# Lesson 1: Intro to Knowledge-Based AI

**Fundamental Conundrums of AI**

1. Intelligent agents have limited resources – processing speed, memory – how AI agents can give near real time performance on many problems?
2. Computation is local, but problems have global constraints – how AI agents can address global problems?
3. Computation logic is deductive, but problems are not – many AI problems are abductive or inductive problems – how AI agents can address abductive or inductive problems?
4. The world is dynamic, but knowledge is limited – how AI agents can address a new problem?
5. Problem solving, reasoning, and learning are complex, but explanation and justification are even more complex. – how AI agent can explain or justify its decision?

**Characteristics of AI problems**

1. Data arrives incrementally
2. Problems exhibit recurring patterns
3. Multiple levels of abstraction or granularity
4. Computationally intractable
5. Knowledge of the world is static, world is dynamic
6. Knowledge is limited, world is open-ended

**Characteristics of AI agents**

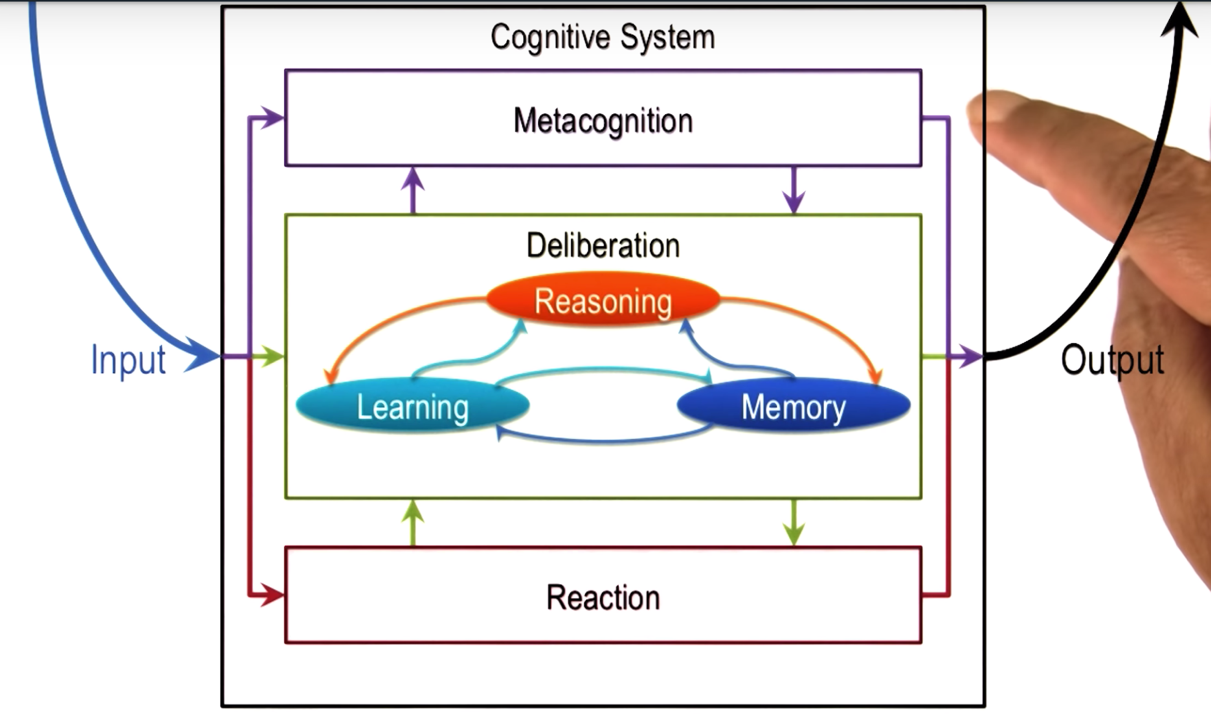
1. Limited computing power
2. Limited sensors
3. Limited attention
4. Computational logic is fundamentally deductive
5. Knowledge is incomplete relative to the world, world is large

**Watson – AI agent’s task**

* Identify the clue
* Search its knowledge database
* Decide on its answer
* Phrase the answer in the form of question

**Fundamental process of KBAI**

* Reasoning
* Learning
* Memory

They are intimately connected or closely related. These 3 are together called as **deliberation**.

**4 schools of AI**

1. Thinking
2. Acting
3. Optimal
4. Like humans

**4 quadrants**

* Thinking and optimal – machine learning
* Optimal and acting – airplane autopilot
* Acting and like humans – improvisational robots
* Like humans and thinking – semantic web

**Cognitive systems**

Cognitive – dealing with human like intelligence

Systems – multiple interactive components such as learning, reasoning and memory

World has some perceptions and cognitive systems is using sensors to perceive this percept.

