KNOWLEDGE PROBLEM SOLVING

Fundamentals of Artificial Intelligence

Session 07a

Pramod Sharma
pramod.sharma@prasami.com

2

Agenda

- Introduction
- Knowledge
- Knowledge Representation
- State Space Search

3

The World as we Know it

Is Sun really going around Earth?

When we see a Tree, is it really there or creation of our brain!!

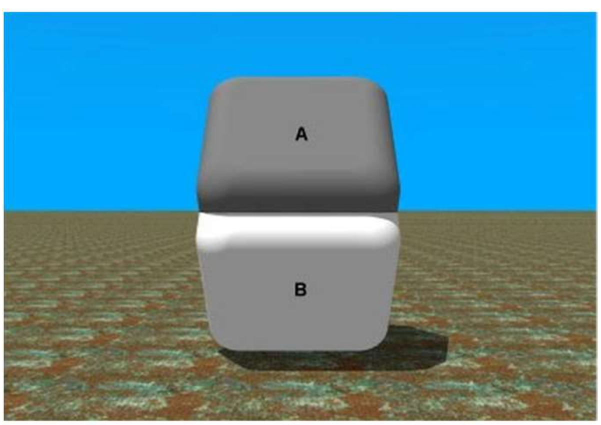
Why do we have mirage!!

4/10/2024

pra-sâmi

4

Popular Illusion



Which area has darker shade, A or B?

Still?

4/10/2024

pra-sâmi

5

The World as we Know it

- We see world around us as we see…
  - Lot of work by eminent Philosophers and Researchers
  - They always wondered how to represent world around us
- In terms of basics,
  - Average adult human consists of 10^{27} atoms and is interacting with zillions of particles
  - Try modeling a person throwing a ball and another person catching it!
- We do not have enough compute power to model this interaction!
  - It's better to model at an abstract level
- Representing in these abstract concepts is called **ontological engineering**

pra-sâmi

6

What is State?

- All information about state…
- All information necessary to make decision about task at hand

pra-sâmi

7

Agent’s Knowledge Representation

Atomic	States are indivisible; No internal structure	Search on atomic states; Heuristic and Blind searches
Propositional (aka Factored)	States are made of state variables that take values (Propositional or multi-values or Continuous)	Search + Inference in (prop logic) and probabilistic (Bayes Nets) representations
Relational	States Describe the objects in the world and their inter-relations	Search + Inference in predicate logic (or relational prob. Models)
First-order	States Describe the objects in the world and their inter-relations along with Functions over the objects	Search + Inference in first order logic

8

Solving Problems by Searching

- How an agent can look ahead to find a sequence of actions that will eventually achieve its goal
- When the correct action to take is not immediately obvious, an agent may need to plan ahead
  - To consider a sequence of actions that form a path to a goal state
- Such an agent is called a problem-solving agent,
- The computational process it undertakes is called search

9

State Space Search

pra-sâmi

10

Defining the problem as State Space Search

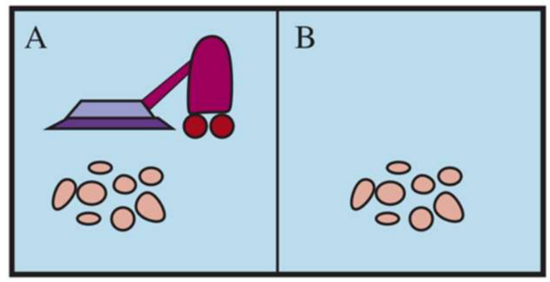
- The state space representation forms the basis of most of the AI methods.
- Its structure corresponds to the structure of problem solving in two important ways:
  - It allows for a formal definition of a problem as the need to convert some given situation into some desired situation using a set of permissible operations.
  - It permits us to define the process of solving a particular problem as a combination of known techniques (each represented as a rule defining a single step in the space) and search, the general technique of exploring the space to try to find some path from current state to a goal state.
- Search is a very important process in the solution of problems for which no more direct techniques are available.

pra-sâmi

11

Toy Vacuum World

- Two Locations : squares A and B
- The vacuum agent perceives:
  - Which square it is in
  - Whether there is dirt in the square
- Actions: move left, move right, suck up the dirt, or do nothing
- Agent function:
  - if the current square is dirty, then suck;
  - otherwise, move to the other square



