IT5005 Artificial Intelligence

1. Introduction

Sirigina Rajendra Prasad
August-2022

Agenda

1. Getting Started

2. Agent Architecture

3. Moving Forward

1. Getting Started

- Intelligent Agents
  - Definitions
  - Objective
- Evolution of AI Systems and Taxonomy
  - Logic-based AI
  - Expert Systems
  - Machine Learning
  - Deep Learning
- Recent Success Stories
- Way Forward

Meaning of Artificial Intelligence

What does Dictionary.com say?

artificial [ahr-t*uh*-**fish**-*uh*l] SHOW IPA

SEE SYNONYMS FOR *artificial* ON THESAURUS.COM

adjective

1 made by human skill; produced by humans (opposed to natural):
artificial flowers.

2 imitation; simulated; sham:
artificial vanilla flavoring.

3 lacking naturalness or spontaneity; forced; contrived; feigned:
an artificial smile.

4 full of affectation; affected; stilted:
artificial manners; artificial speech.

5 made without regard to the particular needs of a situation, person, etc.; imposed arbitrarily; unnatural:
artificial rules for dormitory residents.

6 *Biology*. based on arbitrary, superficial characteristics rather than natural, organic relationships:
an artificial system of classification.

Meaning of Artificial Intelligence

What does Dictionary.com say?

intelligence [in-**tel**-i-j*uh*ns] SHOW IPA

SEE SYNONYMS FOR *intelligence* ON THESAURUS.COM

noun

1 capacity for learning, reasoning, understanding, and similar forms of mental activity; aptitude in grasping truths, relationships, facts, meanings, etc.

2 manifestation of a high mental capacity:
He writes with intelligence and wit.

3 the faculty of understanding.

4 knowledge of an event, circumstance, etc., received or imparted; news; information.

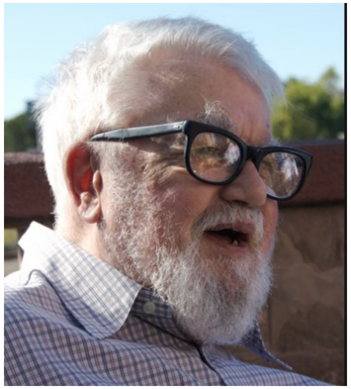
5 the gathering or distribution of information, especially secret information.

6 *Government.*
- a information about an enemy or a potential enemy.
- b the evaluated conclusions drawn from such information.
- c an organization or agency engaged in gathering such information:
military intelligence; naval intelligence.

Many Different Intelligences

- Emotional Intelligence
- Social Intelligence
- Perception Intelligence
- Manipulation Intelligence
- Abstraction Intelligence
- Learning Intelligence
- Reasoning Intelligence

Definition of AI



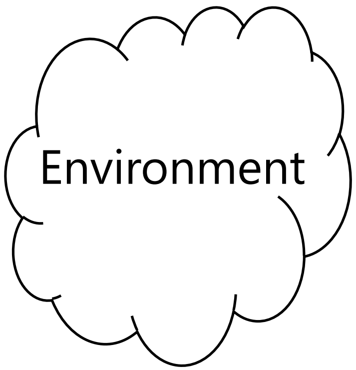
John McCarthy (1927 – 2011)

John McCarthy

“It is the science and engineering of making intelligent machines, especially intelligent computer programs.”

http://www-formal.stanford.edu/jmc/whatisai.pdf
Stonier T. (1992) The Evolution of Machine Intelligence. In: Beyond Information. Springer, London. https://doi.org/10.1007/978-1-4471-1835-0_6

Intelligent Agents



Perception

Sensory organs:
- eyes
- ears
- nose
- tongue
- skin

Action

Actions
- Moving
- Talking
- Eating
- Planning
- Making decisions

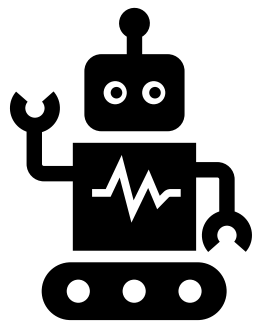
Image: https://scitechdaily.com/researchers-discover-how-the-human-brain-separates-stores-and-retrieves-memories/

Artificial Intelligent Agents

Environment

Perception

Sensors:
- mic
- cameras
- acc/gyro
- sonar
- touch



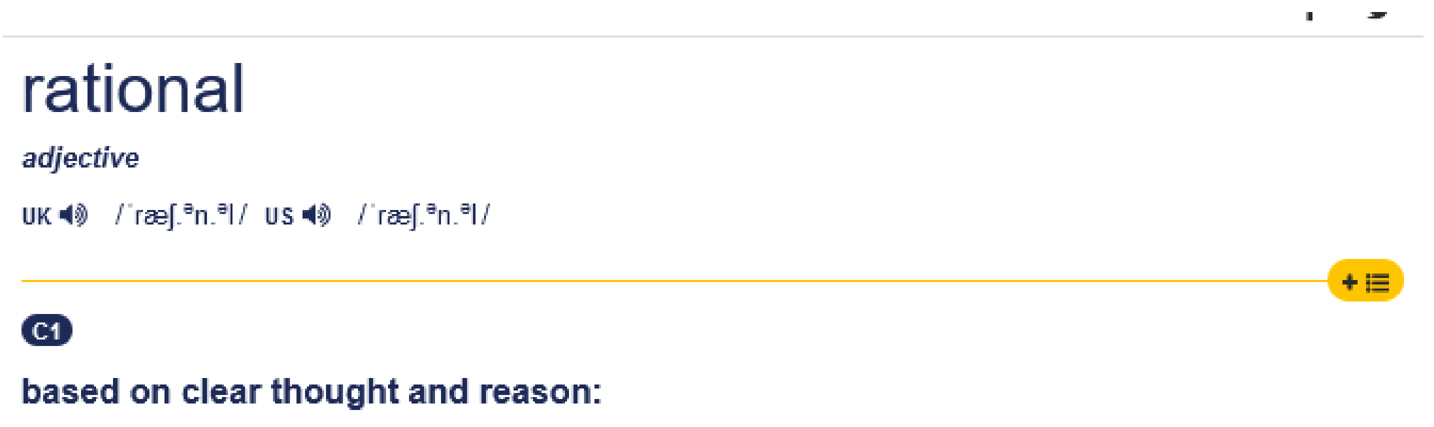
Action

Actions:
- Planning
- Classification
- Forecasting, etc.

Objective: Design of Intelligent (Rational) Agents

- What is a rational agent?
  - ➢ Agent that thinks and acts rationally

- What is rational?


rational

adjective

UK /ˈræʃ.ən.əl/ US /ˈræʃ.ən.əl/

C1

based on clear thought and reason:

https://dictionary.cambridge.org/dictionary/english/rational

Rational Agent

"For each percept sequence, a rational agent should select an action that is expected to maximize its performance, given the evidence provided by the percept sequence whatever the builtin knowledge the agent has"

- **AIMA4e**

1. Getting Started

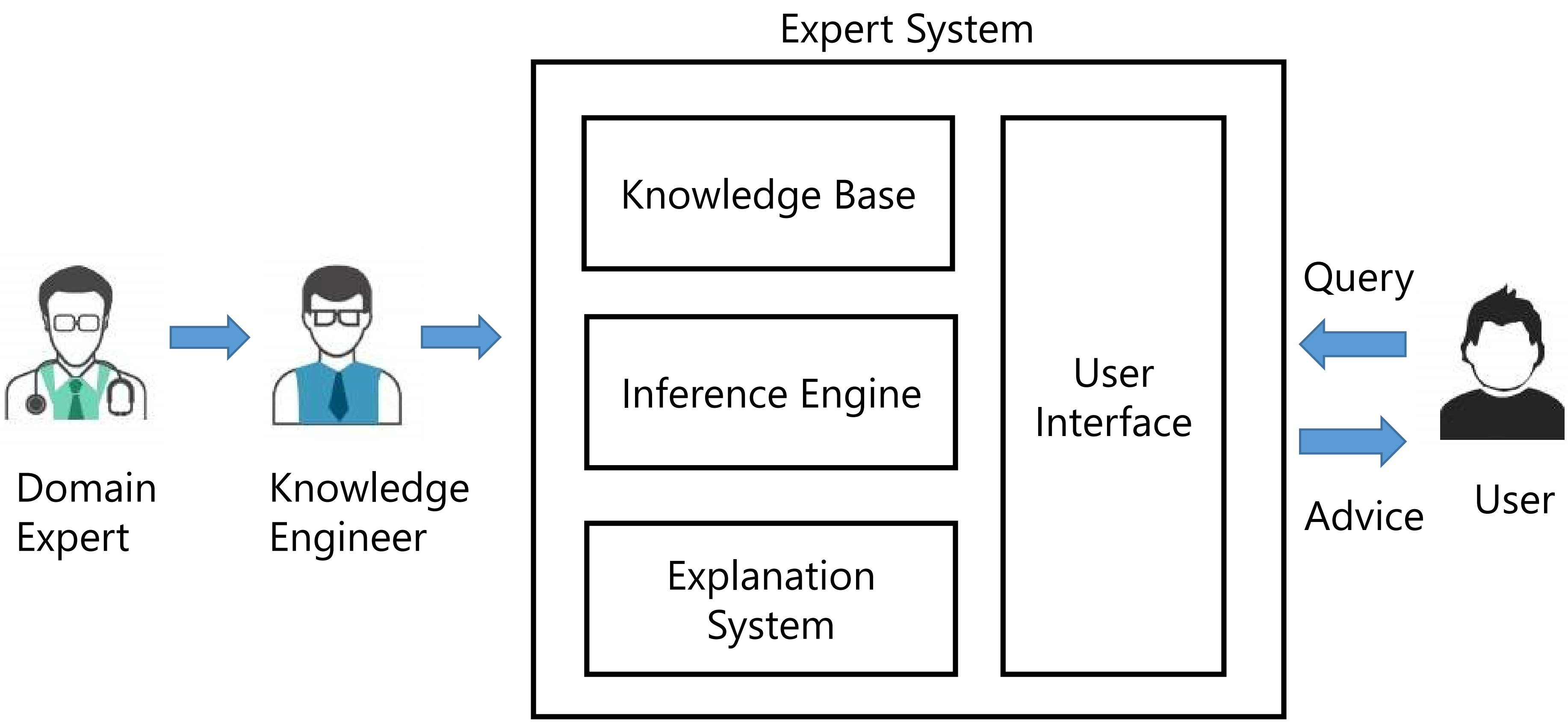
- Intelligent Agents
  - Definitions
  - Objective
- Evolution of AI Systems over years
  - Logic-based AI
  - Expert Systems
  - Machine Learning
  - Deep Learning
- Recent Success Stories
- Way Forward

Logic-based Systems

Logic Theorist is a computer program written in 1956 by Allen Newell, Herbert A. Simon, and Cliff Shaw.[1] It was the first program deliberately engineered to perform automated reasoning and is called "the first artificial intelligence program".[1][a] *See § Philosophical implications* It would eventually prove 38 of the first 52 theorems in Whitehead and Russell's *Principia Mathematica* and find new and more elegant proofs for some.[3]

https://en.wikipedia.org/wiki/Logic_Theorist

Expert Systems



Expert Systems

```
RULE035

PREMISE:  ($AND (SAME CNTXT GRAM GRAMNEG)
                (SAME CNTXT MORPH ROD)
                (SAME CNTXT AIR ANAEROBIC))
ACTION:  (CONCLUDE CNTXT IDENTITY BACTEROIDES TALLY .6)
```

```
IF:   1) The gram stain of the organism is gramneg, and
      2) The morphology of the organism is rod, and
      3) The aerobicity of the organism is anaerobic
THEN:   There is suggestive evidence (.6) that the identity
        of the organism is bacteroides
```

FIGURE 4-3 A MYCIN rule, in both its internal (LISP) form and English translation. The term CNTXT appearing in every clause is a variable in MYCIN that is bound to the current context, in this case a specific organism (ORGANISM-2), to which the rule may be applied.

