CSCE 590-1: From Data to Decisions with Open Data: A Practical Introduction to AI

Lecture 15: Reasoning

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

23RD FEB, 2021

Carolinian Creed: "I will practice personal and academic integrity."

Organization of Lecture 15

- Introduction Segment
  - Recap/ Discussion of Lecture 14
  - Mid-sem review of Project (Mar 2)

- Main Segment
  - Reasoning: drawing insights from knowledge (known or learnt)
  - Propositional Logic
  - Problem-solving agent
    - Deterministic
    - Search

- Concluding Segment
  - About Next Lecture – Lecture 16
  - Ask me anything

Introduction Segment

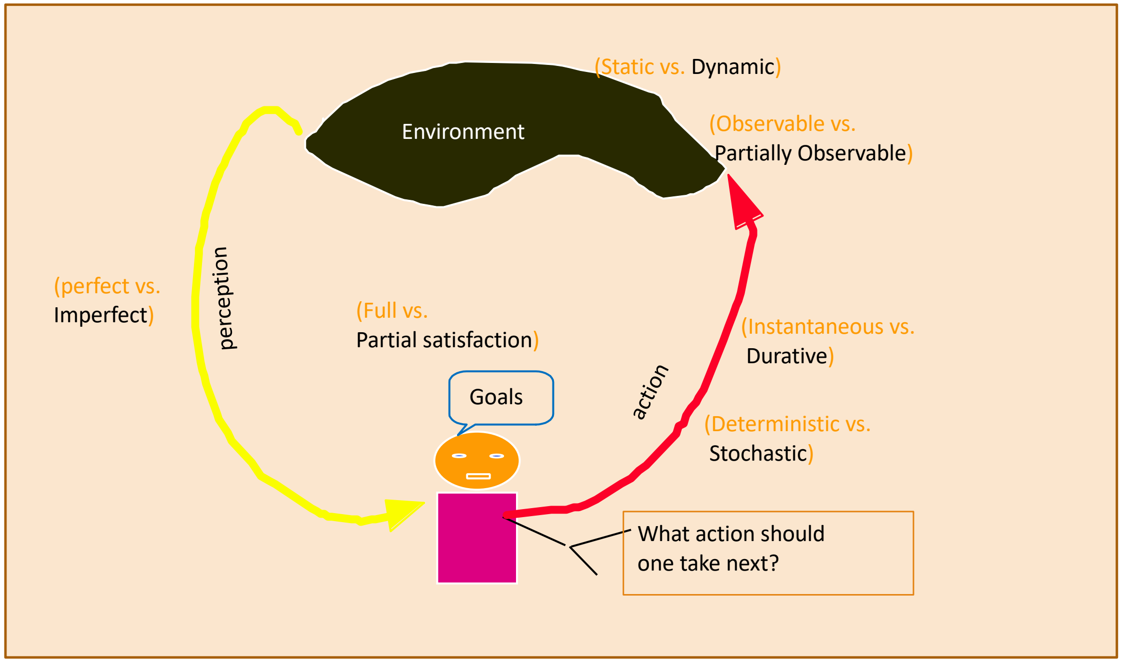
Recap of Lecture 14

- Guest talk on timing errors by Sandeep Sandha, UCLA
- We heard about timing errors and their impact on devices
- Implications for their usage in detecting human activities and applications built using them
- Touched on other Computer Science areas
  - Distributed systems, software engineering, networking – specifically
  - AI on edge devices

Mid-sem review of Project (Mar 2)

- Instructor met students to review projects
  - Project outline should be clear and actionable, relevant to the problem
  - Should increase probability of doing well in course project
- Student responsibilities
  - Work as per plan
  - Flag issues early and discuss alternative
  - Track progress
- Next milestone
  - In-class update: March 25, 2021