Input is percept – take perceptions about the world

Output is actuators – carry out actions on the world

Single or multiple.

**Cognitive System Architecture** (figure given above)

**Reaction** – direct mapping of percepts into actions. E.g. Applying brakes in car when signal turns bright red.

**Deliberation** – Reasoning about the world around us. e.g. driving car and task in to change lanes, so you actually deliberate and look at goal and percepts, come up with plan representing what actions to take.

**3 components of deliberation:** learning, reasoning and memory

**Metacognition** - Reasoning about the internal mental world. It can reason about reaction or deliberation. E.g. change lanes and other cars honk as there was no space for them. will now think about deliberation that led to those actions. Then decide to change or reconfigure or repair the deliberation that to led to the plan of changing lanes.

**Topics in KBAI:**

**Fundamentals:**

Semantic networks, Production systems

Generate and test, means-end analysis, problem reduction

**Planning:**

Logic, Planning

**Common Sense Reasoning:**

Frames, Understanding, Common Sense Reasoning, Scripts

**Learning:**

Learning by recording cases, incremental concepts learning

Classification, Version spaces.

**Analogical Reasoning:**

Learning by recording cases, Case-based reasoning, Explanation-based learning, Analogical reasoning.

**Visuospatial Reasoning:**

Constraint Propagation, Visuospatial Reasoning

**Design and Creativity:**

Configuration, Diagnosis, Design, Creativity

**Metacognition:**

Learning by correcting mistakes, meta-reasoning, ethics in AI

**Lesson 2: Introduction to CS7637**

**Introduction to Computational Psychometrics:**

Study of intelligence, aptitude, knowledge of an AI agent as compared to human

**Raven’s Progressive Matrices**

Matrix problems: 2x2, 3x3, 2x1

**7 principles of this course:**

1. KBAI agents organize knowledge to guide and support reasoning.
2. Learning is incremental
3. Reasoning is top down and bottom up
4. Agents match methods to tasks.
5. Agents use heuristics to find solutions that are good enough, though not necessarily optimal.
6. Agents use recurring patterns in problems
7. The architecture of KBAI agents enables reasoning, learning and memory to supports and constrain each other.

**Lesson 3: Semantic Networks**

**Representations:**

There are several knowledge representations that AI has developed. In each knowledge representation, there is a language which has a **vocabulary** and **content** of some knowledge. E.g. f = ma. Vocabulary is the sign of equality or the algebraic equation, and the content is our knowledge of newtons second law of motion that is force equals mass times acceleration.

**Semantic Networks:**

In order to build a semantic network, first we represent the **objects** as x, y, z. Next represent the relationships using **links** between the nodes representing objects. Then, label the links as **above, inside**.

In order to capture transformation from A to B, we build links between objects in A and objects in B. These links are labeled as **unchanged, expanded, shrunk, deleted, filled and cleared**. If an object appears in A but not in B, then show a dummy node of that object with relationship as deleted. The dummy node is shown with dashed circle node.

**Structure of Semantic Networks:**

* Lexically: nodes (vocabulary)
* Structurally: directional links (complex representations)
* Semantically: application-specific labels (reason the representations)

**Characteristics of Good representations:**

* Make relationships explicit
* Expose natural constraints
* Brings objects and relations together
* Exclude extraneous details
* Transparent, concise, complete, fast, computable.

**Guards and Prisoners:** Quiz and examples given

**Choosing Matches by weights:** Similarity weights

Unchanged – 5pts

Reflected – 4 pts

Rotated – 3 pts

Scaled – 2 pts

Delete – 1 pt.

Shape changed – 0 pt.

**Connections:**

1. Memory – similarity metric – case-based reasoning
2. Analogical reasoning – correspondence problem
3. Cognition – focus on relationship

**Cognitive Connection:**

Semantic network is connected with human cognition.

1. It is a knowledge representation which is used solve a problem.
2. It is related to spreading activation networks. Example: A story with 2 sentences: John wanted to become rich. He got a gun. These nodes get activated and then nearby nodes also get activated to complete the story.

**Lesson 4: Generate & Test**

This is a **problem-solving method**. It is part of fundamental part. **Generator** generates all the possible states and **tester** will remove the states that are not correct.

Dumb generator and dumb tester cannot identify states that are similar to the initial states. This results in problem called **combinatorial explosion**, as small number of states produces large number of successor states very rapidly.

**Smart Testers**: It will dismiss all illegal states, it will also dismiss the states that are similar to the initial state, the states whose successor states had been ruled out or dismissed, and it will merge the states that are identical. This will result in a configuration of states that are similar to **semantic networks**.