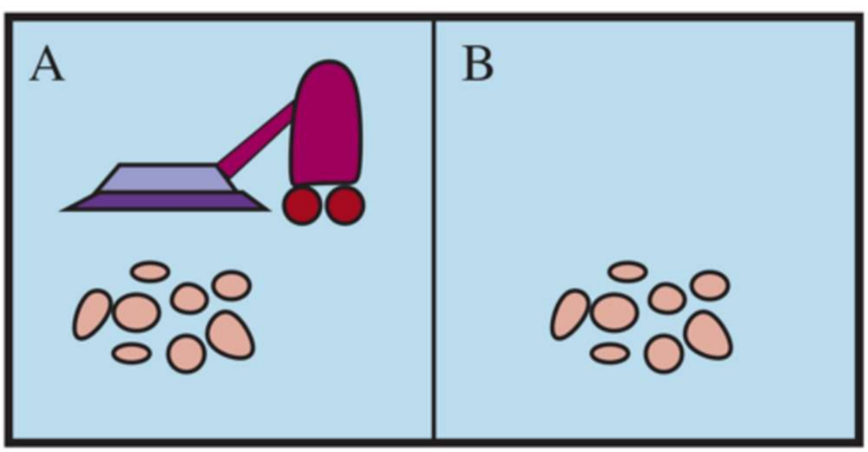
Find most appropriate action by the Agent (vacuum cleaner)!

4/10/2024

pra-sâmi

12

Toy Vacuum World – Possible states

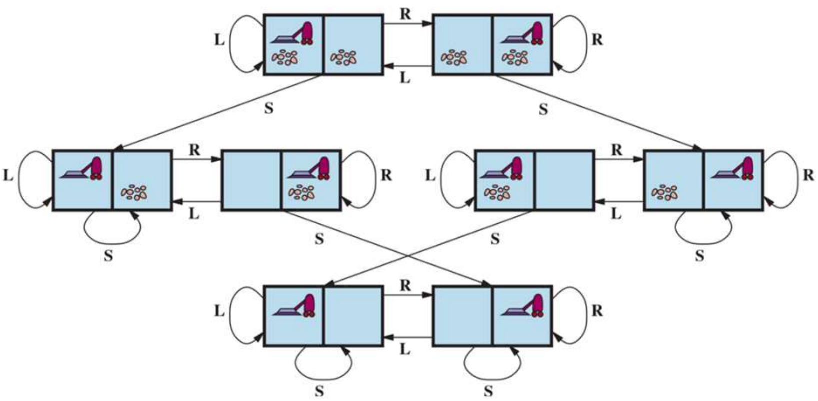


4/10/2024

pra-sâmi

13

Toy Example : Vacuum World



The state space for the vacuum world.
Links denote actions L : Turn Left R : Turn Right S: Suck

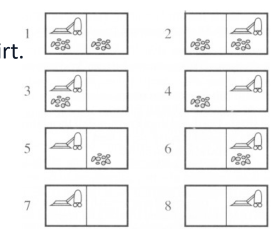
Compared with the real world, this toy problem has discrete locations, discrete dirt, reliable cleaning, and it never gets any dirtier

4/10/2024

pra-sâmi

14

Toy Example : Vacuum World



- States: The state is determined by both the agent location and the dirt locations.
  - The agent could be in one of two locations, each of which might or might not contain dirt.
  - Thus, there are $2 \times 2^2 = 8$ possible world states.
  - A larger environment with n locations has $n \cdot 2^n$ states.
- Initial state: Any state can be designated as the initial state.
- Actions: In this simple environment, each state has just three actions: Left, Right, and Suck. Larger environments might also include Up and Down.
- Transition model: The actions have their expected effects, except that moving Left in the leftmost square, moving Right in the rightmost square, and Sucking in a clean square have no effect.
- Goal test: This checks whether all the squares are clean.
- Path cost: Each step costs 1, so the path cost is the number of steps in the path.

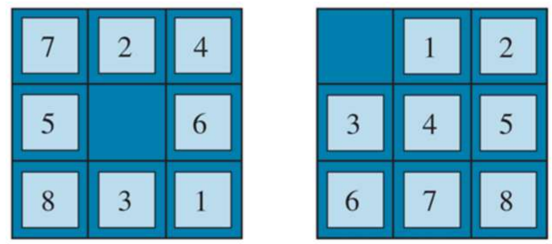
4/10/2024

pra-sâmi

15

Toy Example – 8 Puzzle

- The 8-puzzle consists of a 3×3 board with eight numbered tiles and a blank space
- A tile adjacent to the blank space can slide into the space
- The object is to reach a specified goal state, such as the one shown on the right
- There are total of $\frac{9!}{2}$ configurations possible
- Q: Why do you think its half???

