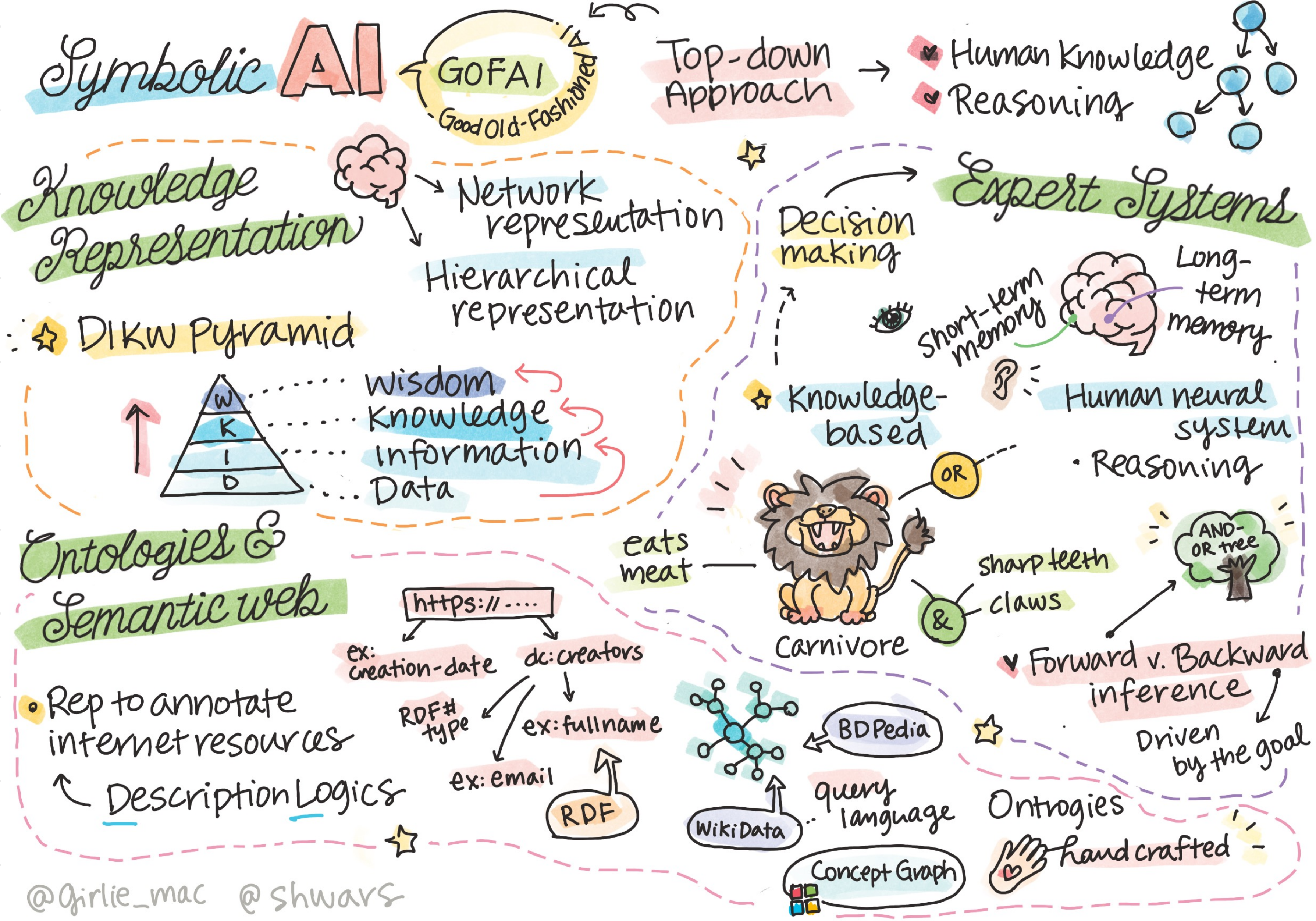
Artificial Intelligence (AI)