Main Segment



<u>Recap</u>: AI as Intelligent Agent

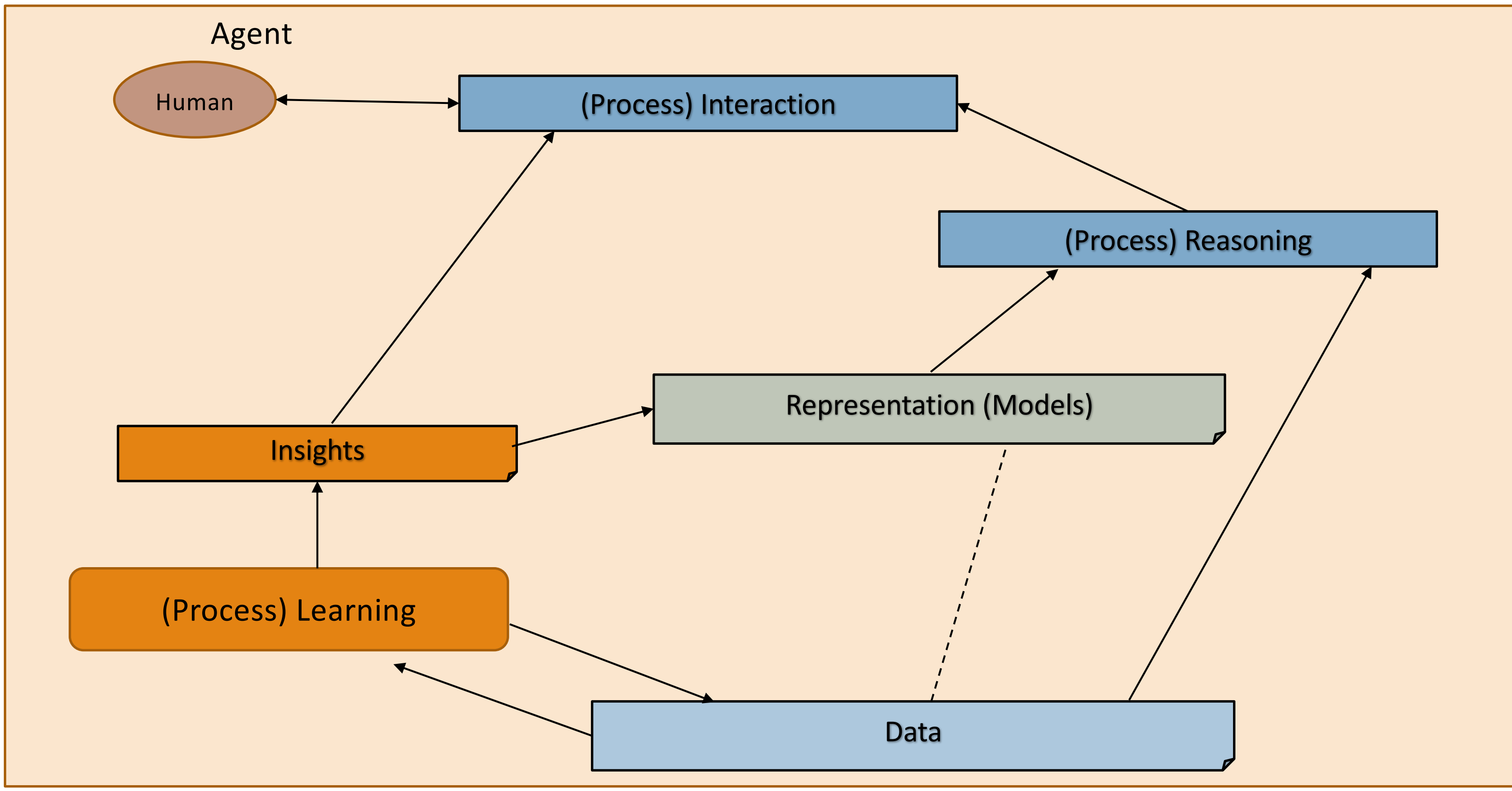
From Subbarao Kambhampati's AI Planning Course

Examples of Agents

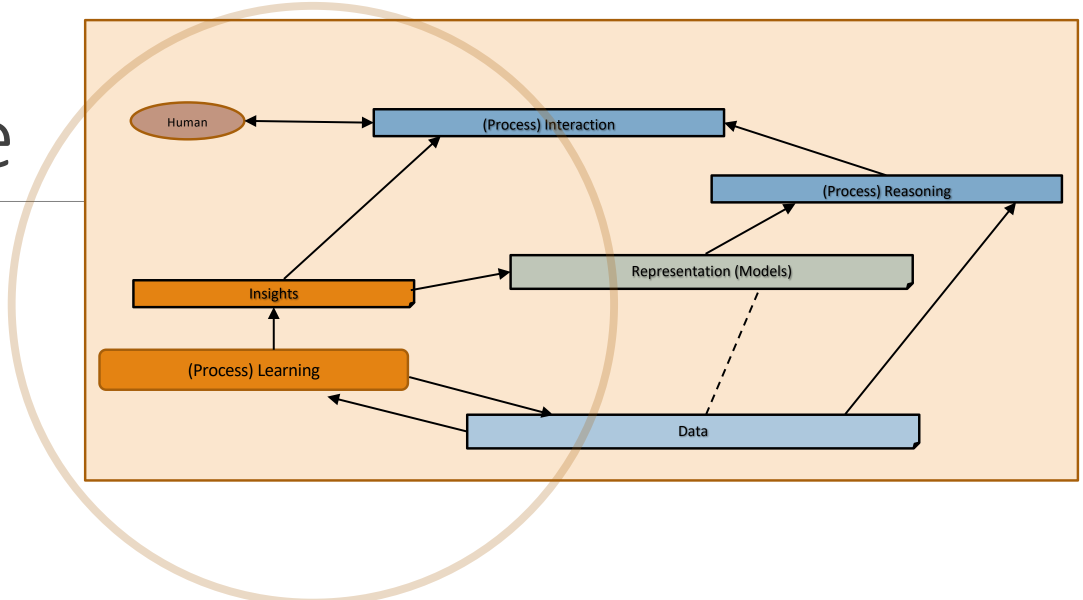
Agent Type	Percepts	Actions	Goals	Environment
Medical diagnosis system	Symptoms, findings, patient's answers	Questions, tests, treatments	Healthy patient, minimize costs	Patient, hospital
Satellite image analysis system	Pixels of varying intensity, color	Print a categorization of scene	Correct categorization	Images from orbiting satellite
Part-picking robot	Pixels of varying intensity	Pick up parts and sort into bins	Place parts in correct bins	Conveyor belt with parts
Refinery controller	Temperature, pressure readings	Open, close valves; adjust temperature	Maximize purity, yield, safety	Refinery
Interactive English tutor	Typed words	Print exercises, suggestions, corrections	Maximize student's score on test	Set of students