https://www.aaai.org/Papers/Buchanan/Buchanan06.pdf

Expert Systems: Handling Uncertainty

Bayesian Network

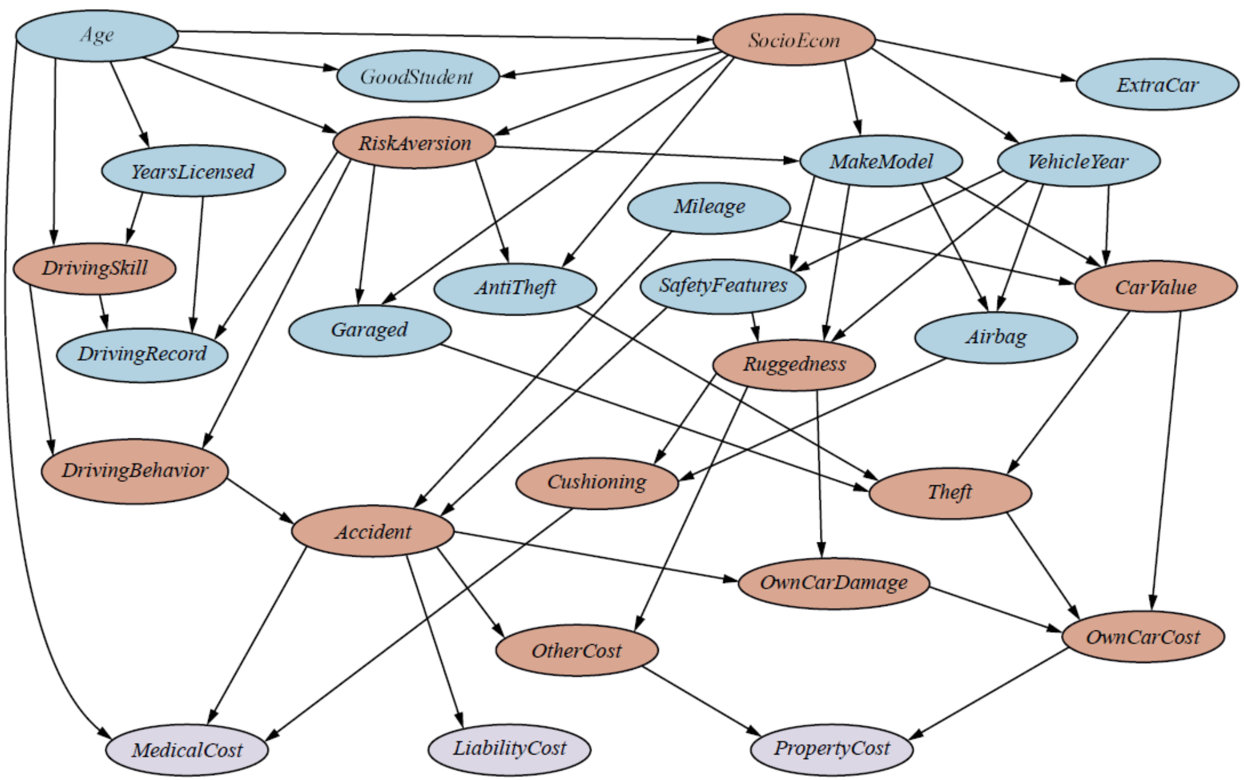
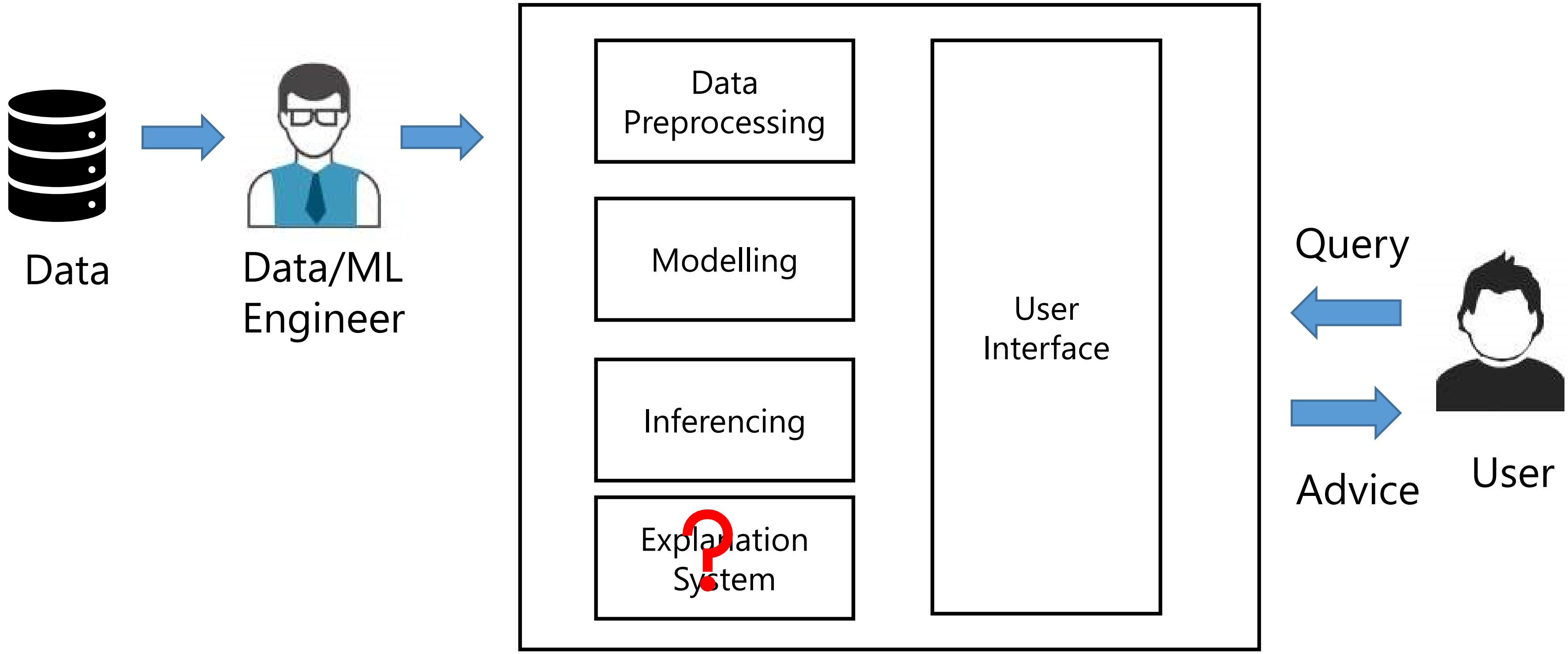
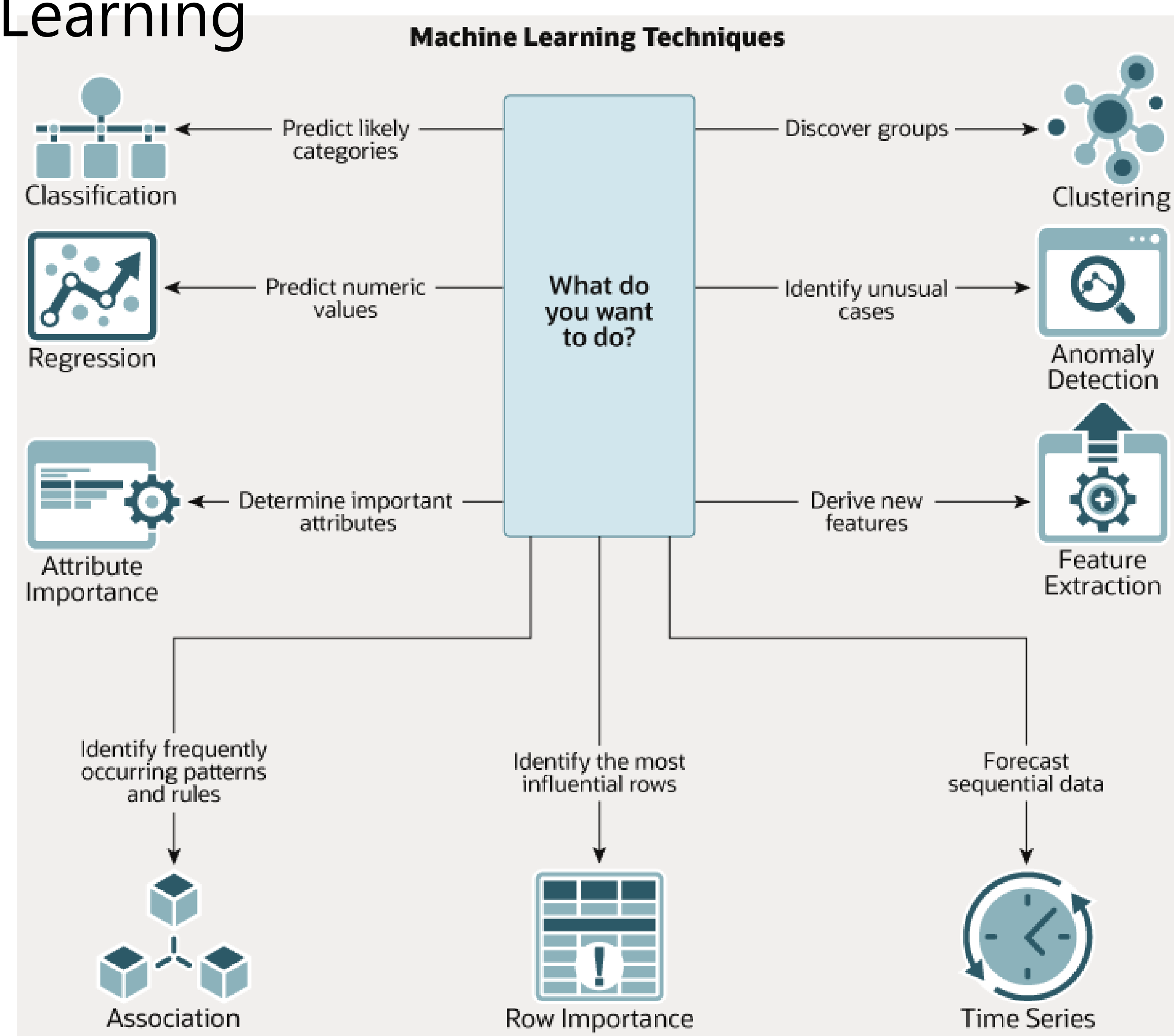


Figure 13.9 A Bayesian network for evaluating car insurance applications.