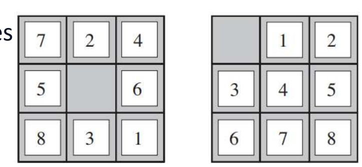


Start State Goal State

16

Toy Example – 8 Puzzle

- States: A state description specifies the location of each of the eight tiles and the blank in one of the nine squares
- Initial state: Any state can be designated as the initial state



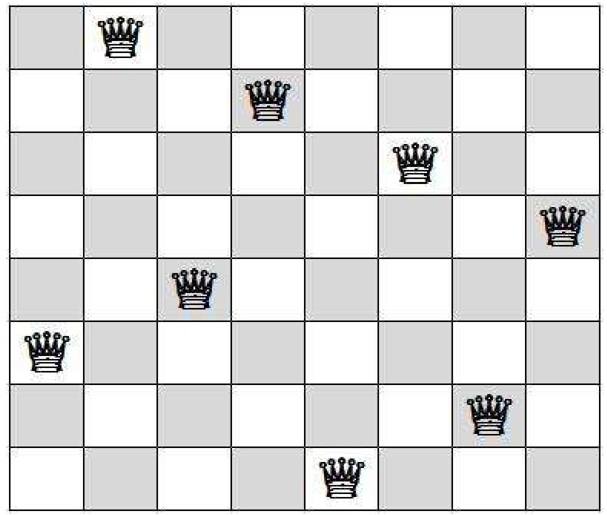
Start State Goal State

- Actions: movements of the **blank space** Left, Right, Up, or Down keeping in mind boundary conditions
- Transition model: Given a state and action, this returns the resulting state
  - E.g.: if we apply Left to the start state in the Figure, the resulting state has the 5 and the blank switched
  - Moves defined as it is applied to blank
- Goal test: This checks whether the state matches the goal configuration
- Path cost: Each step costs 1, so the path cost is the number of steps in the path.

17

Toy Example – 8 Queen Problem

- Goal: place eight queens on a chessboard such that no queen attacks any other
- Incremental formulation : starting with an empty state; each action adds a queen to the state
  - Actions: Add a queen to any empty square.
- States: Any arrangement of 0 to 8 queens on the board is a state.
  - Initial state: No queens on the board.
- Transition model: Returns the board with a queen added to the specified square.



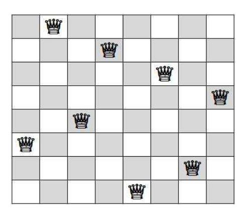
4/10/2024

pra-sâmi

18

Toy Example – 8 Queen Problem

- In this formulation, we have 64 x 63 x 62 ··· 58 x 57 ≈ 1.8×10^{14} possible sequences to investigate.
  - A better formulation would prohibit placing a queen in any square that is already attacked
- States: All possible arrangements of n queens ($0 \le n \le 8$), one per column in the left most n columns, with no queen attacking another.
- Actions: Add a queen to any square in the left most empty SAFE column
  - This formulation reduces the 8-queens state space from 1.8×10^{14} to just 2,057, and solutions are easy to find
- There are simple algorithm that solves even the million-queens problem with ease.



4/10/2024

pra-sâmi

19

Example: Playing Chess

- First have to specify:
  - The starting position of the chess board,
  - The rules that define the legal moves
  - The board positions that represent a win for one side or the other
- The starting position can be described as an 8 by 8 array
  - Each position contains a symbol for appropriate piece
- Define goal - the check mate position
- The legal moves provide the way of getting from initial state to a goal state
- They can be described easily as a set of rules consisting of two parts:
  - A pattern representing the current board position.
  - Another pattern that describes the change to be made to reflect the move

20

Example: Playing Chess

- However, this approach leads to large number of rules & board positions !!
- Using so many rules poses problems such as:
  - No person could ever supply a complete set of such rules
  - No program could easily handle all those rules.
- Even storing so many rules poses serious difficulties

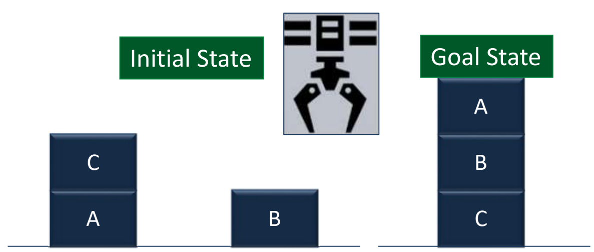
21

Defining Chess Problem as State Space Search

- We need to write the rules describing the legal moves in as general a way as possible.
- For example. White pawn
  - AT Square(file e, rank 2)
  - AND Square(file e, rank 3) is empty
  - AND Square(file e, rank 4) is empty,
  - THEN move the pawn from Square(file e, rank 2) to Square(file e, rank 4)
- In general, the more succinctly we can describe the rules we need, the less work we will have to do to provide them and more efficient the program.