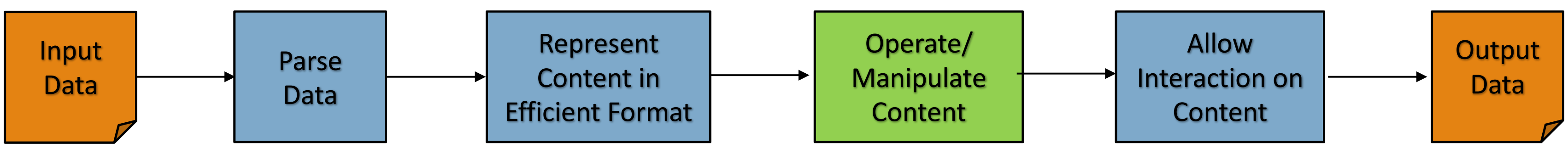
Source: Russell & Norvig, AI: A Modern Approach

Relationship Between AI Processes



Data Analysis Pipeline





Reasoning and Its Role in AI

- Reasoning: drawing insights from knowledge (known or learnt)
- Help various settings for an Intelligent Agent
  - Deterministic
  - Uncertain knowledge
  - Optimal decisions

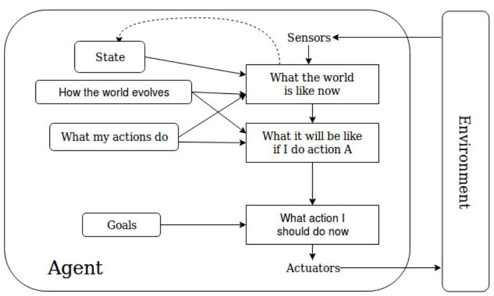


Figure Source: Russell & Norvig, AI: A Modern Approach

Example Situation – Course Selection

- A person wants to pass an academic program in two majors: A and B
- There are three subjects: A, B and C, each with three levels (*1, *2, *3). There are thus 9 courses: A1, A2, A3, B1, B2, B3, C1, C2, C3
- To graduate, at least one course at beginner (*1) level is needed in major(s) of choice(s), and two courses at intermediate levels (*2) are needed
- Answer questions
  - Q1: How many minimum courses does the person have to take ?
  - Q2: Can a person graduate in 2 majors studying 3 courses only?
  - …

How to Tackle Such Situations ?

- Represent and reason with the world
  - Perform Q/A

- Solve problems about courses selection scenarios

Representation - Example

- Domain Description: "There are three subjects: A, B and C, each with three levels (*1, *2, *3)."
- Representation
  - has_studied_courseA1: yes – student has taken course; no – student has not taken
  - has_studied_courseA2
  - has_studied_courseA3
  - has_studied_courseB1
  - has_studied_courseB2
  - has_studied_courseB3
  - has_studied_courseC1
  - has_studied_courseC2
  - has_studied_courseC3

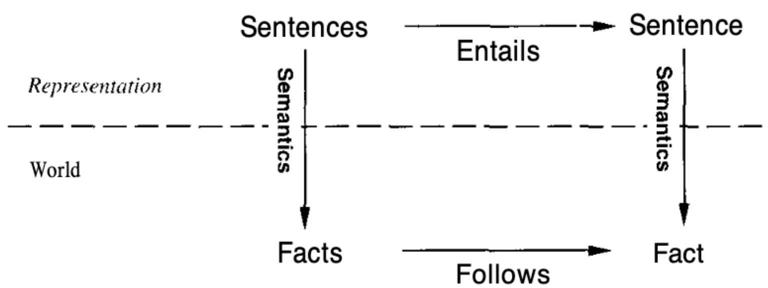
Inference Procedure

- Given a knowledge base (KB), generate new sentences α that are entailed by KB
  - KB $\models \alpha$

- Given KB and α, report whether or not α is entailed by KB
  - KB $\vdash \alpha$

Formal Logic – 1/3

- An automaton for manipulating symbols and drawing conclusions
- Consists of a knowledge base with:
  - a set of true statements (sentences). Sentences have
    - Syntax
    - Semantics – compositional property
  - Proof theory: a set of rules for deducing the entailments / interpretations of the sentences



- Properties of sentences
  - **Valid**: A sentence is **valid** or necessarily true if and only if it is true under all possible interpretations in all possible worlds. Also called a **tautology**
  - **Satisfiable:** A sentence is satisfiable if and only if there is some interpretations in some possible worlds where it is true.

Credits:
- Russell & Norvig, AI - A Modern Approach
- Deepak Khemani - A First Course in AI

Representation - Example

- Domain Description: “There are three subjects: A, B and C, each with three levels (*1, *2, *3).”
- Representation
  - has_studied_courseA1: yes – student has taken course; no – student has not taken
  - has_studied_courseA2
  - has_studied_courseA3
  - has_studied_courseB1
  - has_studied_courseB2
  - has_studied_courseB3
  - has_studied_courseC1
  - has_studied_courseC2
  - has_studied_courseC3

LowerThan_Course_A1_CourseA2
LowerThan_Course_A2_CourseA3
LowerThan_Course_B1_CourseB2
LowerThan_Course_B2_CourseB3
LowerThan_Course_C1_CourseC2
LowerThan_Course_AC_CourseC3

- Previous statements set did not capture hierarchy between levels; new sentences would not have followed the reality in the world. Need more statements – LowerThan as shown.