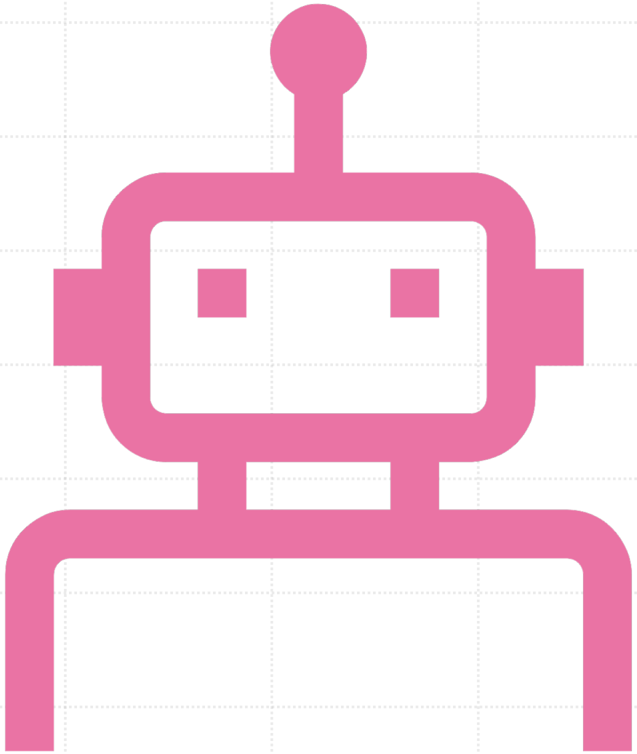
Knowledge Representation and
Expert Systems



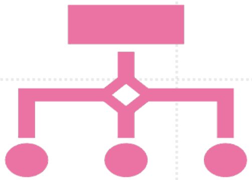
https://github.com/microsoft/AI-For-Beginners/blob/main/lessons/sketchnotes/ai-symbolic.png

AI's Quest for Knowledge

- Aim: Replicate human-like understanding and interpretation of the world.
- Early AI Strategies
  - Top-Down Approach: Dominant in AI's initial phases.
  - Principle: Transfer human knowledge into machine-readable formats for problem-solving.



Two Pillars of the Top-Down Approach



Knowledge Representation

Capturing real-world information and human understanding in a format that machines can process.

Involves structuring data, facts, and rules about the world in a logical and organized manner.

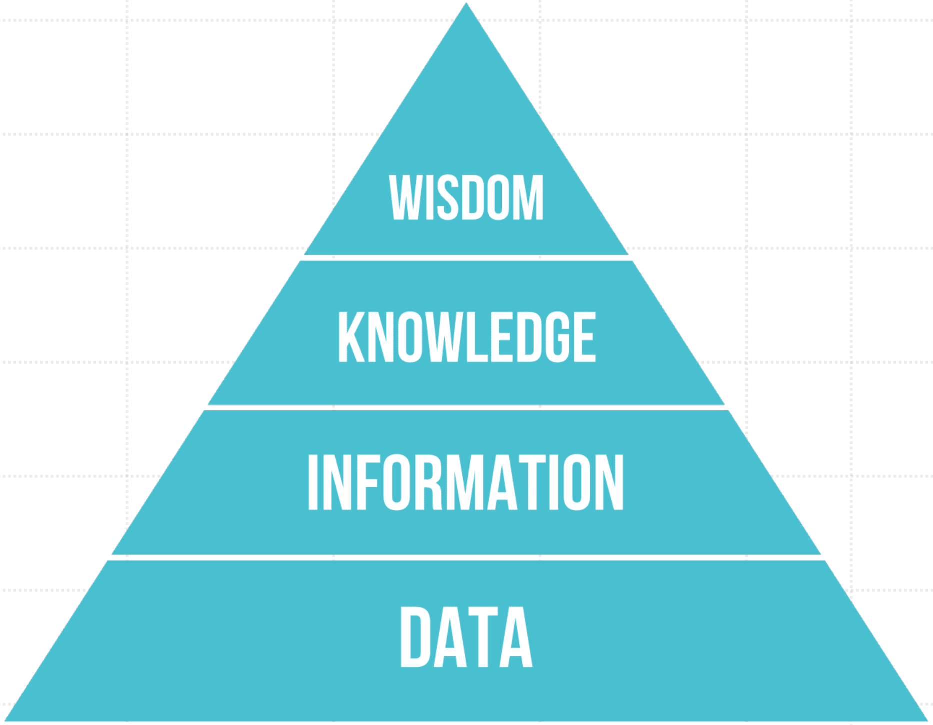
Reasoning

Enabling machines to use the represented knowledge to draw conclusions, make decisions, or solve problems.

Involves logical processes mimicking human thought patterns to reach conclusions or make predictions.