Machine Learning

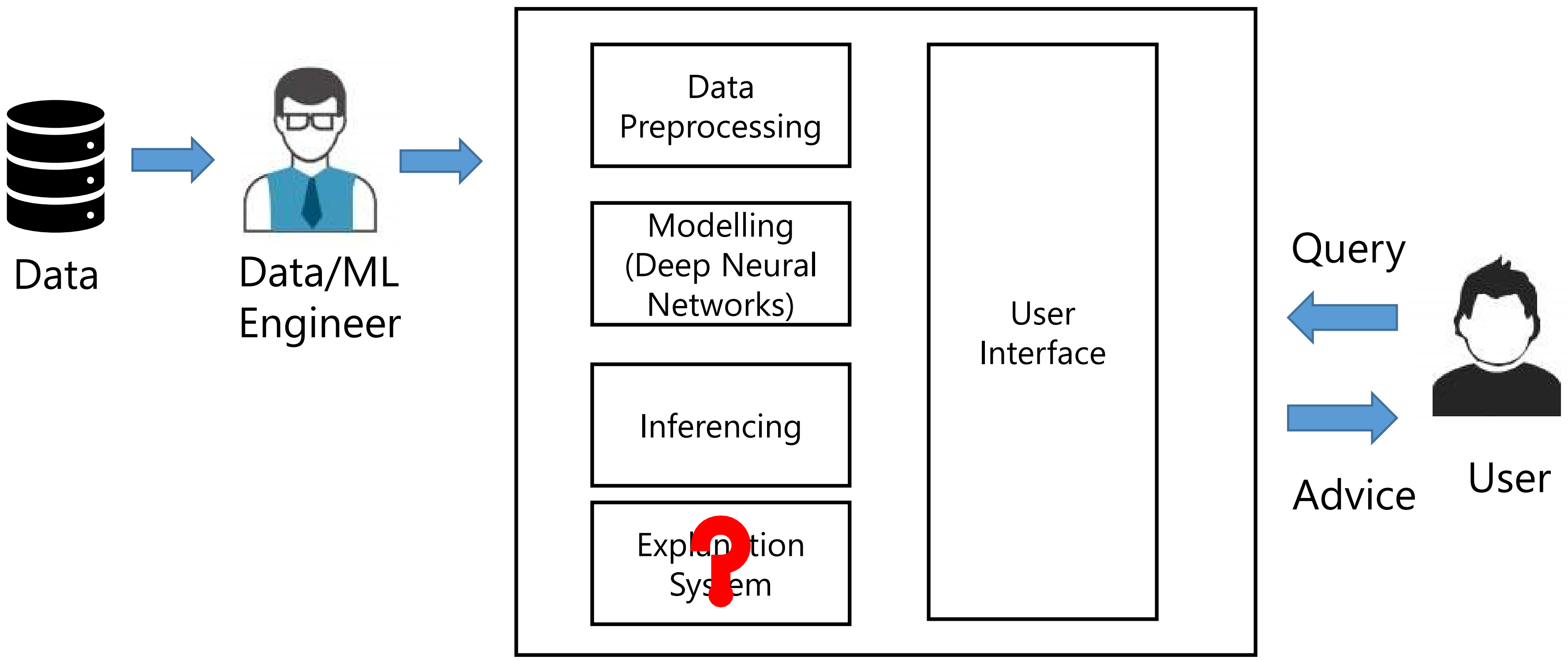


Machine Learning

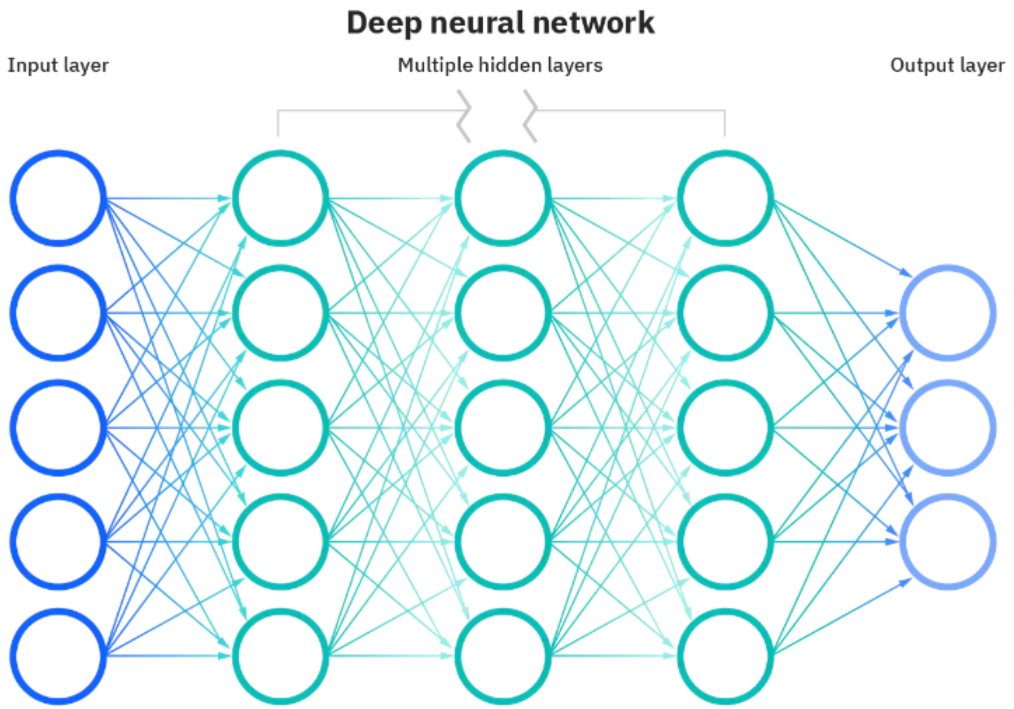


https://docs.oracle.com/en/database/oracle/machine-learning/oml4sql/21/dmapi/oml4sql-basics.html#GUID-2BB02332-4824-4D04-BE04-4265C75BEAA9

Deep Learning

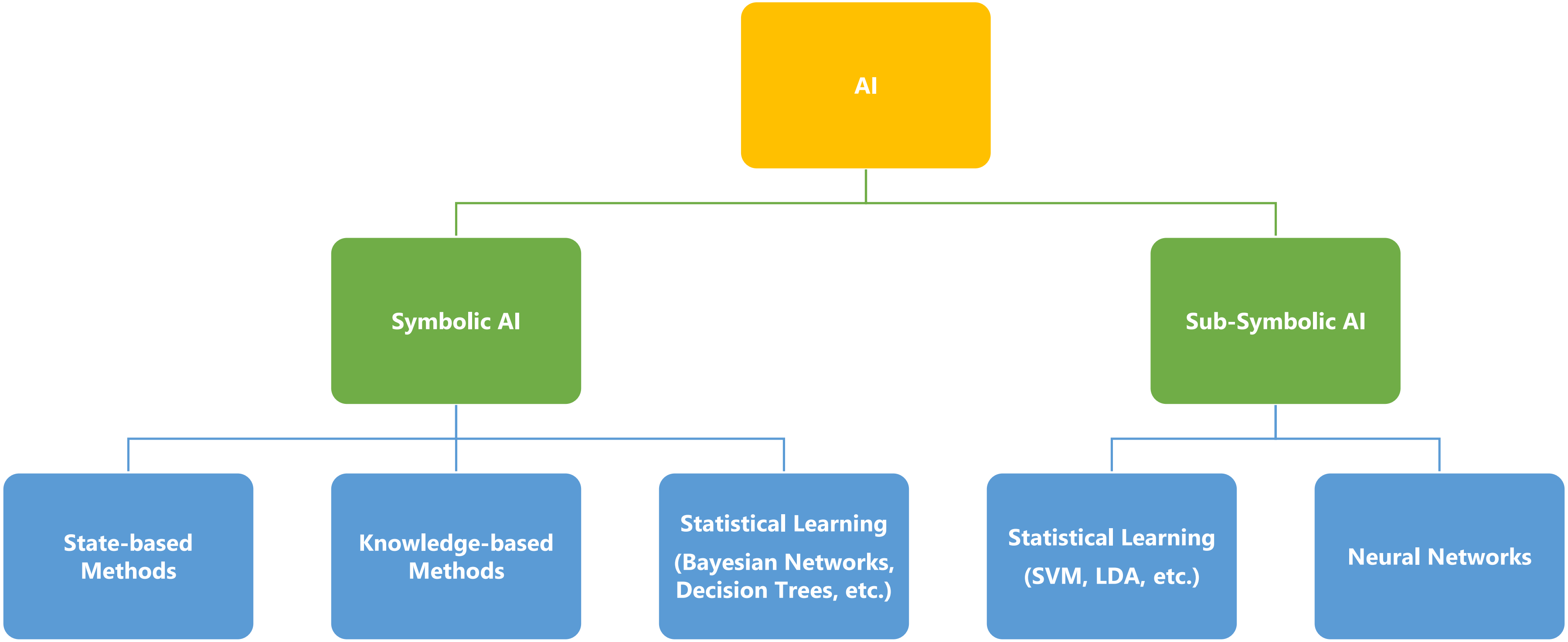


Deep Learning

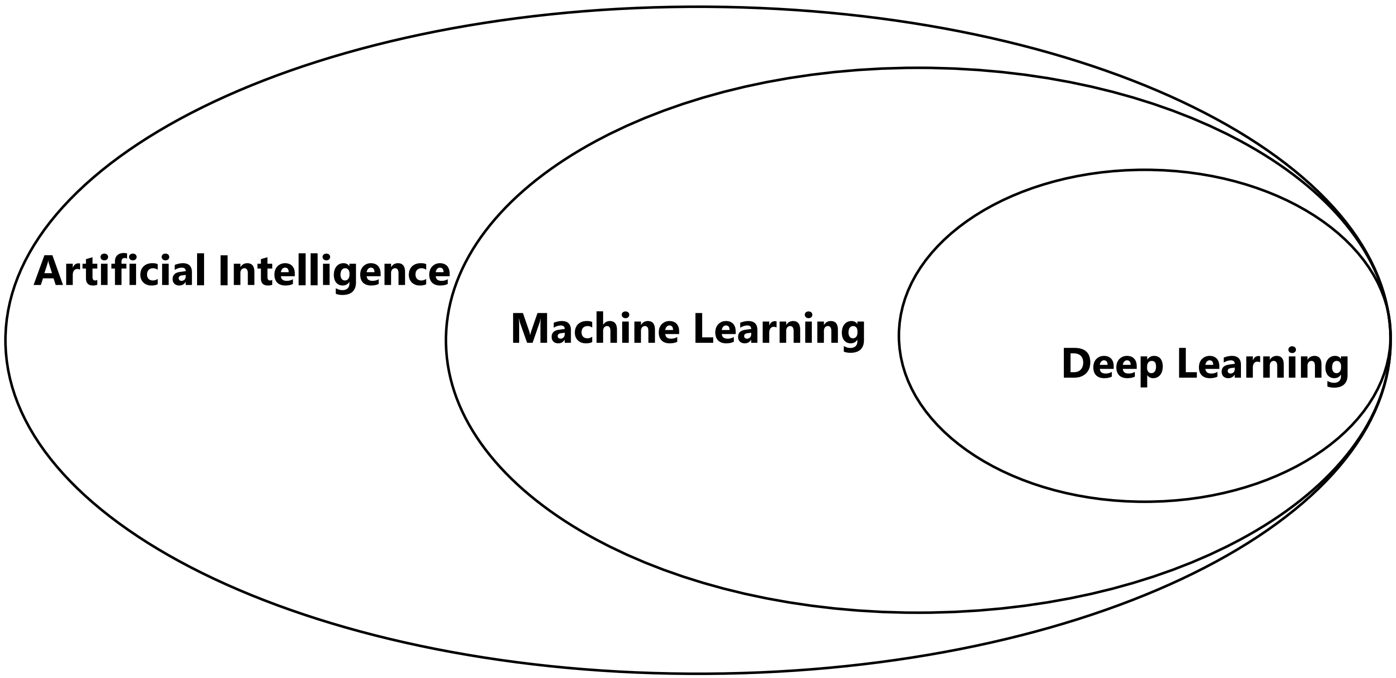


https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks

Taxonomy of AI Techniques



Artificial Intelligence, Machine Learning, and Deep Learning



1. Getting Started

- Intelligent Agents
  - ➢ Definitions
  - ➢ Objective
- Evolution of AI Systems and Taxonomy
  - ➢ Logic-based AI
  - ➢ Expert Systems
  - ➢ Machine Learning
  - ➢ Deep Learning
- Recent Success Stories
- Way Forward

Recent (Success) Stories: Games

Deep Blue beat Gary Kasparov [1997]

Vs

DARPA Grand Challenge (self-driving cars) [2005]

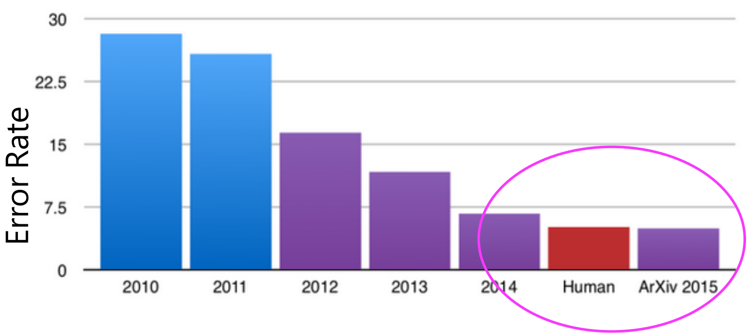
Google's Deep Mind beats world champion Lee Sedol [2016]

Starcraft 2: Deep Mind's AlphaStar [2019]

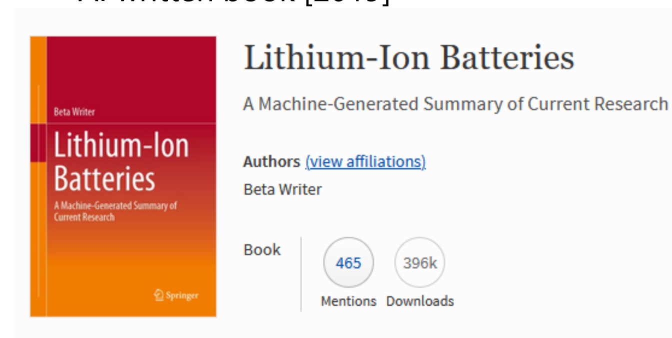
http://content.time.com/time/subscriber/article/0,33009,984305-1,00.html
https://blog.google/technology/ai/alphagos-ultimate-challenge/
Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G. and Lau, K., 2006. Stanley: The robot that won the DARPA Grand Challenge. *Journal of field Robotics*, *23*(9), pp.661-692.