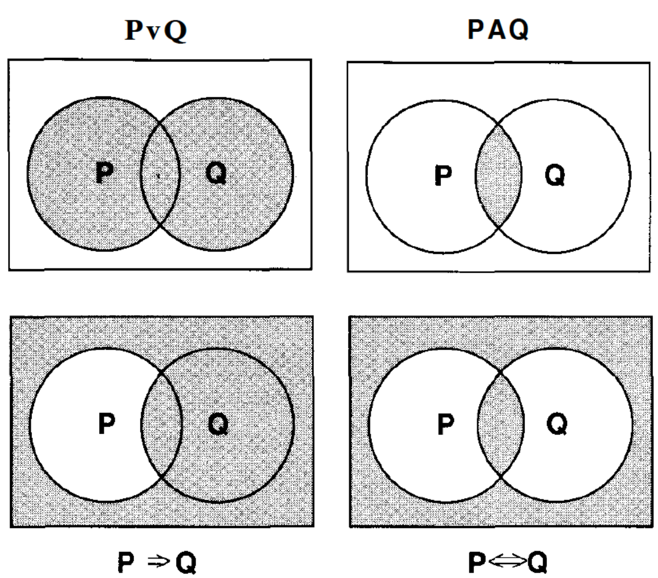
Propositional Logic - Syntax

Sentence	—	*AtomicSentence* *ComplexSentence*
AtomicSentence	—	***True*** | ***False***
	|	*P* *Q* *R* ...
ComplexSentence	—	(*Sentence*)
	|	*Sentence Connective Sentence*
	|	¬*Sentence*
Connective	—	A | V | ⇔ | ⇒

BNF (Backus-Naur Form) grammar of sentences in propositional logic

Source: Russell & Norvig, AI: A Modern Approach

Propositional Logic - Semantics



Model of sentence: Any world in which a sentence is true (under a particular interpretation)

α	β	γ	$\alpha \vee \beta$	$\neg\beta \vee \gamma$	$\alpha \vee \gamma$
False	*False*	*False*	*False*	*True*	*False*
False	*False*	*True*	*False*	*True*	*True*
False	*True*	*False*	*True*	*False*	*False*
False	*True*	*True*	*True*	*True*	*True*
True	*False*	*False*	*True*	*True*	*True*
True	*False*	*True*	*True*	*True*	*True*
True	*True*	*False*	*True*	*False*	*True*
True	*True*	*True*	*True*	*True*	*True*

Truth table to prove soundness of inference

Source: Russell & Norvig, AI: A Modern Approach

Propositional Logic

Inference Procedures

◇ **Modus Ponens or Implication-Elimination:** (From an implication and the premise of the implication, you can infer the conclusion.)

$$\frac{a \Rightarrow \beta, \quad a}{\beta}$$

◇ **And-Elimination**: (From a conjunction, you can infer any of the conjuncts.)

$$\frac{\alpha_1 \text{ A } \alpha_2 \text{ A} \ldots \text{A } \alpha_n}{\alpha_i}$$

◇ **And-Introduction**: (From a list of sentences, you can infer their conjunction.)

$$\frac{\alpha_1, \alpha_2, \bullet\bullet\bullet, \alpha_n}{\alpha_1 \text{ A } \alpha_2 \text{ A} \ldots \text{A } \alpha_n}$$

0 **Or-Introduction**: (From a sentence, you can infer its disjunction with anything else at all.)

$$\frac{\alpha_i}{\alpha_1 \text{ V } \alpha_2 \text{ V} \ldots \text{V } \alpha_n}$$

◇ **Double-Negation Elimination:** (From a doubly negated sentence, you can infer a positive sentence.)

$$\frac{\neg\neg \text{a}}{\alpha}$$

◇ **Unit Resolution:** (From a disjunction, if one of the disjuncts is false, then you can infer the other one is true.)

$$\frac{a \text{ V } \beta, \quad \neg\beta}{a}$$

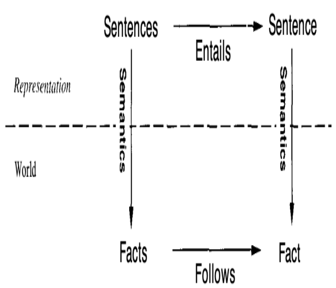
◇ **Resolution:** (This is the most difficult. Because *0* cannot be both true and false, one of the other disjuncts must be true in one of the premises. Or equivalently, implication is transitive.)

$$\frac{a \vee \beta, \quad \neg\beta \text{ V } 7}{a \vee \gamma} \quad \text{or equivalently} \quad \frac{\neg\alpha \Rightarrow \beta, \quad \beta \Rightarrow \gamma}{\neg\alpha \Rightarrow \gamma}$$

Source: Russell & Norvig, AI: A Modern Approach

Formal Logic – 2/3

- Levels at which sentences are encoded
  - Epistemic (also called knowledge): what agents knows or believes
  - Logical: how sentences are encoded to allow inferencing. E.g., symbols
  - Executional: how sentences are encoded during execution. E.g., vectors, symbols

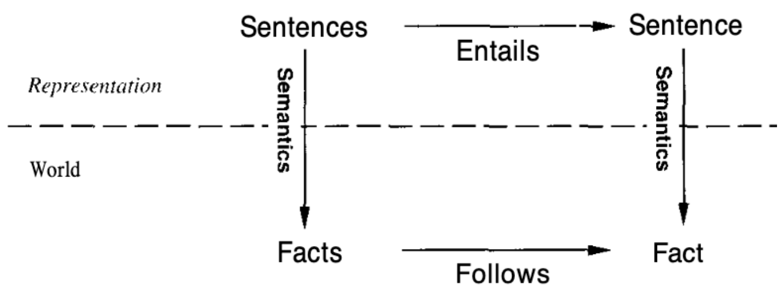


- Properties of sentences
  - **Valid**: A sentence is **valid** or necessarily true if and only if it is true under all possible interpretations in all possible worlds. Also called a **tautology**
  - **Satisfiable:** A sentence is satisfiable if and only if there is some interpretations in some possible worlds where it is true.