Understanding Knowledge in Symbolic AI

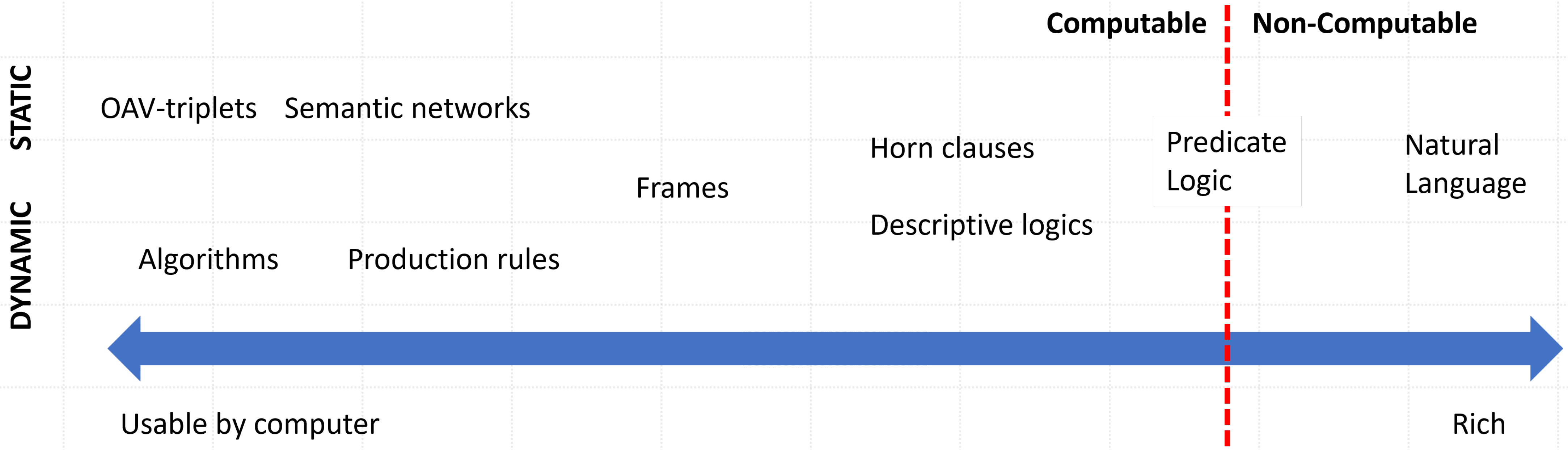
- Differentiation of knowledge from information and data.
- Data, Information, Knowledge, and Wisdom (DIKW Pyramid)
  - Data
    - Represented in physical forms like text or speech.
    - Independent of human perception; transferable between people.
  - Information
    - Personal interpretation of data.
    - Example: Understanding the concept of a "computer" from hearing or reading the word.
  - Knowledge
    - Information integrated into our world model.
    - Involves interconnected ideas about objects or concepts (e.g., understanding what computers are, their functionality, cost, applications).
  - Wisdom
    - Represents a higher level of understanding: meta-knowledge.
    - Concerns the application and contextual use of knowledge.