**Smart Generators**: It will not generate the states that are non-productive for example the states that are similar to the initial state to begin with.

Thus, we can shift the balance between generator and tester depending on where we try to put the knowledge. For example, Genetic Algorithms using rules of recombination to find all possible potential successive states and then the fitness function acts as a tester. It is an effective method for a very large number of problems. It is also ineffective as neither generator nor tester is smart.

**Semantic Networks for Generate and Test**

It provides a level of abstraction in representing the world as it ignores low level of details. Thus, the **knowledge representation and problem-solving method** coupled together can now solve the provide that provide the reasoning.

**The Cognitive Connection**

Humans use generate and test as problem solving method all the time. This is because they lack complete or correct knowledge of the world or due to lack of computational resources or due to lack of correct reasoning. For example, Genetic algorithms is connected with human cognition and also with biological evolution such as crossover and mutation.

**Lesson 5: Means-Ends Analysis**

Fundamental part. Two other problem-solving methods are means-ends analysis and problem reduction. These methods as well as generate and test method are very useful for well-formed problems.

**Block world problem**

**State Spaces:**

In the state space, there is an initial state and a goal state, and a number of states that could be produces from initial state. There is a **path** from initial to goal state through some number of states. How will AI agent find the right path?

* Apply the operator to current state
* Calculate the difference between the current state and the goal state

**Differences in State Spaces:**

Reduce the difference (notion of distance) between the current state and the goal state to find the right path. This is **means-ends analysis**. This by itself cannot solve the problem because it can give 2 or more possible choices. For example. Difference, delta = 2 for two of the generated states.

Difference is 3 if there are 3 differences between current and goal state.

**Means-Ends Analysis** does not always accomplish the goal. It may sometimes take us away from the goal or it can get caught in loops. This method as well as generate and test method are examples of universal error methods. They are costly in terms of computational efficiency.

**Problem Reduction:**

Decompose a complex problem into multiple small problems. The sub-solutions are then composed to a single solution for the problem as a whole. Problem reduction is also a universal error method.

**Means-Ends Analysis for Raven’s**

Knowledge representation answers many questions in solving the problem about similarity, about correspondence. The **means-ends analysis** helps in doing a systematic analysis of the transformations. This can also be related to **problem reduction method** with 3 sub-goals – A to B transformation, use this to transform C to D candidates, compare D with given answers. It can relate to **generate and test method** too. These are weak methods, they do not require lot of knowledge in solving the problem.

**Cognitive Connection:**

Humans working in an unfamiliar domain do not have much knowledge about that domain, they are not experts. This is related to weak methods.

**Lesson 6: Production Systems**

Last topic under fundamental part. Part of it is about learning

* Cognitive Architectures
* Production systems
* Chunking (a mechanism of learning)

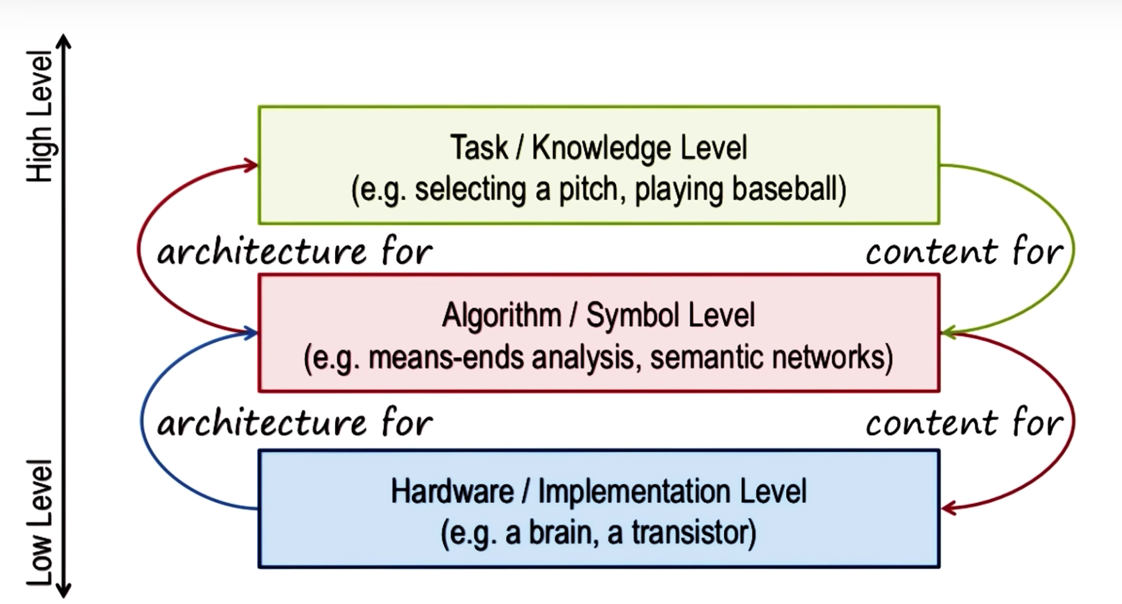
**Function – Cognitive architectures:**

A cognitive agent is a function that maps a perpetual history into an action.

f: P\* -> A

Percepts -> Action

Function of agent is to select an action.