Credits:
- Russell & Norvig, AI - A Modern Approach
- Deepak Khemani - A First Course in AI

Formal Logic – 3/3

- Properties of Logic System
  - **Soundness**: if it produces only true statements
  - **Completeness**: if it produces all true statements
  - **Consistency**: if it does not produce a sentence and its negation



Language	Ontological Commitment (What exists in the world)	Epistemological Commitment (What an agent believes about facts)
Propositional logic	facts	true/false/unknown
First-order logic	facts, objects, relations	true/false/unknown
Temporal logic	facts, objects, relations, times	true/false/unknown
Probability theory	facts	degree of belief 0…1
Fuzzy logic	degree of truth	degree of belief 0…1

Credits:
- Russell & Norvig, AI - A Modern Approach
- Deepak Khemani - A First Course in AI

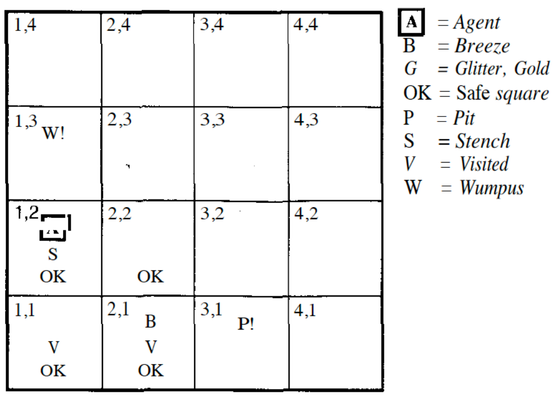
Major Types of Reasoning

- Inference: From premises to conclusions
  - Major types
    - **Deduction**: deriving logical conclusions from premises known or assumed to be true
    - **Induction**: deriving from particular premises to a universal conclusion.
    - **Abduction**: from an observation, find the most likely conclusion from the observations

- Usage
  - Deduction is useful to build knowledge bases from parts
  - Induction: to generalize
  - Abduction is a good source for hypothesis / priors in Bayesian learning

Exercise and Code

- Logical Reasoning
  - From Book: AI – A Modern Approach, https://github.com/aimacode/aima-python/blob/master/logic.ipynb



Source: Russell & Norvig, AI: A Modern Approach

Problem Solving Agents

Goal-directed agents

1. **Goal Formulation: Set of one or more (desirable) world states (e.g. majors in).**
2. **Problem formulation: What actions and states to consider given a goal and an initial state.**
3. **Search for solution: Given the problem, search for a solution --- *a sequence of actions to achieve the goal starting from the initial state.***
4. **Execution of the solution**

Source: Bart Selman's CS 4700 Course

Goal:
Reach Bucharest (Romania)

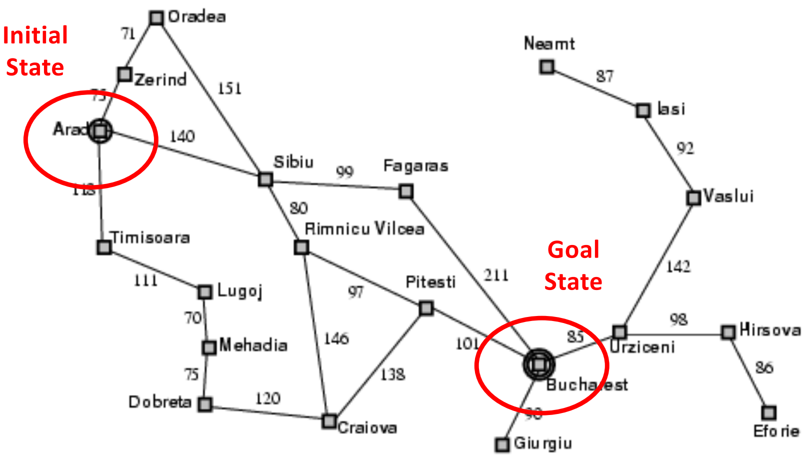
Problem:
action: drive between pair of connected cities (direct road)

state: be in a city
(20 world states)

Solution:
sequence of cities leading from start to goal state, e.g., Arad, Sibiu, Fagaras, Bucharest