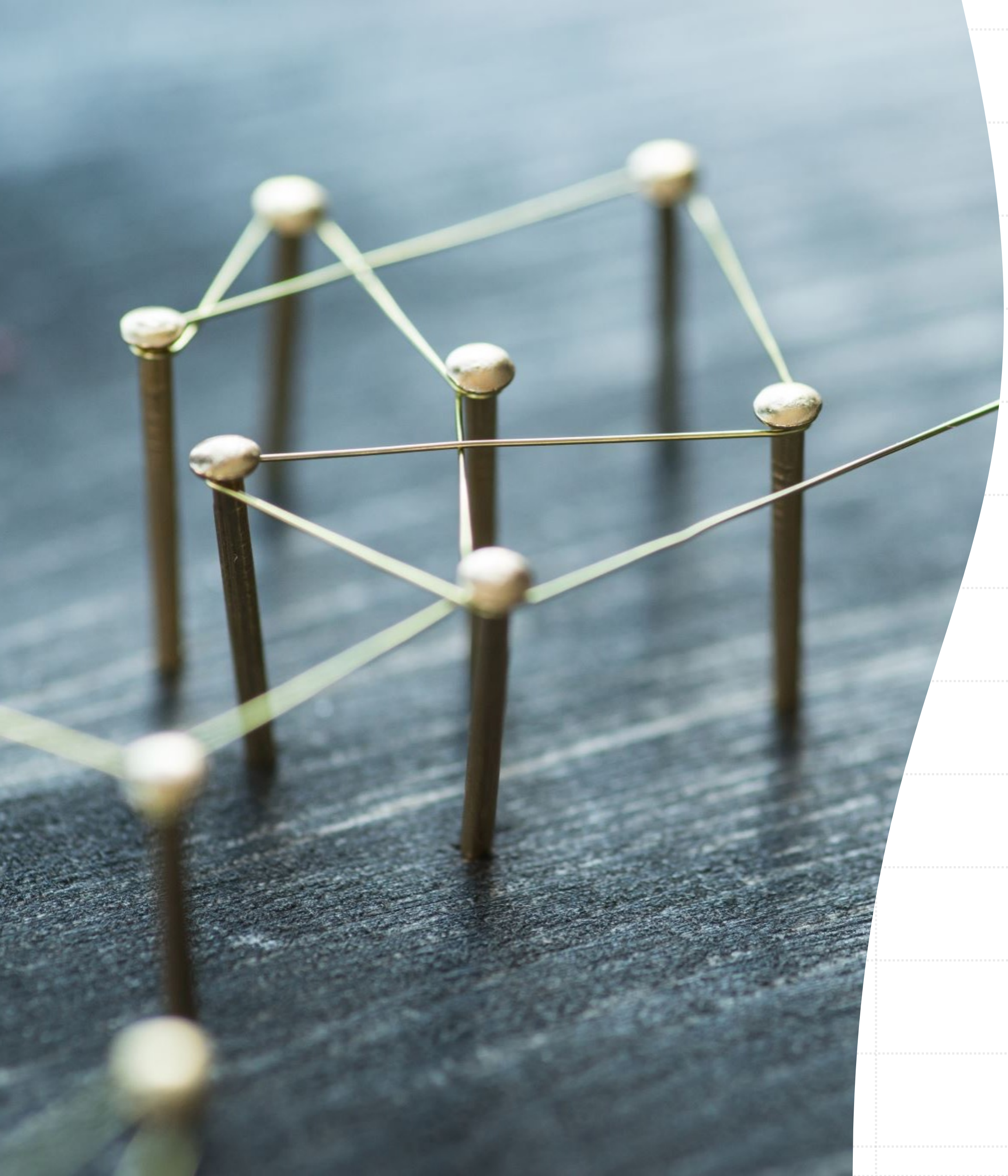
https://commons.wikimedia.org/w/index.php?curid=37705247

Spectrum of Knowledge Representation

- Simple Representations (Left Side)
  - Example: Algorithmic representation (knowledge as a computer program).
  - Pros: Easily utilized by computers.
  - Cons: Lacks flexibility; human knowledge is often non-algorithmic.
- Complex Representations (Right Side)
  - Example: Natural text.
  - Pros: Rich and powerful in content.
  - Cons: Difficult for automatic reasoning and computational processing.



https://github.com/microsoft/AI-For-Beginners/raw/main/lessons/2-Symbolic/images/knowledge-spectrum.png



Classifying Computer Knowledge Representations

- Network Representations
  - Based on human cognition's network of interrelated concepts.
  - Replicated in computers as semantic networks or graphs.
  - Semantic networks: Nodes represent concepts; edges represent relationships.

Object-Attribute-Value Triplets (or Attribute-Value Pairs)

- A practical way to represent semantic networks in computers.
- Utilizes triplets or pairs to denote objects, their attributes, and corresponding values.
- Suitable for organizing and storing structured information.
- Example: Representing programming languages with specific triplets (e.g., [Language Name] - [Attribute] - [Value]).

Object	Attribute	Value
Python	is	Untyped-Language
Python	invented-by	Guido van Rossum
Python	block-syntax	indentation
Untyped-Language	doesn't have	type definitions

Hierarchical and Frame-Based Knowledge Representation

Hierarchical Representations

- Organizes concepts in a hierarchical structure, like a tree.
- Reflects the way humans categorize objects and their characteristics.
- Example: Understanding a canary as a type of bird, with attributes like color and flight speed common to birds.

