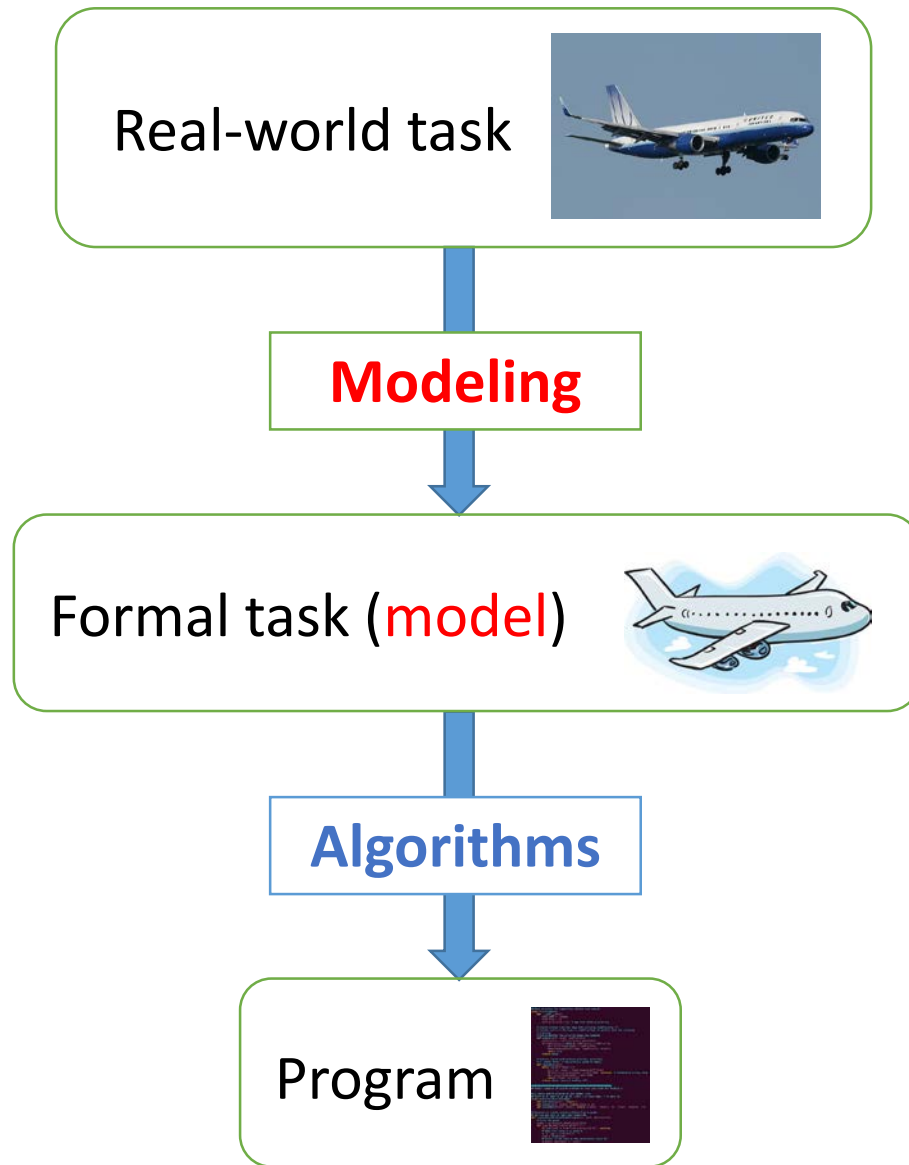
John is a student.

Therefore, John works hard.

Variable-based models would explicitly represent all the students – this is inefficient

Need **expressive power** of logic to represent this ...





Modeling is often expressed as **optimization problems**, which provides a mathematical specification of what we want to compute

# Problem: predicting exam score $y$

- $x$ : # of hours studying
- $y$ : exam score
- Assume  $y = \mathbf{w}x$
- $\mathbf{w}$  is a **learning parameter** we need to estimate (learn) from training data
- $\mathbf{D}_{\text{train}}$ : set of pairs  $\{(x_1, y_1), \dots, (x_n, y_n)\}$
- $\mathbf{f}$ :  $\mathbf{w} \in \mathbb{R}$  that minimizes the squared error  $F(\mathbf{w}) = \sum_{i=1}^n (x_i \mathbf{w} - y_i)^2$
- Example:

$$\{(2,4)\} \Rightarrow 2$$

$$\{(2,4),(4,2)\} \Rightarrow ?$$

# Optimization

- Models are optimization problems:

$$\min_{x \in C} F(x)$$

- Discrete optimization:  $x$  is a discrete object

$$\min_{x \in \{abcd, xyz\}} Length(x)$$

Algorithmic tool: dynamic programming

- Continuous optimization:  $x$  is a vector of real numbers

$$\min_{x \in \mathbb{R}} (x - 5)^2$$

Algorithmic tool: gradient descent

# Summary

- Applications of AI: high-impact, diverse
- Challenges: computational/information complexity
- Paradigm: modeling + algorithms
- Models: learning + [reflex, states, variables, logic]