22

Example - Blocks World States



Initial State	Goal State
On (C, A) Clear (Floor) On (A, Floor) Clear (B) On (B, Floor) Clear (C)	On (A, B) On (B, C) On (C, Floor) Clear (A) Clear (Floor)

23

Example - Blocks World States

- What are your predicates

- What are your operation

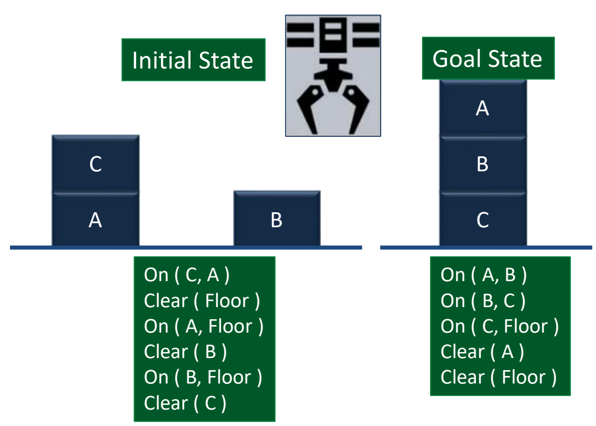
- Domain Predicates
  - On (x, y) – Block x in on block y
  - OnTable (x) – Block x is on table
  - Holding (x) – Grab is holding x
  - Clear(x) – There is nothing on top of x
  - GrabEmpty

- Operations
  - PickUpFromTable (x):
    - Precondition : OnTable (x), Clear(x), GrabEmpty
    - Add : Holding (x)
    - Delete : OnTable(x), GrabEmpty
  - PutDownOnTable (x):
    - Precondition : Holding (x)
    - Add : OnTable(x), GrabEmpty
    - Delete : Holding (x)

24

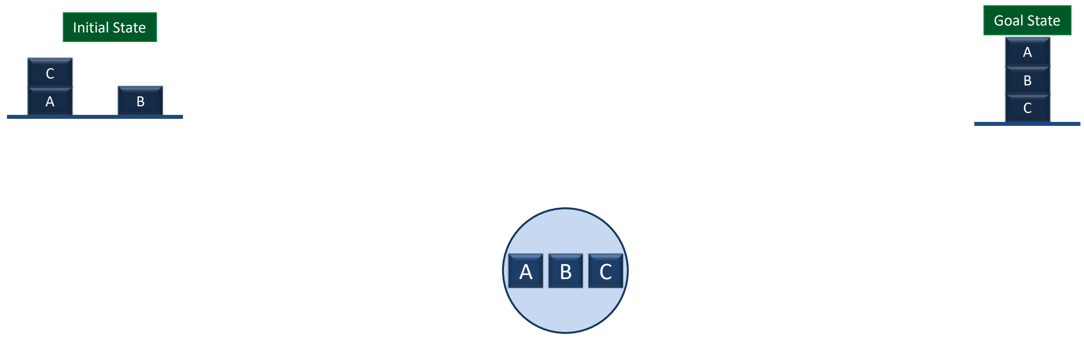
Example - Blocks World States

- Each operator O has the following attributes
  - PC, defines the precondition of the operator
  - D, defines the conditions that will be removed after the operation is executed
  - A, defines the conditions that will be added after the operation is executed
  - C, the cost of the operation