Frame Representation

- Each object or class of objects represented by a "frame."
- Frames contain "slots" for various attributes and their values.
- Slots may have default values, value restrictions, or procedures for value determination.
- Frames are organized hierarchically, akin to object hierarchies in object-oriented programming.

Scenarios

- A specific type of frame representing dynamic, time-evolving situations.
- Useful for modeling complex sequences of events or interactions.

Procedural Representations

Knowledge represented through actions executed under specific conditions.

Useful for dynamic and conditional processes.

Production Rules

If-then statements for logical reasoning and conclusions.

Example: Medical diagnostics based on symptoms and test results (e.g., fever and C-reactive protein levels indicating inflammation).

Facilitates automated reasoning based on predefined conditions.

Algorithms

Sequential procedures for solving specific problems.

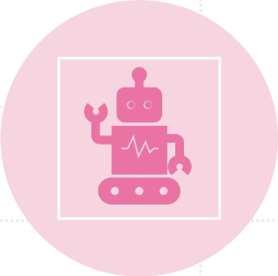
Less common in knowledge-based systems but important in procedural representation.

Logic Representation

- Originating from Aristotle's theories for universal human knowledge representation.
- Predicate Logic
  - Rich mathematical theory; computable subsets used in AI (e.g., Horn clauses in Prolog).
- Descriptive Logic
  - Logical systems for representing and reasoning about object hierarchies.
  - Applied in distributed knowledge representations like the semantic web.

Expert Systems in Symbolic AI

Early success stories in the development of AI.



Designed to simulate the decision-making ability of a human expert in a specific domain.

Components of Expert Systems

- Knowledge Base
  - Contains domain-specific knowledge extracted from one or more human experts.
  - Includes facts, heuristics, and rules pertinent to the domain.
- Inference Engine
  - Applies logical rules to the knowledge base to deduce new information or make decisions.
  - Performs reasoning processes to solve problems or provide recommendations.

Functionality and Application

Expert systems utilize the captured expertise to provide solutions or advice.

Applied in various fields like medical diagnosis, financial services, and engineering.

Anatomy of Expert Systems: Mimicking Human Reasoning

Expert systems designed to parallel human cognitive processes.

Include components analogous to short-term and long-term memory.

Components of Knowledge-Based Systems

Problem Memory

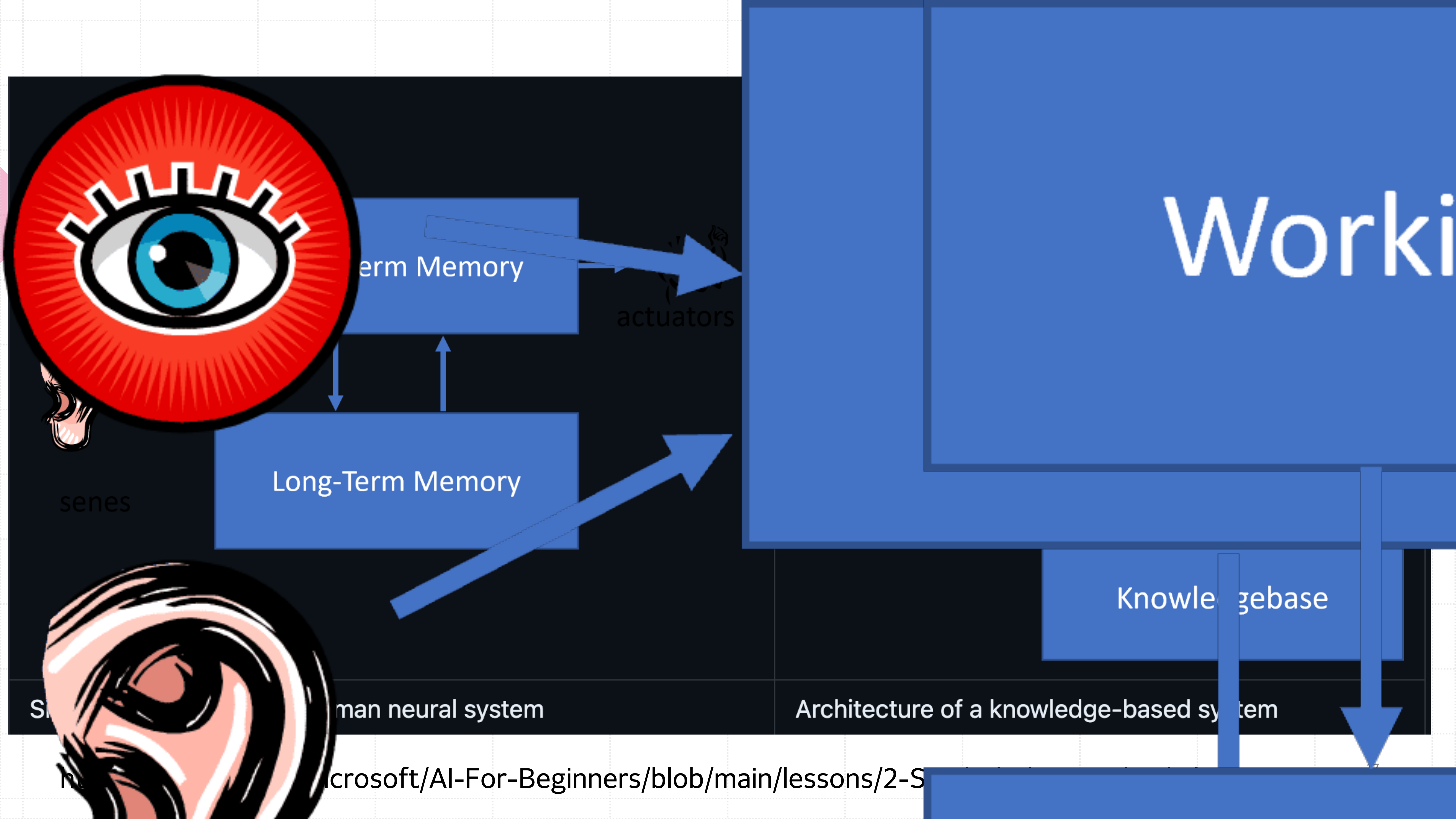
- Represents the current state of the problem being solved.
- Contains specific, situational data (e.g., a patient's temperature or blood pressure).
- Termed "static knowledge" as it captures a snapshot of the problem at a specific point.