Recent Success Stories

ImageNet Large Scale Visual Recognition Challenge (**ILSVRC**)



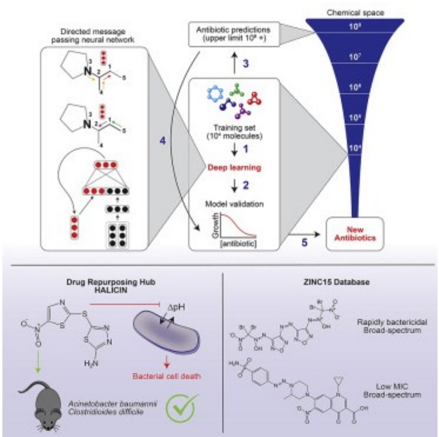
AI written book [2019]



1.2.1 NiO/CNTs Derived from Metal-Organic Frameworks as Superior Anode Material for Lithium-Ion Batteries [1]

That the introduction of CNTs can enhance the lithium-ion storage performance of NiO/CNT composites is demonstrated by the results [1]. That NiO/CNT composites are appealing as potential anodes for Li-ion batteries is demonstrated by the results [1]. At 100 mA g^{-1}, NiO/CNTs-10 shows the highest reversible capacity of 812 mAh g^{-1} after 100 cycles [1]. The excellent electrochemical performance of NiO/CNT composites must be attributable to the formation of 3D conductive network structure with porous NiO microspheres connected by CNTs; this CNTs benefits the buffering of the volume expansion during the cycling process and the electron transfer ability [1]. Reveal performance, which is satisfied, is based on MOFs

Drug by AI: Halicin to kill e-coli

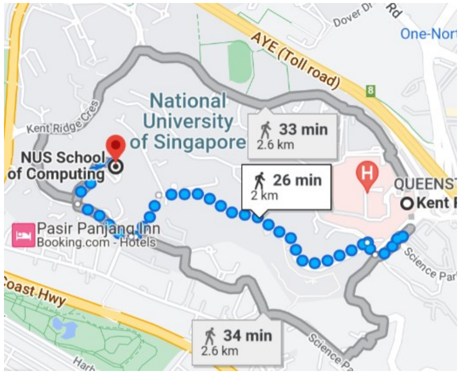


https://developer.nvidia.com/blog/mocha-jl-deep-learning-julia/
https://link.springer.com/book/10.1007/978-3-030-16800-1#about
https://booksby.ai/

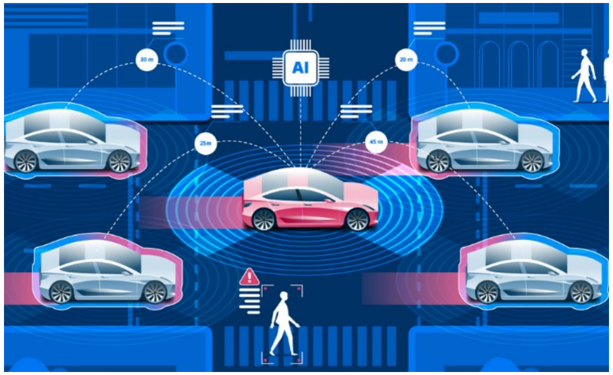
Stokes, J. M. *et al.* A Deep Learning Approach to Antibiotic Discovery, *Cell* https://doi.org/10.1016/j.cell.2020.01.021 (2020)

AI in Daily Life

Virtual Assistants



Maps and Route Planning



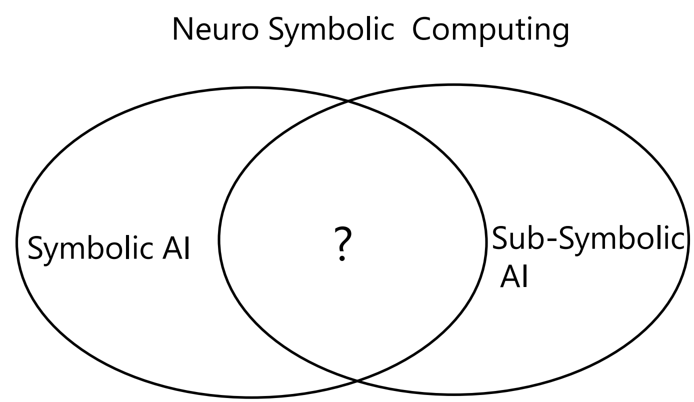
Autonomous Cars

1. Getting Started

- Intelligent Agents
  - Definitions
  - Objective
- Evolution of AI Systems and Taxonomy
  - Logic-based AI
  - Expert Systems
  - Machine Learning
  - Deep Learning
- Recent Success Stories
- Way Forward

What Next for AI?

- Democratization
- Explainable and Trustworthy AI
- Bias in AI
- Industrialization
- Artificial General Intelligence



https://www.gartner.com/en/articles/the-4-trends-that-prevail-on-the-gartner-hype-cycle-for-ai-2021

2 Agent Architecture

2. Agent Architecture

- Design Space

- Workflow
  - State Representation
  - Factored/Feature Representation
  - Relational Representation

- Types of Agents

Design Space

- Agent design choice depends on several factors

  - ➢ Nature of Problem
  - ➢ Nature of Environment
  - ➢ Abilities of Agent
  - ➢ Types of Goals and Desires
  - ➢ Reasoning

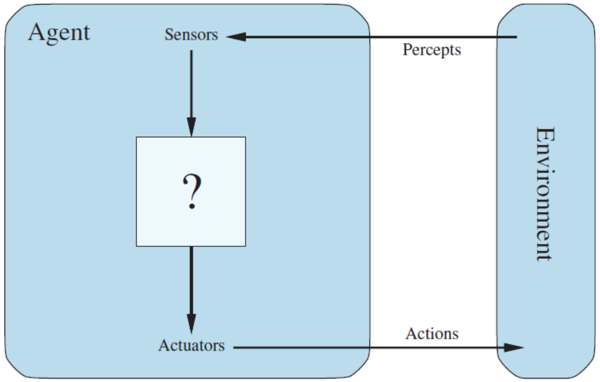


Figure 2.1 Agents interact with environments through sensors and actuators.

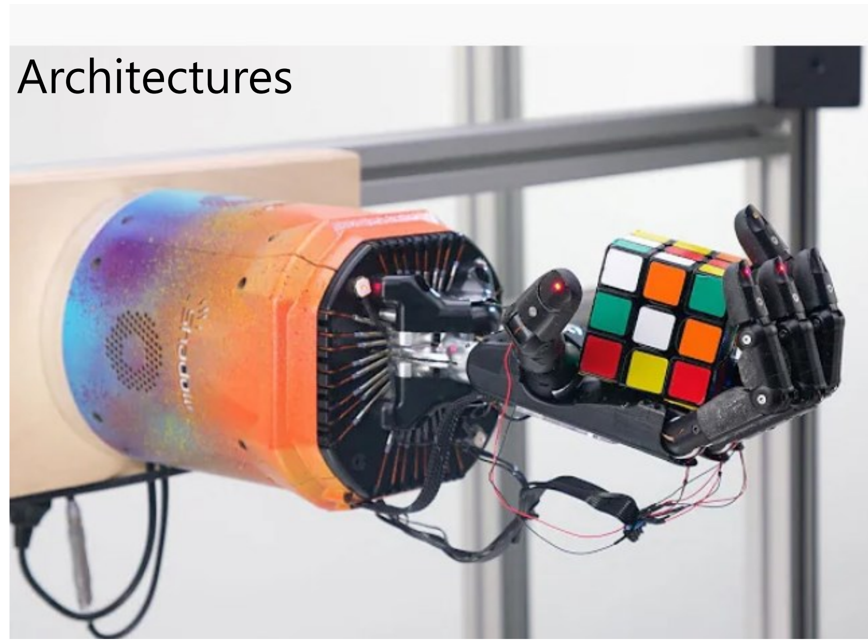
Design Space

Dimension	Values
Modularity	Flat, Modular, Hierarchical
Environment	Static, Dynamic
Representation Scheme	States, Features, Relations
Observability	Fully Observable, Partially observable
Parameter Types	Discrete, Continuous
Uncertainty	Deterministic, Stochastic
Learning	Knowledge is given (known), knowledge is learned (unknown)
Number of Agents	Single Agent, Multiple Agent

AIMA3e (Section 2.3.2) and **AIFCA (Section 1.5)**

Design Space

Flat, Modular, Hierarchical Architectures



https://www.youtube.com/watch?v=kVmp0uGtShk

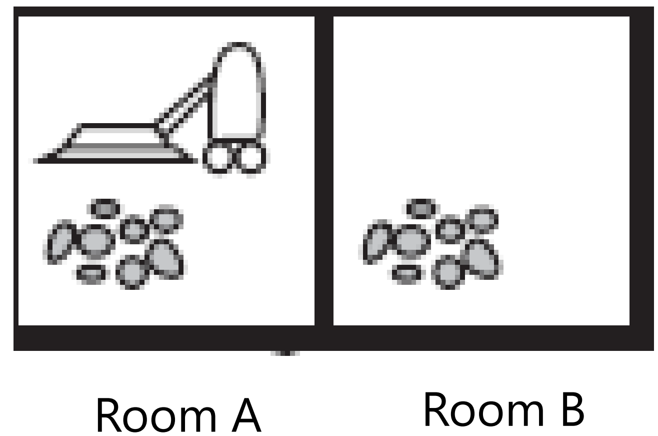
Design Space

- Environment
  - ➢ Static
  - ➢ Dynamic

- Representation
  - ➢ Atomic/State
  - ➢ Factored/Feature
  - ➢ Relational

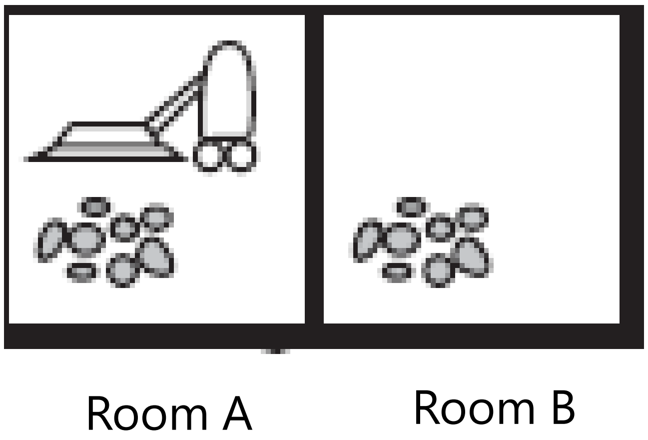
- Observability
  - ➢ Fully Observable
  - ➢ Partially Observable
  - ➢ Zero Observability

- Parameter Types
  - ➢ Discrete/Continuous (Representation, Percepts, Actions)



Design Space

- Uncertainty
  - ➢ Sensing Uncertainty
  - ➢ Action Uncertainty
- Learning
  - ➢ Knowledge is given or not
- Number of Agents
  - ➢ Single
  - ➢ Multiple