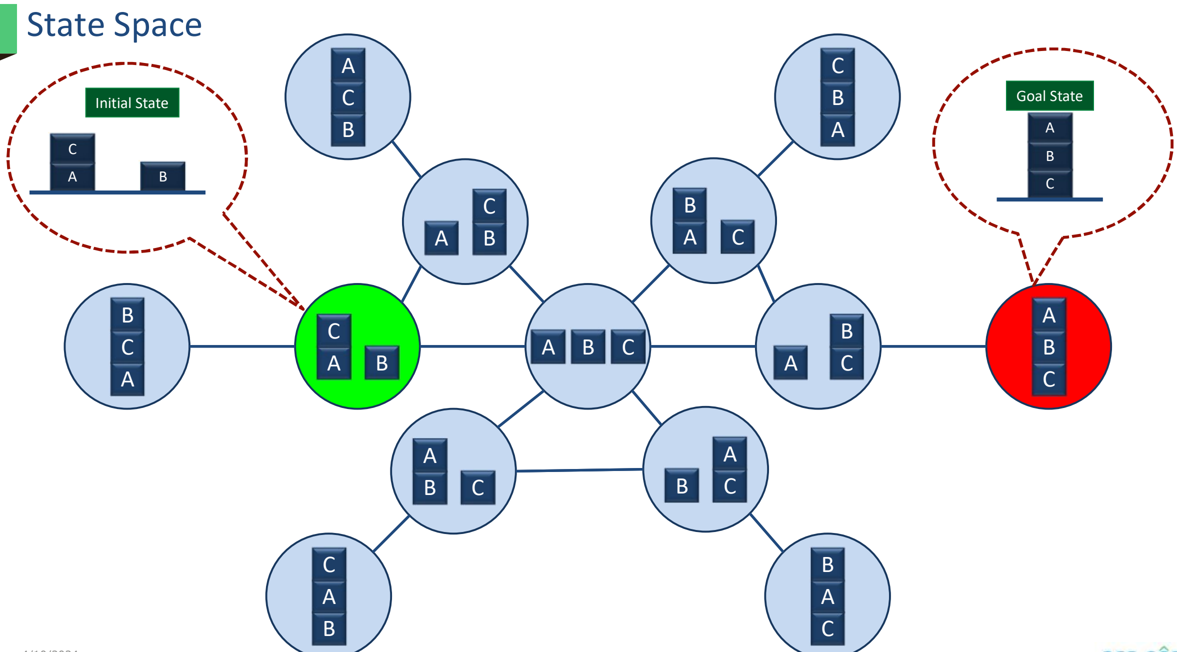
25

State Space



26

State Space



27

Example - Water Jug Problem

- The state space can be described as the set of ordered pairs of integers (x, y)
- Where x represents the number of gallons of water in the 4-gallon jug and y represents the quantity of water in 3-gallon jug
- Such that x = 0, 1, 2, 3 or 4 and y = 0,1,2 or 3
- The start state is (0,0)
- The goal state is (2,n)

28

Production Rules for Water Jug Problem

	Current state	Description	Next State
1	(x, y) and x < 4	Some water in both, Fill the 4 gallon jug	(4,y)
2	(x, y) and y < 3	Some water in both, Fill the 3 gallon jug	(x,3)
3	(x, y) and x > 0	Pour some water (d) out of the 4 gallon jug	(x-d, y)
4	(x, y) and y > 0	Pour some water (d) out of the 3-gallon jug	(x, y-d)
5	(x, y) and x > 0	Empty the 4 gallon jug	(0, y)
6	(x, y) and y > 0	Empty the 3 gallon jug on the ground	(x,0)
7	(x, y) and x + y >= 4 and y > 0	Pour water from the 3 gallon jug into the 4 gallon jug until the 4 gallon jug is full	(4, y - (4 – x))
8	(x, y) and x + y >= 3 and x > 0	Pour water from the 4 gallon jug into the 3 gallon jug until the 3 gallon jug is full	(x - (3 - y), 3)
9	(x, y) and x + y <= 4 and y > 0	Pour all the water from the 3 gallon jug into the 4 gallon jug	(x + y, 0)
10	(x, y) and x + y <= 3 and x > 0	Pour all the water from the 4 gallon jug into the 3 gallon jug	(0, x + y)
11	(0, 2)	Pour the 2 gallons from 3 gallon jug into the 4 gallon jug	(2, 0)
12	(2, y)	Empty out the 2 gallons in the 4 gallon jug	(0, y)

29

Possible Solution

- Match current state and take action
- choice of appropriate mechanism to guide the search for solution.

Gallons in the 4-gallon jug	Gallons in the 3-gallon jug	Rule applied
0	0	2
0	3	9
3	0	2
3	3	7
4	2	5 or 12
0	2	9 or 11
2	0	

pra-sâmi

30

State Space Search

- Define a state space that contains all the possible configurations of the relevant objects
- Specify one or more states within that space that describe possible situations from which the problem solving process may start (initial state)
- Specify one or more states that would be acceptable as solutions to the problem. (goal states)
- Specify a set of rules that describe the actions (operations) available.

pra-sâmi

31

Knowledge

- Knowledge is information about a domain that can be used to solve problems in that domain
  - This knowledge must be represented in the computer
- As part of designing a program, we must define how the knowledge will be represented
- Representation scheme: the form of the knowledge that is used in an agent
- Representation of some piece of knowledge is the internal representation of the knowledge
  - A representation scheme specifies the form of the knowledge
- Knowledge Base: the representation of all of the knowledge that is stored by an agent

4/10/2024

pra-sâmi

32

Knowledge Representation

- Typically, a task is loosely defined
- So is “what constitutes a solution”
  - “Deliver parcels promptly when they arrive”
  - “Something is broken in my house”
  - “Fix whatever is wrong with the electrical system of the house”
- When can I claim that I am done!
- Humans are best at understanding, reasoning, and interpreting knowledge
  - We have common sense
- Humans know things, which is knowledge
  - As per their knowledge they perform various actions in the real world
- In absence of knowledge, even humans will not know how to perform a certain task
- How machines do all these things comes under knowledge representation and reasoning

4/10/2024

pra-sâmi

33

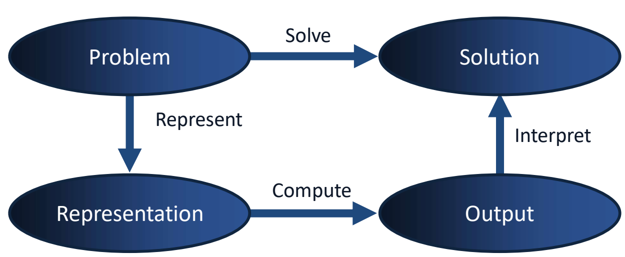
One Important Caveat

- We have chosen to use generalization to represent content and organization of knowledge
- Most aspect of the world is hard to capture in such general terms
- Most generalizations have exceptions
- "tomatoes are red" is a useful rule,
- Some tomatoes are green, yellow, or orange
- Therefore there will be exceptions to all possible rules