Knowledge Base

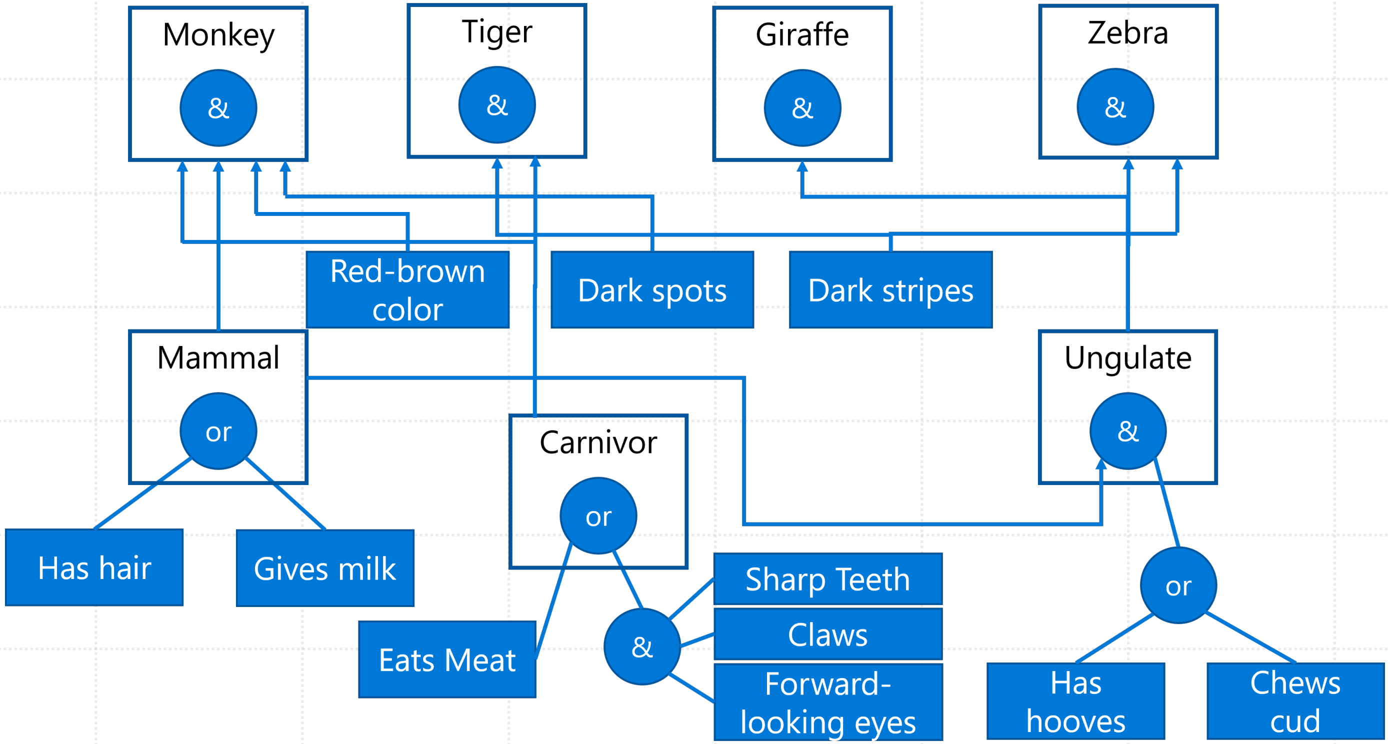
- Analogous to long-term memory in human reasoning.
- Houses long-term, domain-specific knowledge.
- Consists of rules and facts extracted from human experts.
- Remains consistent across different consultations; known as "dynamic knowledge" for its role in navigating problem states.