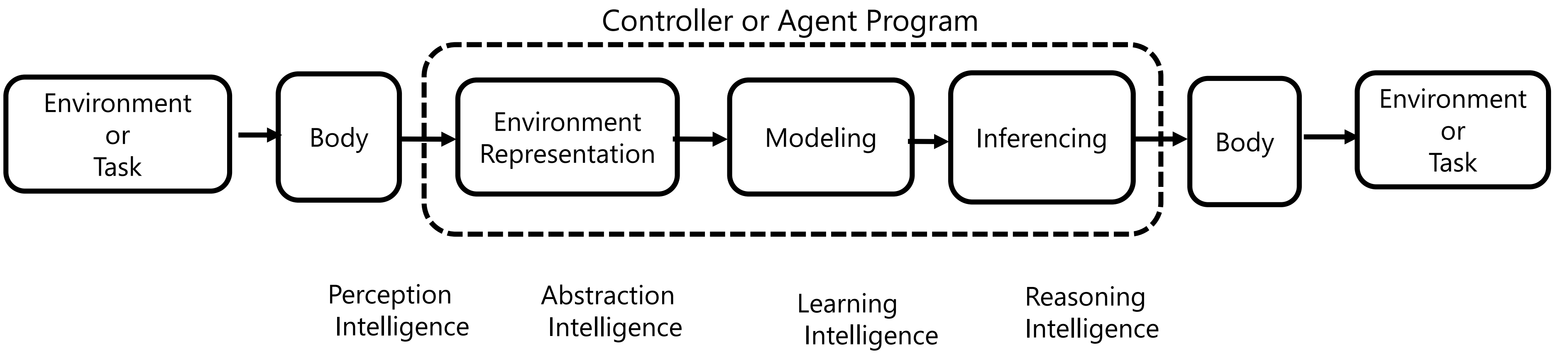
2. Agent Architecture

- Design Space

- Workflow
  - ➢ State Representation
  - ➢ Factored/Feature Representation
  - ➢ Relational Representation

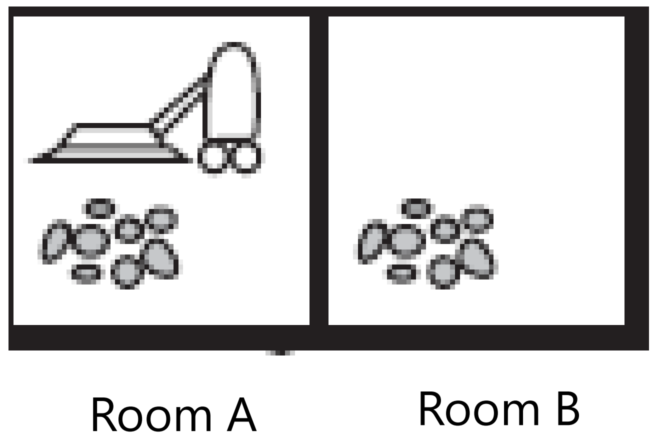
- Types of Agents

Workflow



https://www.darpa.mil/about-us/darpa-perspective-on-ai

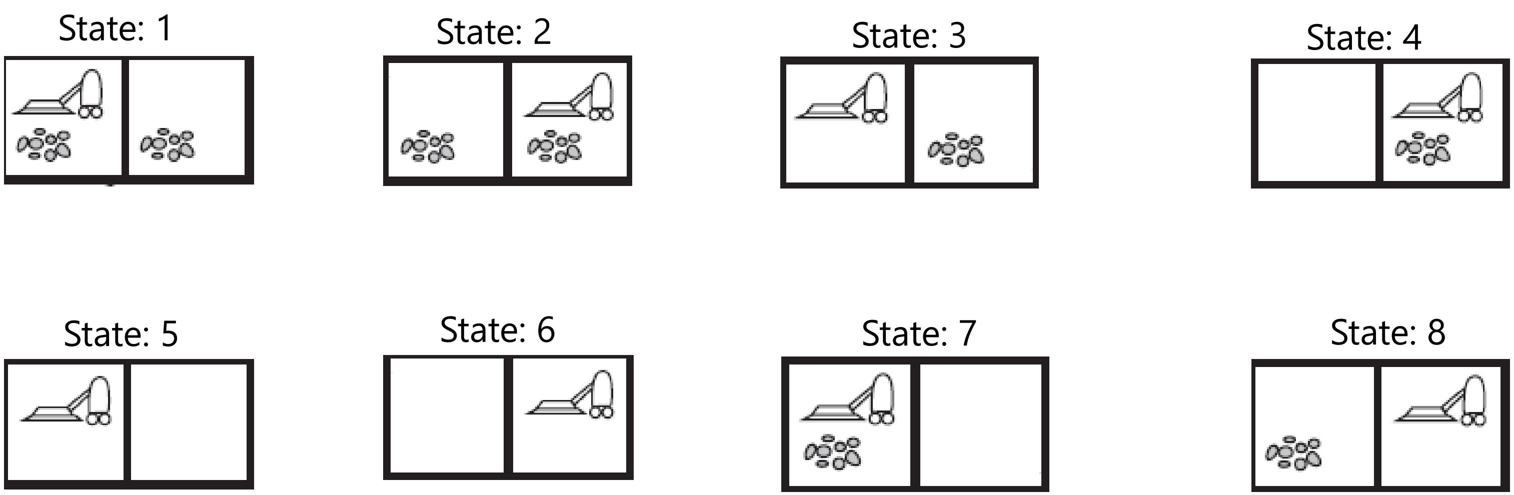
Environment Representation



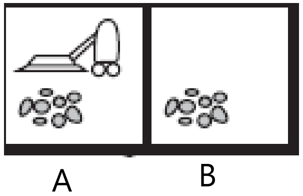
Atomic or State
Features
Relational

State-based Representation

Environment Representation

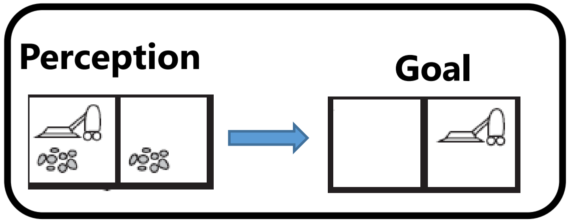


State-based Representation

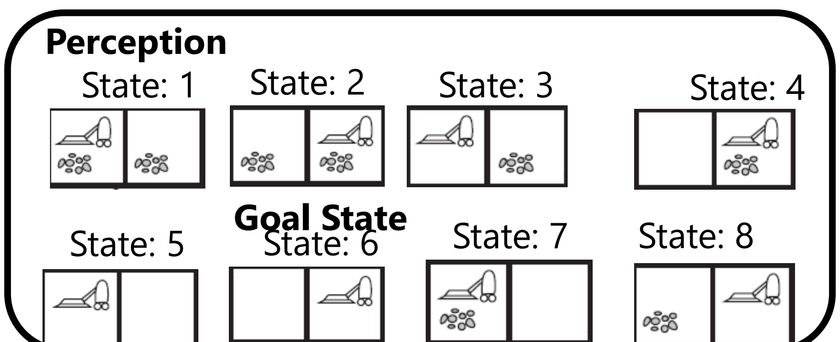


Room-A Status	Room-B Status	Vacuum Location	State
dirty	dirty	A	1
dirty	dirty	B	2
clean	dirty	A	3
clean	dirty	B	4
clean	clean	A	5
clean	clean	B	6
dirty	clean	A	7
dirty	clean	B	8

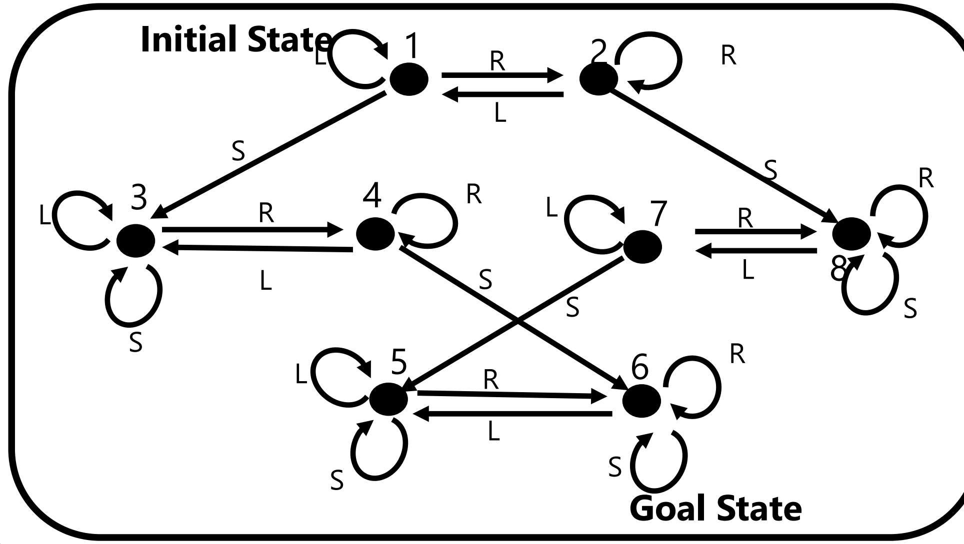
State Space Graph



Environment
Representation



Modeling
(State Space
Modeling)



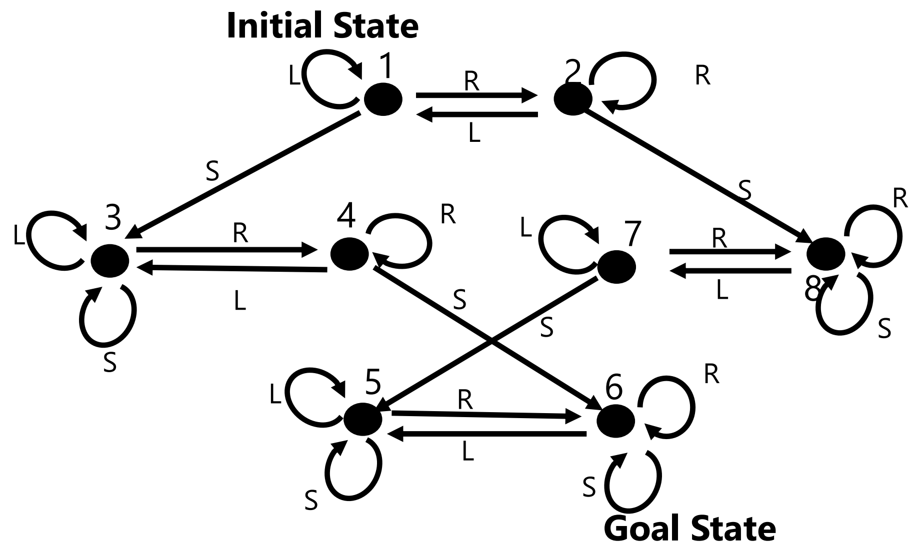
State Space Graph

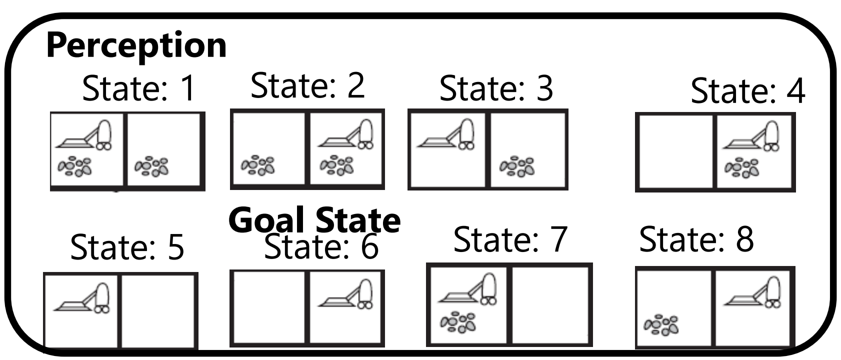
Each node is a state

Edge represents the action

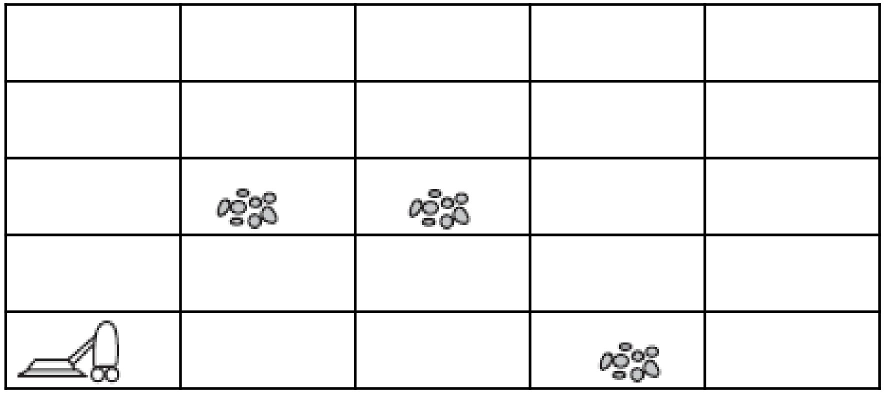
Directed Graph

Inferencing

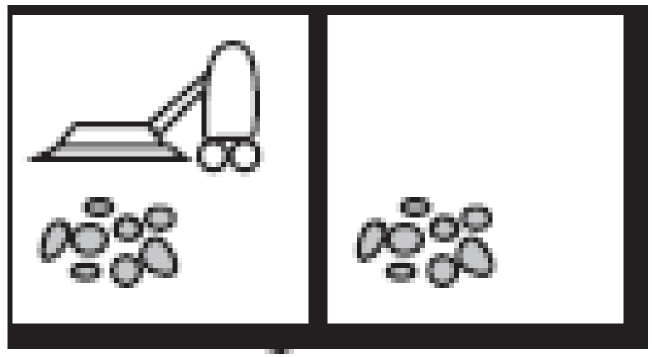




Number of states in $K \times K$ grid?