**Levels – Cognitive architectures:**

**Assumptions of a Cognitive architecture:**

* Goal oriented
* Rich, complex environment
* Significant knowledge
* Symbols and abstraction
* Flexible and function of environment (dependent on environment)
* Learning

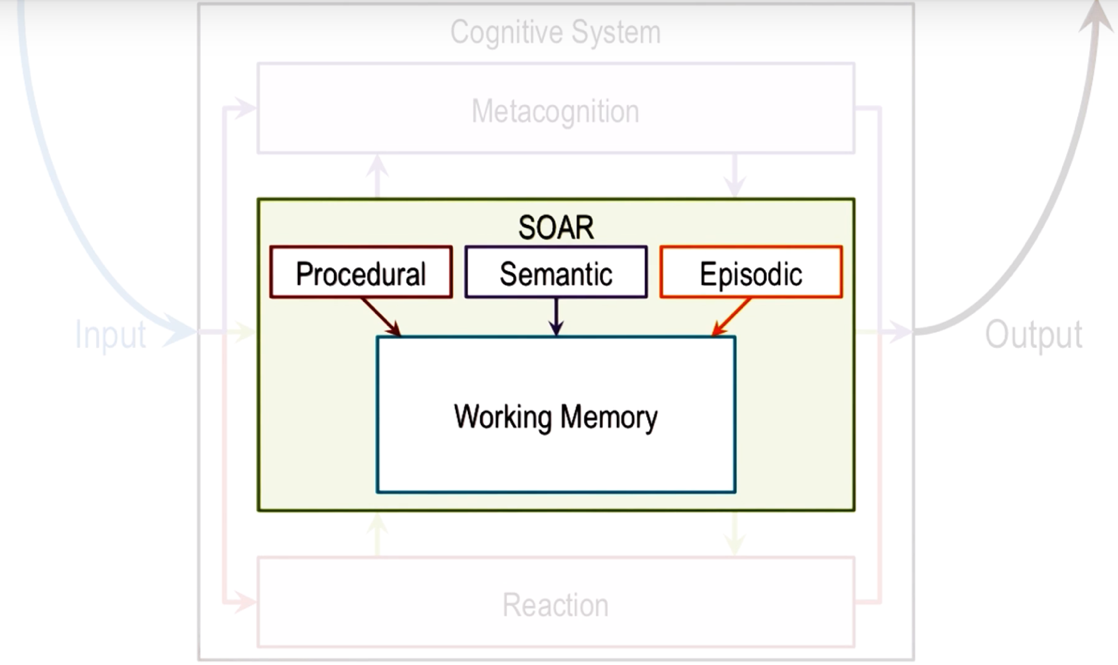
**Architecture + Content = Behavior**

If architecture is fixed, we can simply change the knowledge content to get different behaviors. Similarly, when trying to understand human behavior, we can map behavior to content if architecture is fixed.

**A cognitive architecture for production systems:**

SOAR – specific cognitive architecture for deliberation

* Long term memory - 3 kinds of knowledge -> procedural, semantic and episodic
* Working memory



**Bringing in Memory:**

SOAR’s long-term memory consists of various kinds of knowledge. In baseball example, we are talking about procedural knowledge (how to do something). Procedural knowledge is represented in the form of production rules. r1, r2, .. Given the contents of working memory (percepts), each production rule is of the form ‘if something then something’ i.e. antecedents and consequences (actions). Antecedents are connected through relationships such as and, and or. Similarly, consequences too are connected through and, and or.

As contents of working memory change rapidly, new production rules get activated in the long-term memory very slowly. There is constant interaction between working memory and long-term memory.

**Chunking:**

When SOAR’s architecture selects 2 rules, this is called as impasse. This occurs when the decision maker cannot decide. At this time, SOAR will attempt to **learn a rule** that might break the impasse by invoking **episodic** knowledge. This is called **Chunking**. SOAR searches in episodic memory for a previous event for knowledge to break the impasse. It then encapsulates the result of the previous event in the form of a rule.