34

Knowledge Representation

- To solve a problem, the designer of a system must
  - Flesh out the task and determine what constitutes a solution
  - Represent the problem in a language with which a computer can reason
  - Compute an output, which is an answer presented to a user or a sequence of actions to be carried out in the environment
  - Interpret the output as a solution to the problem



General framework for solving problems

35
Compromise Among Many Competing Objectives

- A representation should be rich
- Should be as close to the problem as possible
  - Compact, natural, and maintainable
  - Relationship between the representation and the domain being represented,
  - Easy to determine whether the knowledge represented is correct
  - A small change in the problem should result in a small change in the representation of the problem
- Should be amenable to efficient computation
  - Able to express features of the problem that can be exploited for computational gain
  - Able to trade off accuracy and computation time
- Should be able to be acquired from people, data and past experiences
  - The data points must be available somewhere

4/10/2024 pra-sâmi

36
Define your goals

- What is a solution to the problem?
- How good must a solution be?
  - Define "good"
- How can the problem be represented?
- What distinctions in the world are needed to solve the problem?
- What specific knowledge about the world is required?

Similar to Feasibility Study and Requirement Analysis in SDLC

4/10/2024 pra-sâmi

37

Define your goals

- How can an agent acquire the knowledge from experts or from experience?
- How can the knowledge be debugged, maintained, and improved?
- How can the agent compute an output that can be interpreted as a solution to the problem?
- Is worst-case performance or average-case performance?
- Is it important for a human to understand how the answer was derived?

38

Defining a Solution - When Can I say I am done!

- Much of software engineering involves refining the specification of the problem
- Problems are not well specified
  - Missing information cannot be assumed
- Example
  - If you ask a travel agent to find out all the information about resorts that may have health issues
  - You do not want the agent to return the information about all resorts, even though all of the information you requested is in the result
  - However, if the trading agent does not have complete knowledge about the resorts, returning all of the information may be the only way for it to guarantee that all of the requested information is there!
- Much work in AI is motivated by common sense reasoning
- We want the computer to be able to make common sense conclusions about the unstated assumptions

39

Built-in Common Sense

Takeout Garbage and only Garbage… Not everything in the house?

pra-sâmi

40

Correct and Complete Answer

- Assume we have some way of defining the problem, let's look at the solution
- Answer returned is incorrect or incomplete?
  - How do you define complete or correct answer?
- There four common classes of solutions
  - Optimal solution
  - Satisficing solution
  - Approximately optimal solution
  - Probable solution

pra-sâmi

41

Optimal Solution

- An optimal solution to a problem is one that is the best solution according to some measure of solution quality
- This measure is typically specified as an ordinal, where only the order matters
- When combining multiple criteria or when reasoning under uncertainty, where the relative magnitudes also matter; we need a cardinal measure,
- An ordinal measure:
  - The robot to take out as much trash as possible
  - The more trash it can take out, the better
  - Leave rest of my stuff alone
- A cardinal measure:
  - Take as much of the trash as possible to the garbage can
  - Minimizing the distance traveled
  - Explicitly specify a trade-off between the effort required and the proportion of the trash taken out
  - It may be better to miss some trash than to waste too much time

42

Satisficing Solution

- Often an agent does not need the best solution to a problem
  - But just needs some solution
- A satisficing solution is one that is good enough according to some description of which solutions are adequate
- For example, a person may tell a robot that it must take all of trash out, or tell it to take out three items of trash
- A mother may tell her son to fix his room
  - Generally it includes clean up the room, make the bed, arrange all fresh cloths, put soiled cloths in laundry basket
  - Ok if he just makes the bed, does not matter if he picks all soiled cloths or not, etc…

43

Approximately Optimal Solution

- One of the advantages of a cardinal measure of success is that it allows for approximations
- An approximately optimal solution - measure of quality is close to the "*best theoretical value*"
- Typically agents do not need optimal solutions to problems; they only must get close enough
- For example, the robot may not need to travel the optimal distance to take out the trash but may only need to be within, say, 10% of the optimal distance
- For most problems - Much easier computationally to get an approximately optimal solution than an optimal solution
- However, for other problems, it is (asymptotically) just as difficult to guarantee finding an approximately optimal solution as it is to guarantee finding an optimal solution
- Some approximation algorithms guarantee that a solution is within some range of optimal, but for some algorithms no guarantees are available

44

Probable solution

- A probable solution: even though it may not actually be a solution, is likely to be a solution
- This is one way to approximate, in a precise manner, a satisficing solution
- For example, in the case where the delivery robot could drop the trash or fail to pick it up when it attempts to, you may need the robot to be 80% sure that it has picked up items of trash