Factored representation



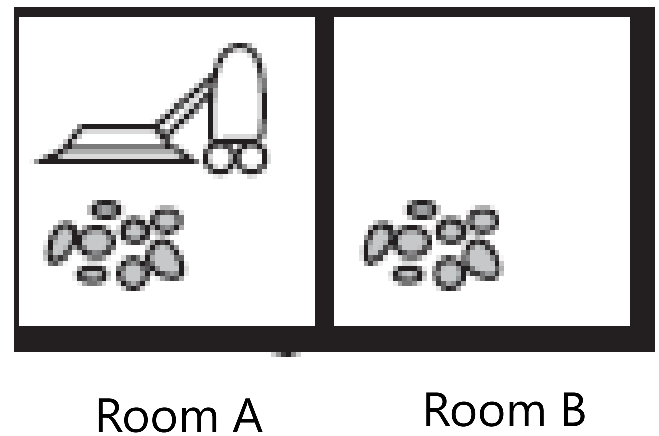
Room A Room B

Two features:

status = {clean, dirty}
location = {A,B}

Factored representation

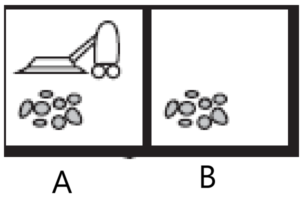
- **Percept**:
  - *status = Dirty*, *location= A*, *location=B*
    - Propositional/Boolean Variables
    - True/False



- **Actions:**
  - *Suck, MoveRight, MoveLeft*

- **Inferencing**
  1. **If** *status* **=** *Dirty* **then return** *Suck*
  2. **If** *location* **=** *A* **and** *status* **=** *Clean* **then return** *MoveRight*
  3. **If** *location* **=** *B* **and** *status* **=** *Clean* **then return** *MoveLeft*

Factored representation

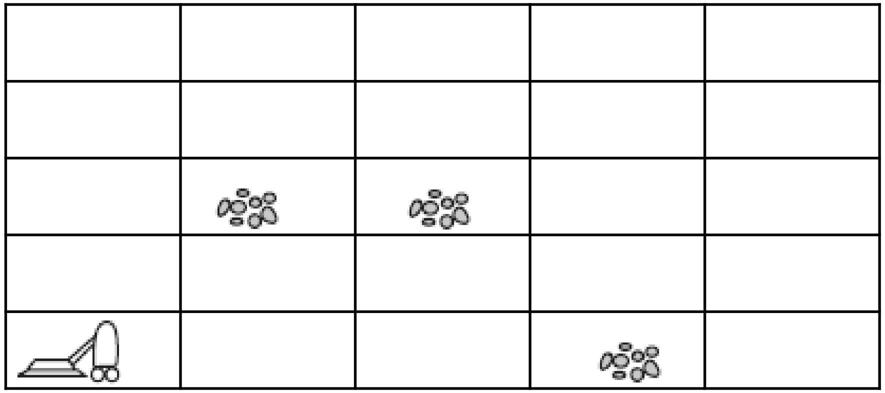


Status = Dirty	Location = A	Location = B
$true$	$true$	$true$
$true$	$true$	$false$
$true$	$false$	$true$
$true$	$false$	$false$
$false$	$true$	$true$
$false$	$true$	$false$
$false$	$false$	$true$
$false$	$false$	$false$

$$\#\ of\ variables = \log_2(\#\ of\ states)$$

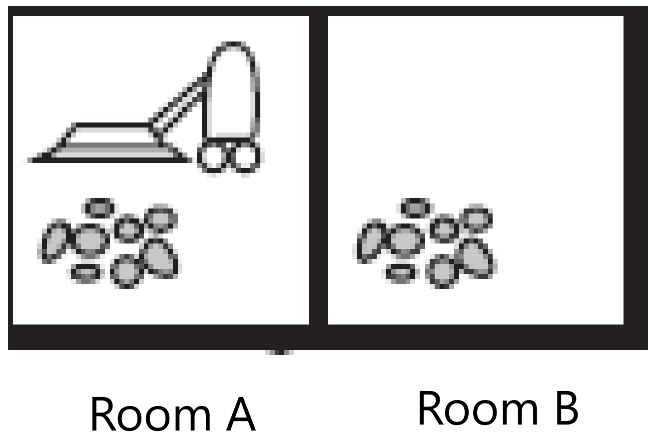
Propositional/Boolean variables

Feature-based Representation



- # rules are still exponential in # of variables

Relational Representation



Objects:

$Vacuum, RoomA, RoomB$

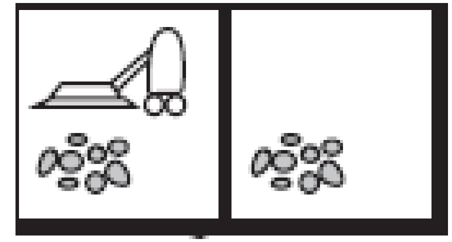
Relations:

$In(x, y)$
$Suck(x, y)$
$Dirty(y)$
$MoveRight(x)$
$MoveLeft(x)$

x: Agent
y: Room

Compact and Expressive Representation

Relational Representation



Room A Room B

Objects:

$Vacuum$, $RoomA$, $RoomB$

Relations:

$In(x, y)$
$Suck(x, y)$
$Dirty(y)$
$MoveRight(x)$
$MoveLeft(x)$

x: Agent
y: Room

Inferencing:

$\forall y$ **If** $In(Vacuum, y)$ **and** $Dirty(y)$, **then** $Suck(x, y)$
If $In(Vacuum, RoomA)$ **and** $not\ Dirty(RoomA)$, **then** $MoveRight(Vacuum)$
If $In(Vacuum\ RoomB)$ **and** $not\ Dirty(RoomA)$, **then** $MoveLeft(Vacuum)$

Representations: Summary

- States
- Factors
- Relations

2. Agent Architecture

- Design Space

- Workflow
  - State Representation
  - Factored/Feature Representation
  - Relational Representation

- Types of Agents

Table-driven Agent

```
function TABLE-DRIVEN-AGENT(percept) returns an action
   persistent: percepts, a sequence, initially empty
               table, a table of actions, indexed by percept sequences, initially fully specified

   append percept to the end of percepts
   action ← LOOKUP(percepts, table)
   return action
```

Figure 2.7 The TABLE-DRIVEN-AGENT program is invoked for each new percept and returns an action each time. It retains the complete percept sequence in memory.

Simple Reflex Agents

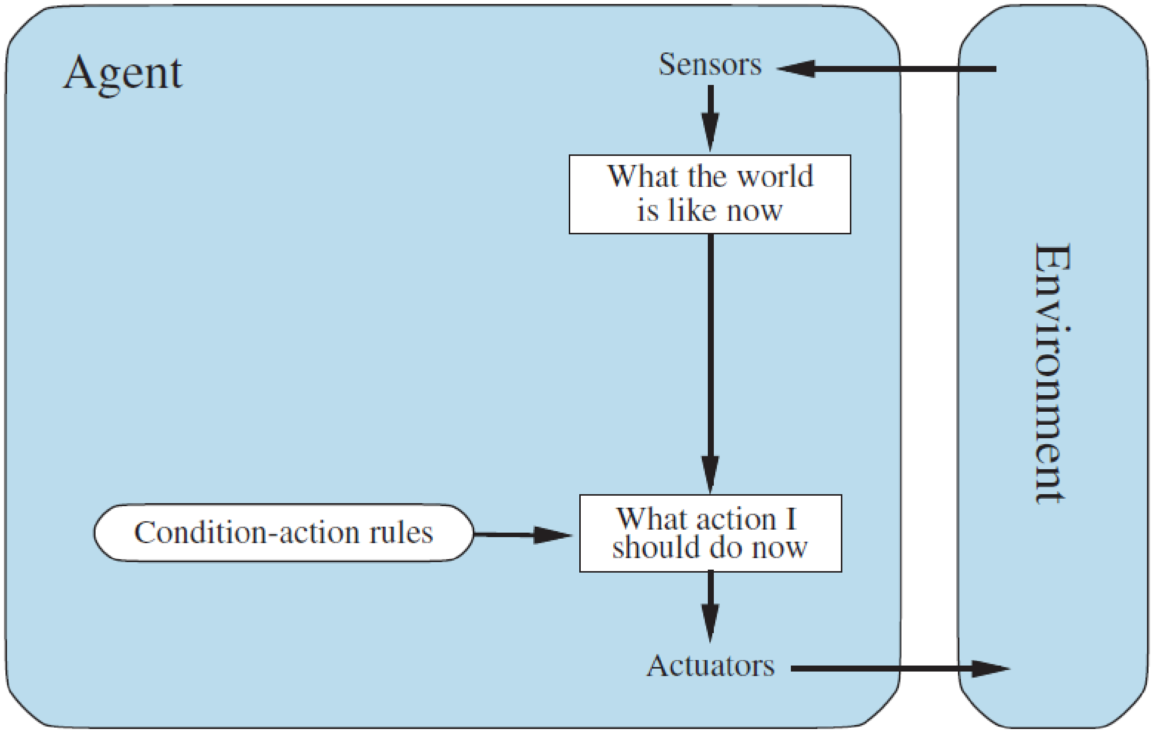


Figure 2.9 Schematic diagram of a simple reflex agent. We use rectangles to denote the current internal state of the agent's decision process, and ovals to represent the background information used in the process.

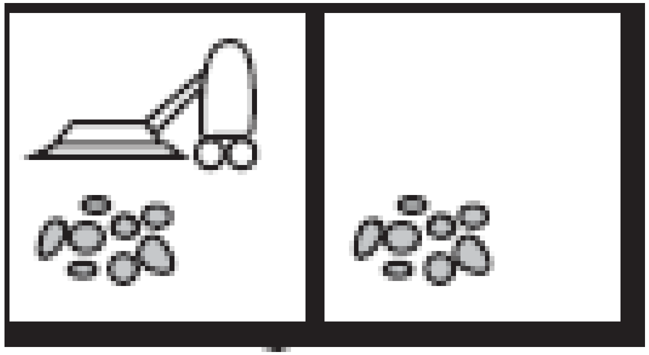
Simple Reflex Agents

function SIMPLE-REFLEX-AGENT(*percept*) **returns** an action
  persistent: *rules*, a set of condition–action rules

  state ← INTERPRET-INPUT(*percept*)
  rule ← RULE-MATCH(*state*, *rules*)
  action ← *rule*.ACTION
  return *action*

Figure 2.10 A simple reflex agent. It acts according to a rule whose condition matches the current state, as defined by the percept.

Simple Reflex Agents: Example



Room A   Room B

function REFLEX-VACUUM-AGENT([*location*,*status*]) **returns** an action

  if *status* = *Dirty* **then return** *Suck*
  else if *location* = *A* **then return** *Right*
  else if *location* = *B* **then return** *Left*

Figure 2.8 The agent program for a simple reflex agent in the two-location vacuum environment. This program implements the agent function tabulated in Figure **??**.