Execution
drive from Arad to Bucharest according to the solution

Example: Path Finding Problem

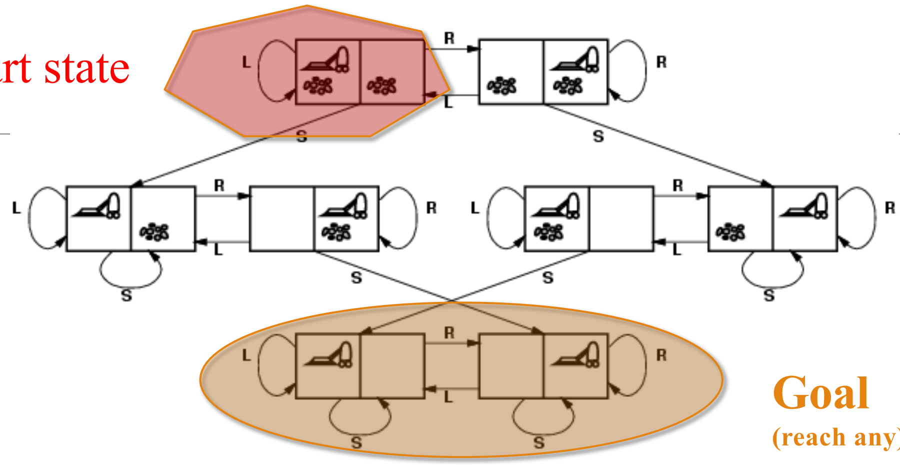


Environment: fully observable (map), deterministic, and the agent knows effects of each action.

Source: Bart Selman's CS 4700 Course

Example: Vacuum World State Space Graph

Start state



Goal
(reach any)

States: The agent is in one of 8 possible world states.

Actions: *Left, Right, Suck [simplified: left out No-op]*

goal test: No dirt at all locations

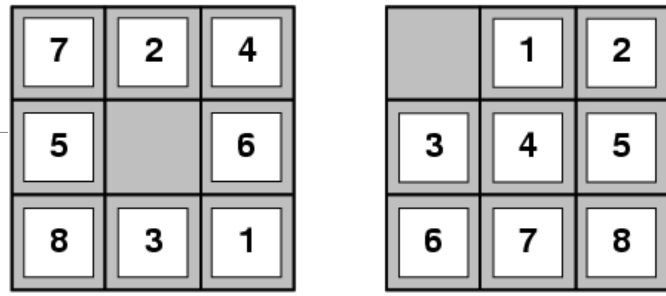
path cost: 1 per action

Minimum path from Start to Goal state: **3 actions**

Alternative, longer plan: **4 actions**

Source: Bart Selman's CS 4700 Course

Example: The 8-puzzle"sliding tile puzzle"



Start State   Goal State

States: the boards, i.e., locations of tiles

Actions: move blank left, right, up, down

goal test: goal state (given; tiles in order)

Path cost: 1 per move

Note: finding optimal solution of *n*-puzzle family is NP-hard! Also, from certain states you can't reach the goal. Total number of states 9! = 362,880

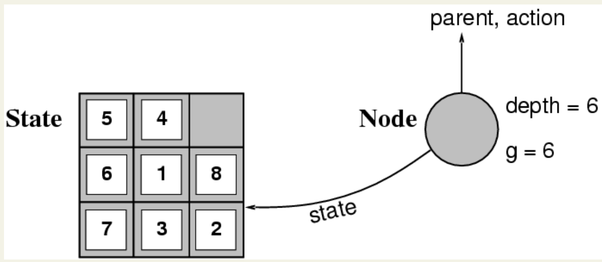
Source: Bart Selman's CS 4700 Course

Search Techniques

Search Concepts: States and Nodes

A state is a representation of a physical configuration.

A node is a data structure constituting part of a search tree includes state, tree parent node, action (applied to parent), path cost (initial state to node) *g(x)*, depth



The `Expand` function creates new nodes, filling in the various fields and using the `SuccessorFn` of the problem to create the corresponding states.

Fringe is the collection of nodes that have been generated but not (yet) expanded. Each node of the fringe is a leaf node.

Source: Bart Selman's CS 4700 Course

Tree-search Algorithms

Basic idea: simulated exploration of state space by generating successors of already-explored states (a.k.a. ~ expanding states)

```
function TREE-SEARCH( problem, strategy) returns a solution, or failure
   initialize the search tree using the initial state of problem
   loop do
       if there are no candidates for expansion then return failure
       choose a leaf node for expansion according to strategy
       if the node contains a goal state then return the corresponding solution
       else expand the node and add the resulting nodes to the search tree
```

Source: Bart Selman's CS 4700 Course

Implementation: General Tree Search

```
function TREE-SEARCH( problem, fringe) returns a solution, or failure
   fringe ← INSERT(MAKE-NODE(INITIAL-STATE[problem]), fringe)
   loop do
       if fringe is empty then return failure
       node ← REMOVE-FRONT(fringe)
       if GOAL-TEST[problem](STATE[node]) then return SOLUTION(node)
       fringe ← INSERTALL(EXPAND(node, problem), fringe)
```

```
function EXPAND( node, problem) returns a set of nodes
   successors ← the empty set
   for each action, result in SUCCESSOR-FN[problem](STATE[node]) do
       s ← a new NODE
       PARENT-NODE[s] ← node;   ACTION[s] ← action;   STATE[s] ← result
       PATH-COST[s] ← PATH-COST[node] + STEP-COST(node, action, s)
       DEPTH[s] ← DEPTH[node] + 1
       add s to successors
   return successors
```

A search strategy is defined by picking the order of node expansion.