The "Good-Bad-Ugly" part of the Confusion Matrix

- Often you want to distinguish the false positive error rate (the proportion of the answers given by the computer that are not correct) from the false-negative error rate (which is the proportion of those answers not given by the computer that are indeed correct)
- Some applications are much more tolerant of one of these errors than of the other
- While testing a medical patient, you may want to be little more sensitive to false-negative than to false-positive

45

Representation

pra-sâmi

46

What to Represent

- Object:
  - Concept is similar to what we have learned in Object Oriented Programming – Class, Objects
  - Facts about objects in our world domain
  - E.g., Guitars contains strings, trumpets are brass instruments, etc.
- Events:
  - Events are the actions which occur in our world
- Performance:
  - Behavior which involves knowledge about how to do things
- Meta-knowledge:
  - It is knowledge about what we know
- Facts:
  - Facts are the truths about the real world and what we represent
- Knowledge-Base:
  - Group of sentences
- Fluent:
  - A condition that can change over time
  - Needs an argument that depends on time.
  - For example, the condition “the box is on the table”, if it can change over time, an third argument is necessary to specify the time:
    - means that the box is on the table at time .

pra-sâmi

47

Objects

- The organization of objects into categories is a vital part of knowledge representation
- Interaction with the world takes place at the level of individual objects
- Much reasoning takes place at the level of categories
- A shopper may be keen to buy a shirt rather than Peter-England size 40, white color, shirt kept in aisle 3, rack 2, 4th row
- Categories also serve to make predictions about objects once they are classified
- Categories serve to organize and simplify the knowledge base through inheritance
  - All instances of the category Food are edible
  - Fruit is a subclass of Food
  - Apples is a subclass of Fruit

Every apple is edible

4/10/2024

pra-sâmi

48

Events

- **Situation calculus** represents actions and their effects
  - Bathtub is empty → start filling → tub is full
  - What happens in-between
  - Can't even describe two actions simultaneously
  - What else is going on during action
- **Event calculus** helps us in describing what else is going on
  - Consider fluent that a traveller Mohan is at Mumbai
    - At(Mohan, BOM)
  - Test if it is true
    - T(At(Mohan, Mumbai) , True)
  - We can define event E1 of Mohan taking a flight as
    - E1 ∈ Flying and Flyer (E1, Mohan) and Origin(E1, BOM) and Destination(E1, DEL)
  - If it too long, define an alternative three-argument version of the category of flying events
    - E1 ∈ Flying (Mohan , BOM, DEL)

4/10/2024

pra-sâmi

49

Event Calculus – one version

- T(f, t): Fluent f is true at time t
- Happens(e, i): Event e happens over the time interval i
- Initiates(e, f, t): Event e causes fluent f to start at time t
- Terminates(e, f, t): Event e causes fluent f to cease at time t
- Clipped(f, i): Fluent f ceases to be true at some point during time interval i
- Restored(f, i): Fluent f becomes true sometime during time interval i

pra-sâmi

50

Performance

- The events we have seen so far are what we call discrete events—they have a definite structure
- Mohan's trip has beginning, middle and end
- How to capture something which happens during performance
  - Waiting for inflight dining service
- The category of events denoted by Flying has another dimension if we subdivide the process
  - Wait for inflight service to start, that event is still a member of Flying
  - And is true for any subinterval