Model and Goal-based Agent

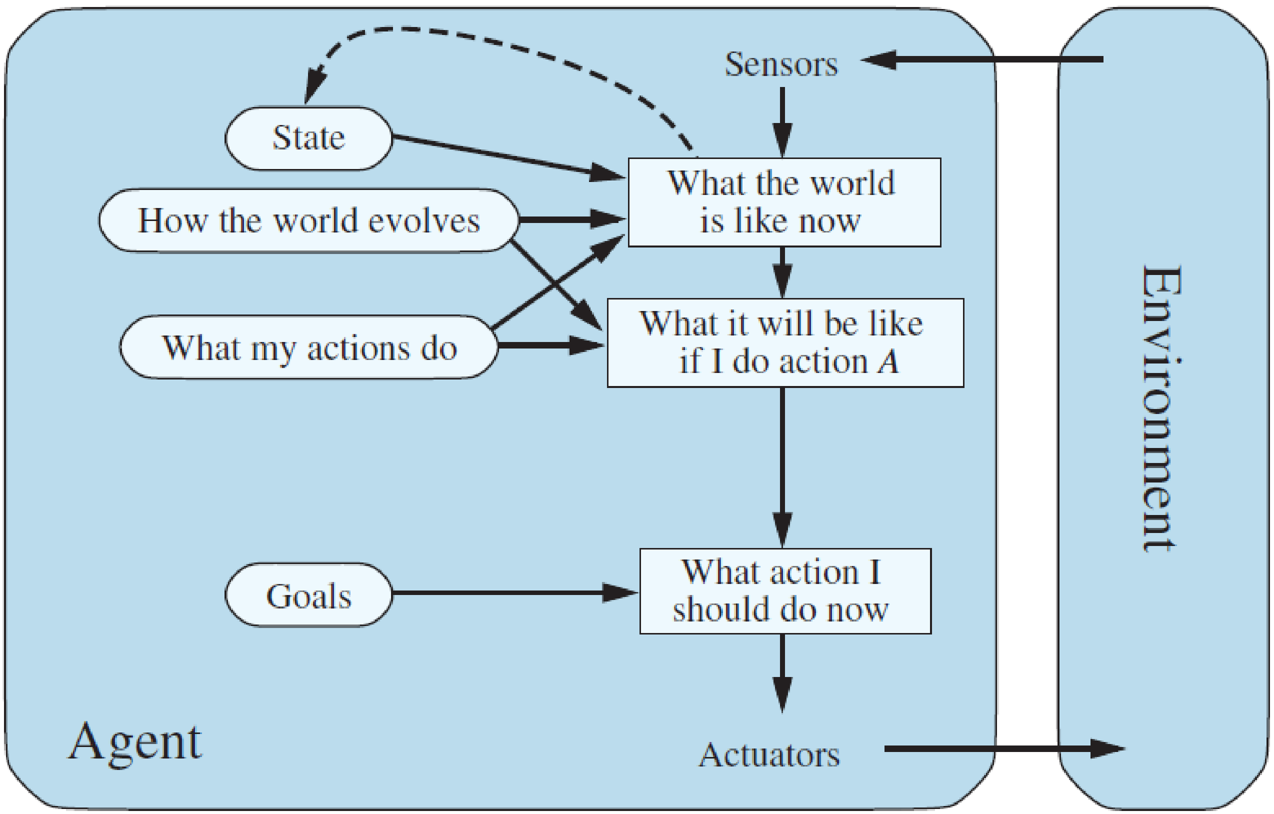


Figure 2.13 A model-based, goal-based agent. It keeps track of the world state as well as a set of goals it is trying to achieve, and chooses an action that will (eventually) lead to the achievement of its goals.

Model-based and Goal-based Agent

Percept x	Action z
1.0	1.000000000000000
1.1	1.048808848170152
1.2	1.095445115010332
1.3	1.140175425099138
1.4	1.183215956619923
1.5	1.224744871391589
1.6	1.264911064067352
1.7	1.303840481040530
1.8	1.341640786499874
1.9	1.378404875209022
⋮	⋮

function SQRT(x)
  $z \leftarrow 1.0$ /* *initial guess* */
  repeat until $|z^2 - x| < 10^{-15}$
    $z \leftarrow z - (z^2 - x)/(2z)$
  end
  return z

Figure 2.2 Part of the ideal mapping for the square-root problem (accurate to 15 digits), and a corresponding program that implements the ideal mapping.

Utility-based Agent

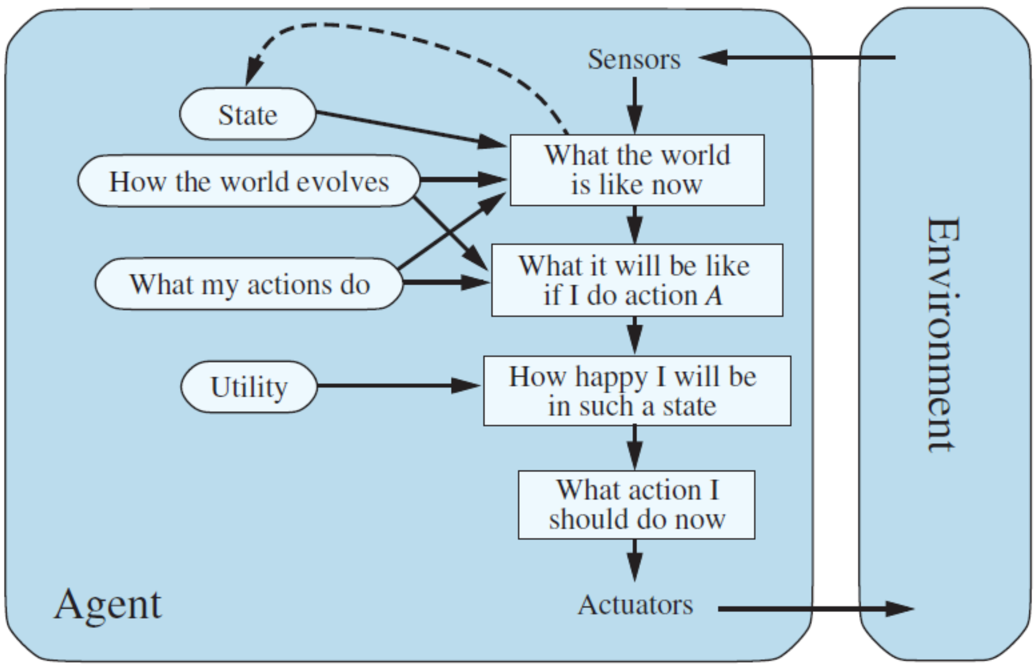


Figure 2.14 A model-based, utility-based agent. It uses a model of the world, along with a utility function that measures its preferences among states of the world. Then it chooses the action that leads to the best expected utility, where expected utility is computed by averaging over all possible outcome states, weighted by the probability of the outcome.

Learning-based Agent

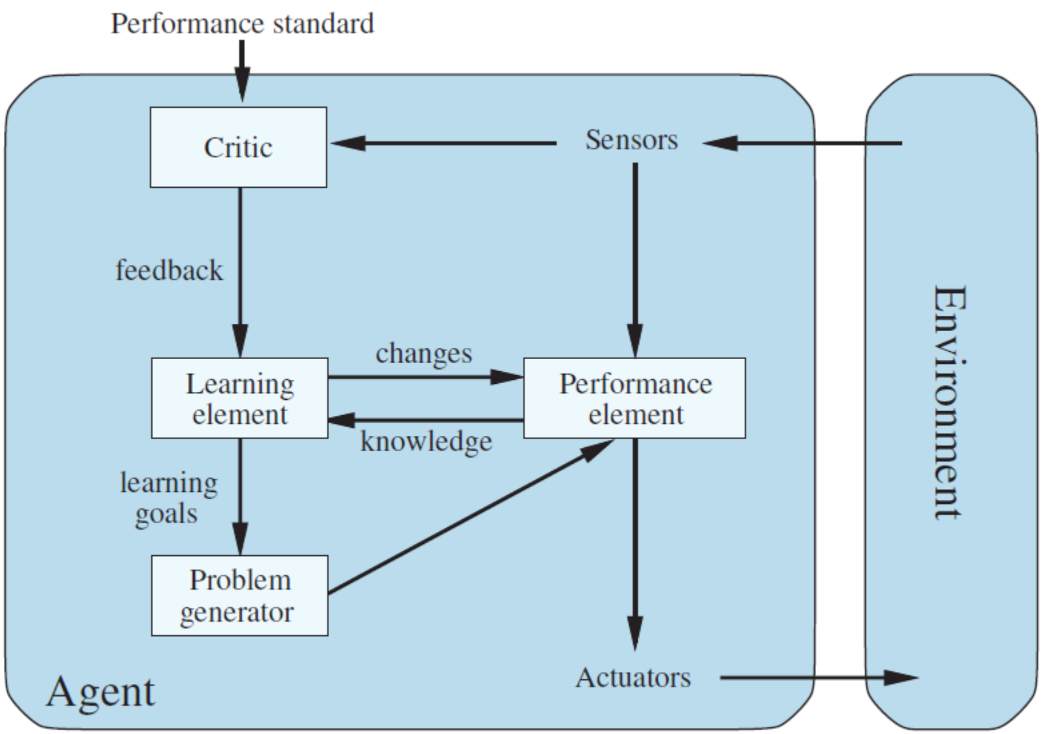


Figure 2.15 A general learning agent. The “performance element” box represents what we have previously considered to be the whole agent program. Now, the “learning element” box gets to modify that program to improve its performance.

AI Agent vs Traditional Program

AI Computing

- Declarative
  - Separates knowledge from its reasoning
  - Uses rules, symbolic reasoning, and can learn from data
- Explains the decisions
- Allows inexact reasoning
- Easy to add new knowledge

Conventional Computing

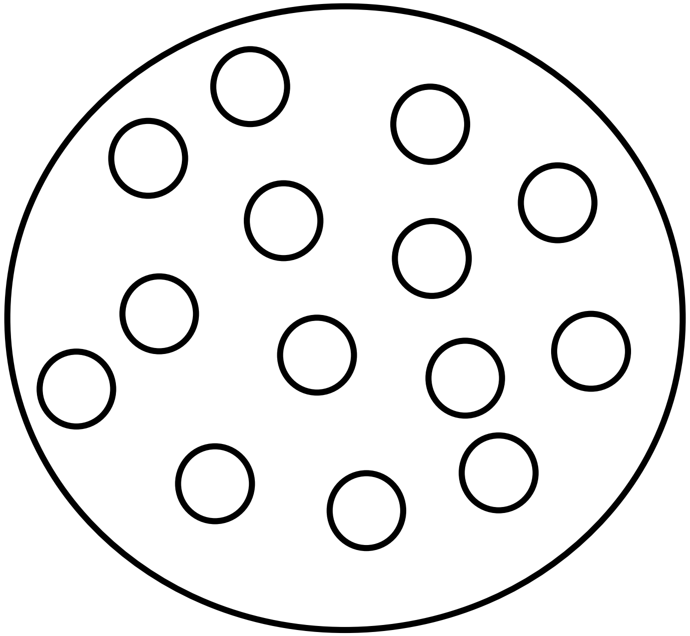
- Imperative
  - No separation between knowledge and reasoning
  - Do not separate knowledge from control structure to process knowledge
- Do not explain the decisions
- Works only on complete and exact data
- Difficult to revise with new knowledge