Uninformed Search Strategies

Uninformed (blind) search strategies use only the information available in the problem definition:

- Breadth-first search
- Uniform-cost search
- Depth-first search
- Depth-limited search
- Iterative deepening search
- Bidirectional search

Key issue: type of queue used for the fringe of the search tree (collection of tree nodes that have been generated but not yet expanded)

Source: Bart Selman's CS 4700 Course

Search Strategies

- A search strategy is defined by picking the **order of node expansion.**

- Strategies are evaluated along the following dimensions:
  - **completeness**: does it always find a solution if one exists?
  - **time complexity**: number of nodes generated
  - **space complexity**: maximum number of nodes in memory
  - **optimality**: does it always find a least-cost solution?

- Time and space complexity are measured in terms of
  - *b:* maximum branching factor of the search tree
  - *d:* depth of the least-cost solution
  - *m*: maximum depth of the state space (may be ∞)

Source: Bart Selman's CS 4700 Course

Exercise and Code

- Search Methods
  - From Book: AI – A Modern Approach, https://github.com/aimacode/aima-python/blob/master/search.ipynb

Source: Russell & Norvig, AI: A Modern Approach

Goal-Based Agents
Generating Sequence of Actions

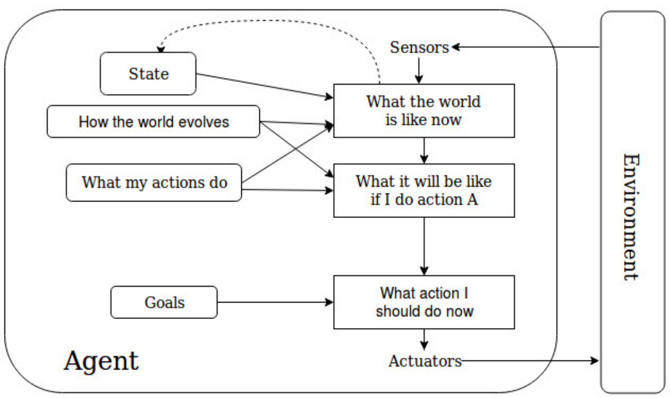
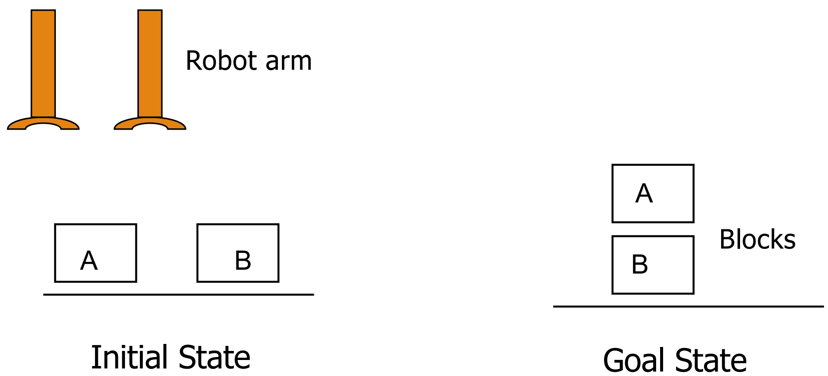


Figure Source: Russell & Norvig, AI: A Modern Approach

Reasoning Illustration - Planning Example

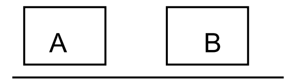
Blocks World



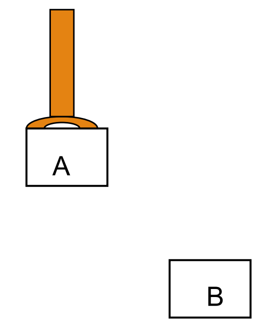
All robots are equivalent

Reasoning Illustration - Representation

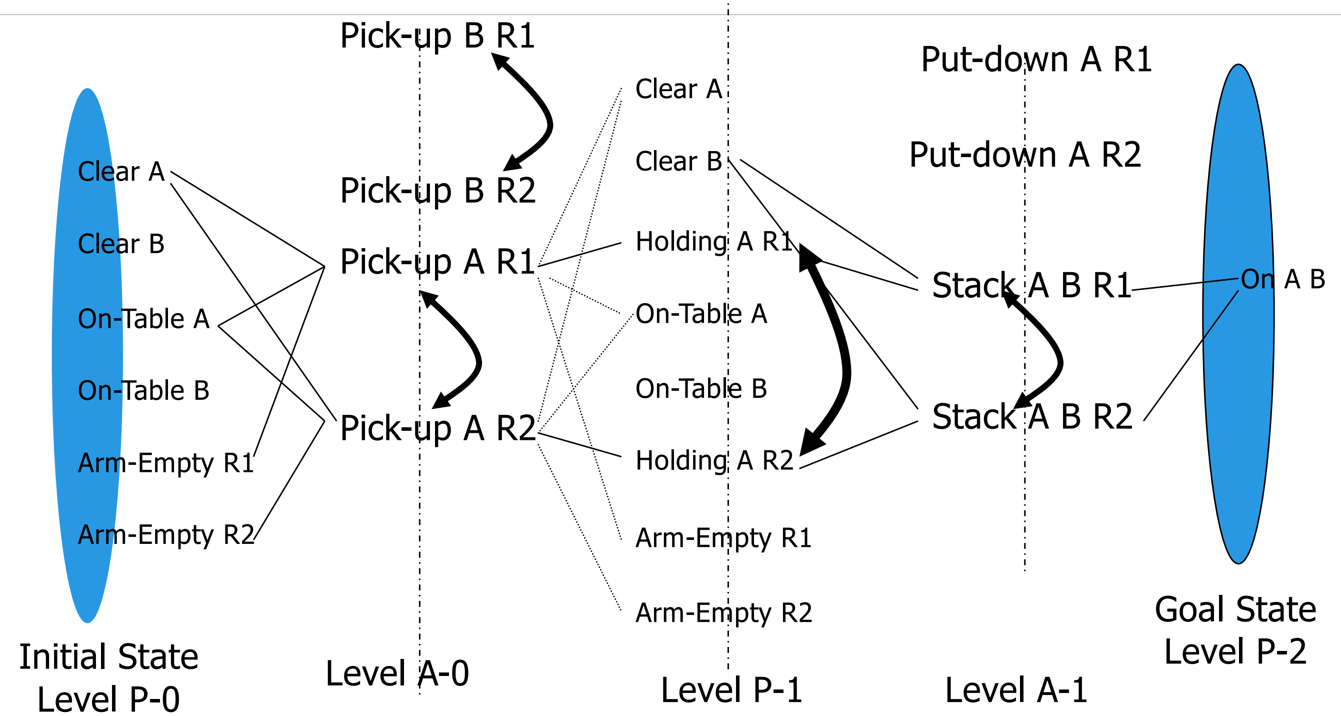
States: ((On-Table A) (On-Table B) …)