pra-sâmi

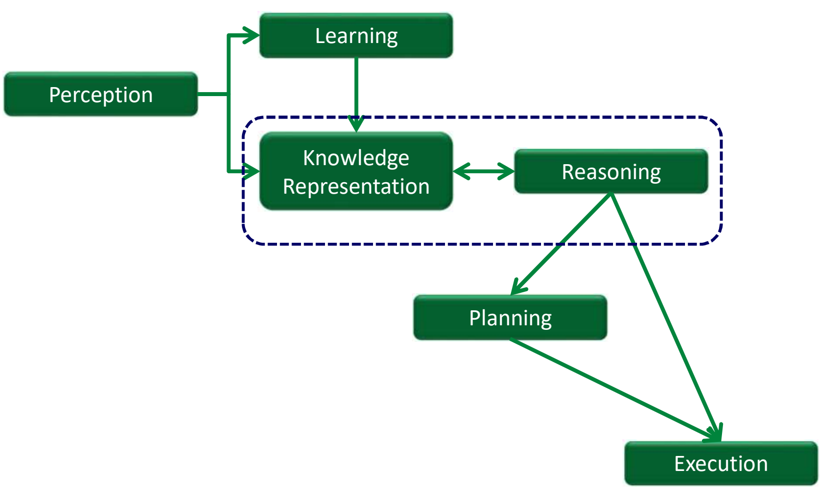
51

Types of knowledge

- Declarative Knowledge:
  - Declarative knowledge is to know about something
  - It includes concepts, facts, and objects.
  - It is also called descriptive knowledge and expressed in declarative sentences
  - It is simpler than procedural language
- Procedural Knowledge
  - It is also known as imperative knowledge.
  - Procedural knowledge is a type of knowledge which is responsible for knowing how to do something
  - It can be directly applied to any task
  - It includes rules, strategies, procedures, agendas, etc.
  - Procedural knowledge depends on the task on which it can be applied
- Meta-knowledge:
  - Knowledge about the knowledge is called Meta-knowledge
- Heuristic knowledge:
  - Heuristic knowledge is representing knowledge of some experts in a field or subject
  - Heuristic knowledge is rules of thumb based on previous experiences, awareness of approaches, and which are good to work but not guaranteed
- Structural knowledge:
  - Structural knowledge is basic knowledge to problem-solving
  - It describes relationships between various concepts such as kind of, part of, and grouping of something
  - It describes the relationship that exists between concepts or objects

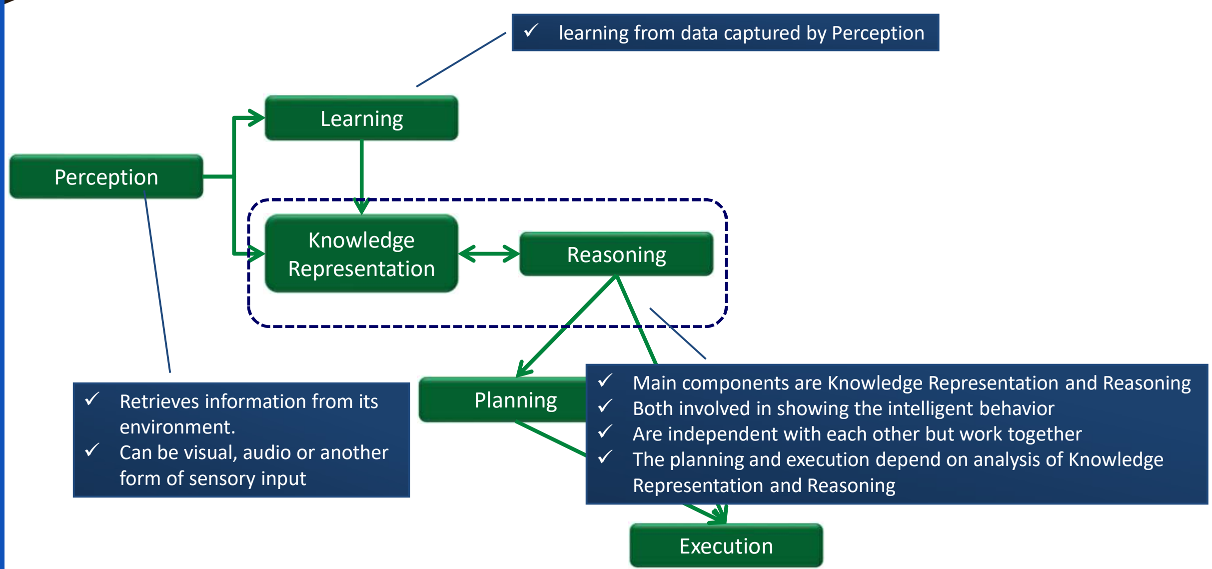
52

AI Knowledge Cycle



53

AI Knowledge Cycle



55

Reflect…

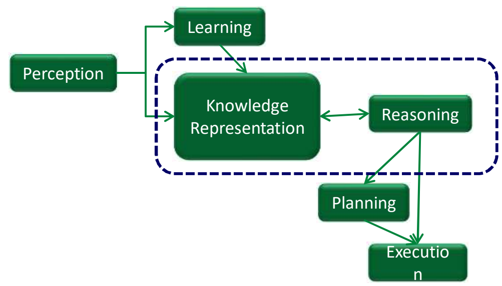
- Knowledge and its representation to create system for efficient search
- Agent senses the world around it and create notion of a world in memory
- We want to model world in abstract terms
  - **Ontological engineering :** representing in abstract concepts
- Defining goals is as important as understanding the problem
- Unambiguous definition of solution is a must
- There four common classes of solutions
  - Optimal solution
  - Satisficing solution
  - Approximately optimal solution
  - Probable solution
- Special-purpose representation systems, such as semantic networks and description logics, have been devised to help in organizing a hierarchy of categories.
- Inheritance is an important form of inference, allowing the properties of objects to be deduced from their membership in categories

57

Reflect

- What to represent in Knowledge
  - Objects
  - Events
  - Performance or execution of events
  - Meta-knowledge
  - Facts
  - Knowledge base
  - Fluent

- AI Knowledge Cycle



pra-sâmi

58

Agenda

- Goal Formulation by Example
- Search by example : Toy and real world Problems
- Type of Search
- Informed search/ Heuristic Search

pra-sâmi

59

pra-sâmi