* Decision making -> reasoning
* Learning -> chunking

**The cognitive connection:**

The working memory in production system is the short-term memory in human cognition. When given arithmetic problems, there are strong similarities between the behavior of programs like SOAR and behavior of humans.

**Lesson 7: Frames**

Example: Ashok ate a frog

Frame is Action of eating, slots and fillers, e.g. subject is slot and Ashok is a filler

Unit: Atom is production rule

Frames are molecules, they have a structure. Frame representation: **Ate**

* Subject: Ashok
* Object: Frog
* Location:
* Time:
* Utensils:
* Object-alive: False
* Object-is: in-subject
* Subject-mood: happy

**Complex Frame Systems:**

More Frames:

* **Person** name:Angela surname:Smith (corresponds to subject in Ate frame)
* **Restaurant** name: Olive Garden location:Atlanta price-range:$$ (corresponds to location in Ate frame)

**Properties of Frame**

* Represent stereotypes
* Provide default values
* Exhibit inheritance (Class may have instances which are inheriting properties from class)

**Frames and Semantic Nets**

* shape: triangle
* size: large
* fill: false
* inside: y
* above: z

**Frames and Production Systems:**

Working memory has frame with slots and fillers, which capture conceptual knowledge that is stored in the Semantic memory.

**The Cognitive Connection:**

1. Frames are atoms of knowledge representations and production systems are its molecules. Production rule can only capture small amount of information, whereas a frame can capture a large amount of information in organized manner as a packet.
2. Frames – construct a theory of cognitive processing not entirely bottom up but partially top down. It can generate expectations of the world which makes it top down.
3. Frames- capture stereotypes - cognitively efficient.

**Lesson 8: Learning by Recording Cases.**

**Preview:**

* Learning by recording cases
* Nearest neighbor method
* k-nearest neighbor
* cases in the real world

When a new problem arises, agent gives answer based on the cases already recorded in its memory. It simply sees which case most closely resembles the new situation.

**Learning by recording cases**

Given new problem a, Retrieve most similar prior problem, b, from memory. Apply b’s solution to problem a. Case is encapsulation of past experiences.

**Case retrieval by nearest neighbor**

Given existing case Xc, Yc and new problem Xn, Yn, Euclidean distance,

D = sqrt( square(Yc-Yn) + square(Xc-Xn) )

**Nearest Neighbor in k-dim space**

Given existing case at (C1, C2, ..Ck) and new problem at (P1, P2, …Pk), Euclidean distance,

D = Sqrt( (Summation of i=1tok) Square(Ci-Pi) )

**Limitations**: high dimensional space, store based on qualitative labels not numeric labels

**The cognitive connection**

Human memory deals with daily activities, same problems again and again. Memory supplied with answer. Reasoning, memory and learning.

**Lesson 9: Case-Based Reasoning**

Case based = extract something from memory and reuse it.

Process of case-based reasoning – 4 steps

**Retrieval** – retrieving a case from memory similar to the current problem (example: k-nearest neighbor)

**Adaptation** – adapting the solution to that case to fit the current problem

**Evaluation** – evaluating how well the adapted solution addresses the current problem

**Storage** – storing the new problem and solution as a case

This process unifies memory, reasoning and learning.

**Assumptions of case-based reasoning:**

Patterns exist in the world.

Similar problems have similar solutions.

**Case Adaptation by Model of the World**

**Case Adaptation by Recursive Reasoning**

New case is composed with the previous case to get a full solution. This is called as Compound analogy

**Case adaptation by rules**

Heuristic method. For example, to go back the same route, heuristic suggests to just flip all the terms.

**Case Evaluation**

**Case Storage**

Case choice is an important way of learning. We are constantly assimilating and accumulating new cases.

**Case Storage by Index**

Indexical structure – it allows for effective and efficient retrieval. Tabular form.

As number of dimensions increases, this method becomes less efficient. So, we have another method called discrimination trees.

**Discrimination Trees**: Is the origin North of 5N? Yes or No?

Search process is made more efficient by asking right questions to the right nodes. Big O notation, here its logarithmic not linear.

**Incremental Learning**: with each case, some new knowledge structure is learned.

**Advances Case-based Reasoning**

The process is not always linear. If in the Evaluation phase, the solution failed, then try adapting again. Alternatively, abandon case and retrieve a different case from memory. If retrieved case exactly matches current problem, then no adaptation needed, it directly goes from retrieval to evaluation phase. For e.g. kNN method. Storing failed cases helps us anticipate failures that might occur with new problems. Store only very successful cases that can help us cover larger problems to avoid utility problems.