Artificial Intelligence: A guide to Intelligent Systems: Table 2.1

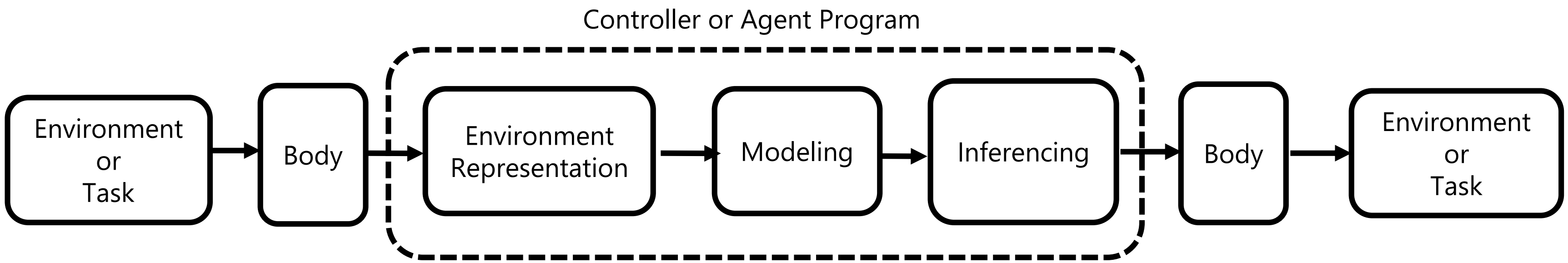
3. Moving Forward

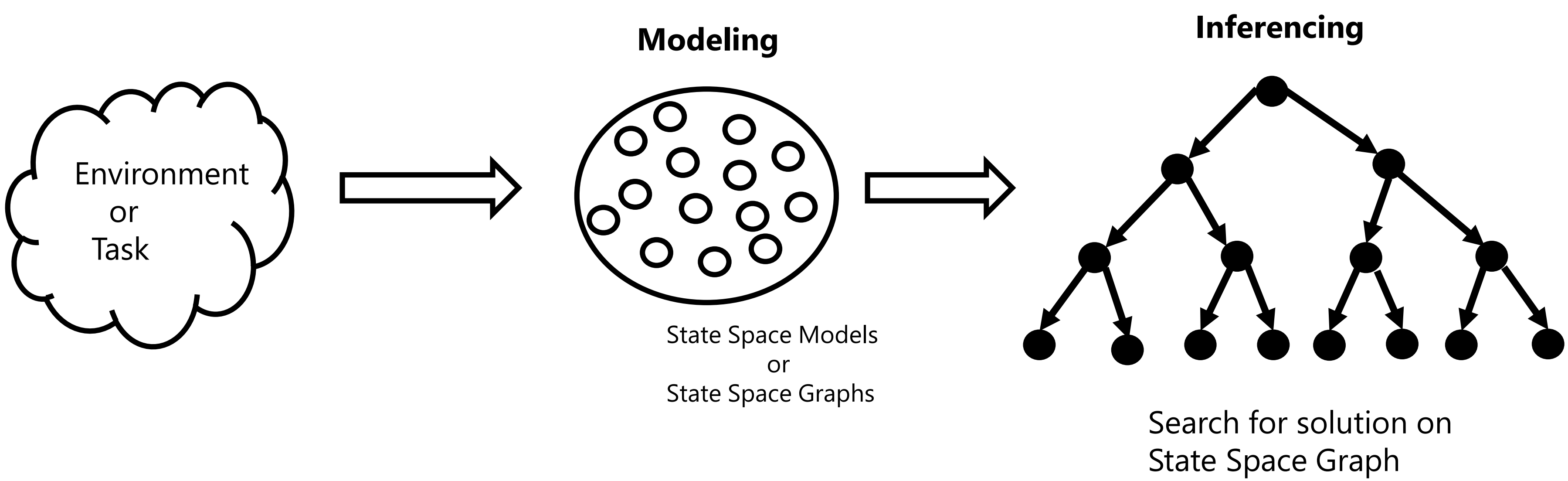
Modeling with Atomic Representation

- Single Agent
  - Uninformed Search
    - Goal is known
    - **Objective**: find path to goal
  - Informed Search
    - Goal is known
    - Use heuristics
    - **Objective**: find path to goal
  - Local Search
    - Configuration Problem
    - Path to goal is irrelevant
    - **Objective**: find the optimal
  - Markov Decision Processes
    - Goal is known
    - Action effects are uncertain
    - **Objective**: Find an optimal policy
- Multiple Agents
  - Adversarial Search
    - Two agents with competing objectives
    - **Objective**: Maximize utility of an agent



Problem Solving by Search: Workflow





Problem Solving by Search: Applications

- Puzzles
  - Slide Puzzles,
  - N-Queens
  - Rubik's cube, etc.

- Games
  - Pacman
  - Tic-Tac-Toe
  - Chess
  - Black Gammon, etc.

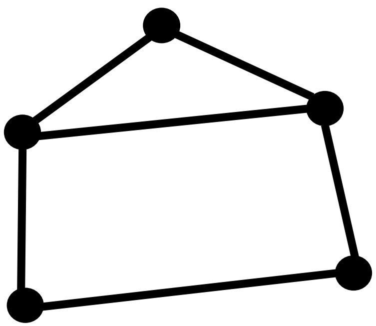
- Real-World Applications
  - Route Planning
  - Robot Motion Planning
  - VLSI Layout Planning
  - Job Scheduling, etc.

Preliminaries: Data Structures

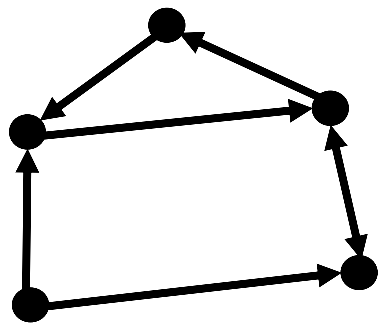
- List
- Tuple
- Dictionary
- Linked List
- Queue
- Stack
- Priority Queue

Preliminaries: Data Structures

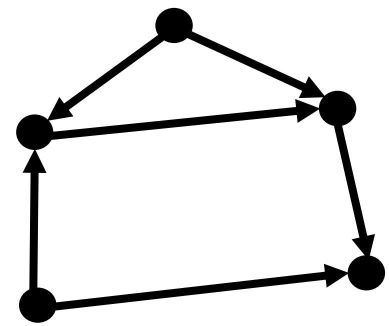
A



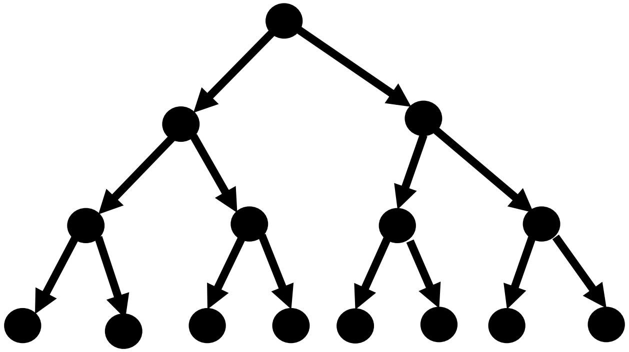
B



C

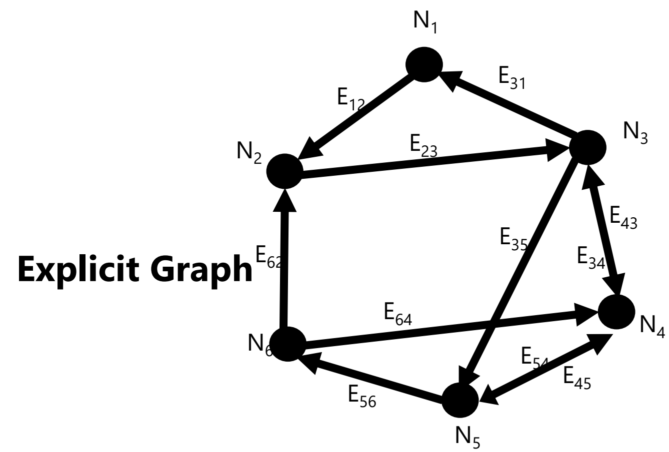


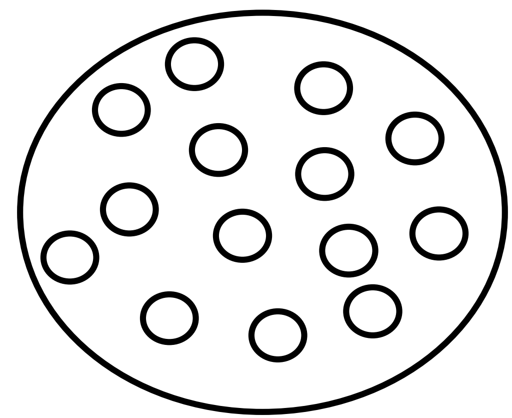
D



Preliminaries: Graphs

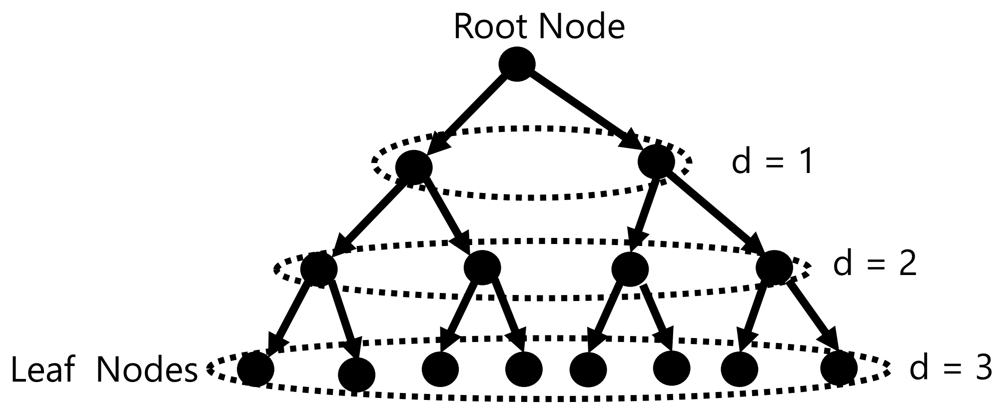
- How to describe graphs?
  - ➢ Explicit Graphs
    - o Adjacency Lists
    - o Linked Lists
  - ➢ Implicit Graphs
    - o Graph is not explicitly generated
    - o Obtained through modeling or problem formulation
    - o Can have infinite number of nodes



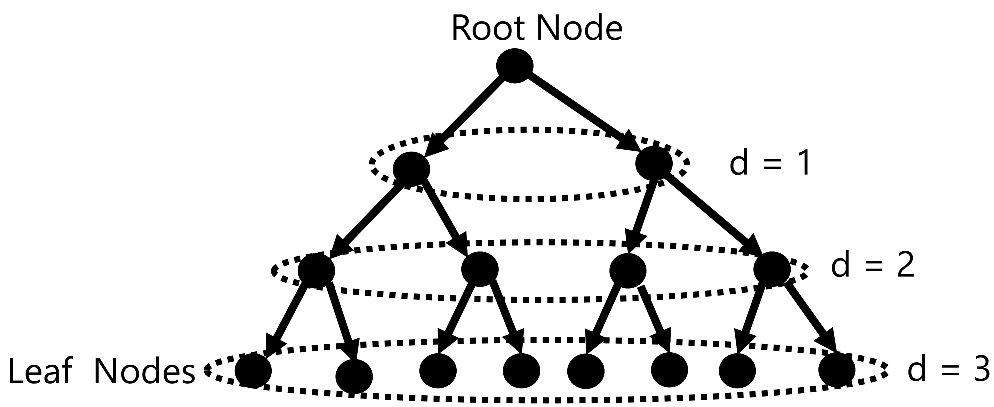


Implicit Graph

Preliminaries: Trees



Preliminaries: Trees



$$\begin{aligned}
\text{Branching Factor} &= b \\
\text{Maximum Depth} &= m \\
\text{Number of Nodes at depth } d &= b^d \\
\text{Number of Leaf Nodes} &= b^m \\
\text{Number of Nodes} &= 1 + b + b^2 + \ldots b^d + .. + b^m
\end{aligned}$$

Preliminaries: Performance Measures

- Completeness
  - Complete: if algorithm can reach goal
  - Incomplete: if algorithm cannot reach goal
- Sound
  - If algorithm is providing correct answers
- Optimality (*aka* rationality)
  - Optimal: if algorithm finds an optimal (lowest cost) path to goal
- Time Complexity
  - Time taken to find the solution
  - Measured in terms of number of nodes generated and visited
- Space Complexity
  - Memory needed to find the solution
  - Measured in terms of number of nodes stored while building the graph

Prerequisites

- Familiarity with the following
  - Data Structures
    - Linked Lists
    - Stacks, Queues, Priority Queues, etc.
  - Algorithms
    - Search Algorithms
    - Dynamic Programming
  - Discrete Mathematics
    - Truth Tables of Logical Operators
    - Theorem Proving (Modus Ponens, etc.)
  - Probability Theory and Random Variables
    - Discrete Random Variables
      - Joint Distributions
      - Expectation
  - Numerical Methods
    - Iterative methods for solving equations

Conclusions

- AI Systems
- Agent Architecture
- Environment Representation
  - States
  - Features
  - Relational
- Types of Agents
- Data Structures
  - Preliminaries