Actions: ((Name: (Pickup ?block ?robot)
Precondition: ((Clear ?block)
(Arm-Empty ?robot)
(On-Table ?block))
Add: ((Holding ?block ?robot))
Delete: ((Clear ?block)
(Arm-Empty ?robot)))…)



Reasoning Illustration - Planning Process



Active Area of Research

Considerations

- What to find:
  - Any workable plan
  - Optimal plan – but then what is the criteria
  - All plans
  - Diverse plans
- How to find
  - Plan at the end
  - Plan anytime
- How to represent problem
- How to explain solution

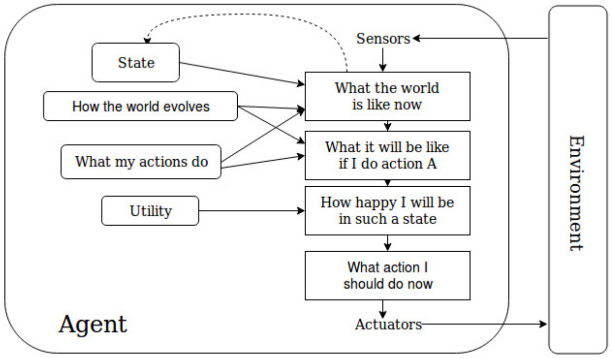
Lecture 15: Concluding Comments

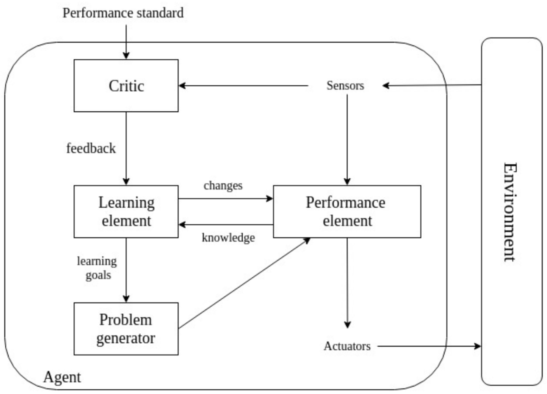
- We revisited what it means to have an AI system as Intelligent Agent
- Understood knowledge representation and relationship to facts in the world
- Discussed logic procedures to draw inferences
- Discussed search methods to solve problems
- Saw a planning problem – generating sequence of actions to meet a goal

Concluding Segment

Exercise – Intelligent Agent Types

- Search Methods
  - From Book: AI – A Modern Approach, https://github.com/aimacode/aima-python/blob/master/vacuum_world.ipynb





Source: Russell & Norvig, AI: A Modern Approach

Adjustment to Upcoming Schedule

15	Mar 4 (Th)	**Reasoning and Search**	**Semester - Midpoint**
16	Mar 9 (Tu)	Agent – Optimization	
17	Mar 11 (Th)	Agent – Handling Uncertain World	
18	Mar 16 (Tu)	Agent – Learning	
19	Mar 18 (Th)	Text: Data Prep (NLP)	Quiz 3
20	Mar 23 (Tu)	Text: Analysis - Supervised (NLP)_	
21	Mar 25 (Th)	Review, Paper presentations, Discussion	
22	Mar 30 (Tu)	Text: Advanced – Summarization, Sentiment	
23	Apr 1 (Th)	Text: Visualization, Explanation	
24	Apr 6 (Tu)	Multimodal Agents: Structured+Text: Examples	
25	Apr 8 (Th)	Case Study 1: Water	Quiz 4
26	Apr 13 (Tu)	Case Study 2: Finance	

Focus on Integrated Agent Behavior (Lectures 17, 18)

Reduce focus on Case-studies (1 per domain)

About Next Lecture – Lecture 16

Lecture 16: Reasoning, Agents Continued

- Reasoning – advanced settings
  - Agents Making Optimal Decisions
- What is an Optimal Decision ?
- Some methods