Inference Engine

- Central processing unit of the system.
- Manages navigation through the problem state space.
- Asks questions and selects appropriate rules for each problem state.
- Integrates problem memory with the knowledge base to deduce conclusions or solutions.



crosoft/AI-For-Beginners/blob/main/lessons/2-S



IF the animal eats meat
OR (animal has sharp teeth
AND animal has claws
AND animal has forward-looking eyes
)
THEN the animal is a carnivore

Expert system of determining an animal based on its physical characteristics
https://github.com/microsoft/AI-For-Beginners/raw/main/lessons/2-Symbolic/images/AND-OR-Tree.png

Understanding AND-OR Trees and Production Rules

AND-OR Tree

- Visual tool for mapping out production rules.
- Aids in initial stages of knowledge extraction from experts.

Transition to Rules for Computer Representation

- More practical to use rule-based representation for computational purposes.
- Each rule consists of conditions (left-hand side) and an action (right-hand side).

Role of Object-Attribute-Value (OAV) Triplets

Conditions and actions in rules often formatted as OAV triplets.

Example of an OAV triplet: [Object] - [Attribute] - [Value].

Working Memory and Rules Engine in Expert Systems

- **Working Memory**
  - Stores OAV triplets relevant to the current problem.
  - Represents the 'problem memory' or 'static knowledge'.
- **Rules Engine**
  - Searches for rules whose conditions match the current OAV triplets in working memory.
  - Executes applicable rules, generating new OAV triplets and evolving the problem state.

Forward vs. Backward Inference

- Starts with some initial data about the problem available in the working memory, and then executes the following reasoning loop:
  1. If the target attribute is present in the working memory - stop and give the result
  2. Look for all the rules whose condition is currently satisfied - obtain conflict set of rules.
  3. Perform conflict resolution - select one rule that will be executed on this step. There could be different conflict resolution strategies:
     ① Select the first applicable rule in the knowledge base
     ② Select a random rule
     ③ Select a more specific rule, i.e. the one meeting the most conditions in the "left-hand-side" (LHS)
  4. Apply selected rule and insert new piece of knowledge into the problem state
  5. Repeat from step 1.

Dynamic Knowledge Acquisition in Expert Systems

In certain scenarios, beginning with minimal pre-existing knowledge is more effective.

Approach involves actively gathering information as needed, rather than relying solely on a pre-filled knowledge base.

Example: Medical Diagnosis

Process often starts without comprehensive prior medical analysis.

Information is collected progressively based on the evolving needs of the diagnosis.

Medical tests and analyses are conducted in response to specific questions or symptoms, guiding the diagnostic process.

Interactive Questioning Strategy

System asks targeted questions to gather relevant information.

Each response informs subsequent questions, leading to a more precise understanding of the problem.

This approach mirrors how medical professionals often operate in real-world diagnostics.

Backward inference - driven by the goal - the attribute value that we are looking to find:

Select all rules that can give us the value of a goal (i.e. with the goal on the RHS ("right-hand-side")) - a conflict set

If there are no rules for this attribute, or there is a rule saying that we should ask the value from the user - ask for it, otherwise:

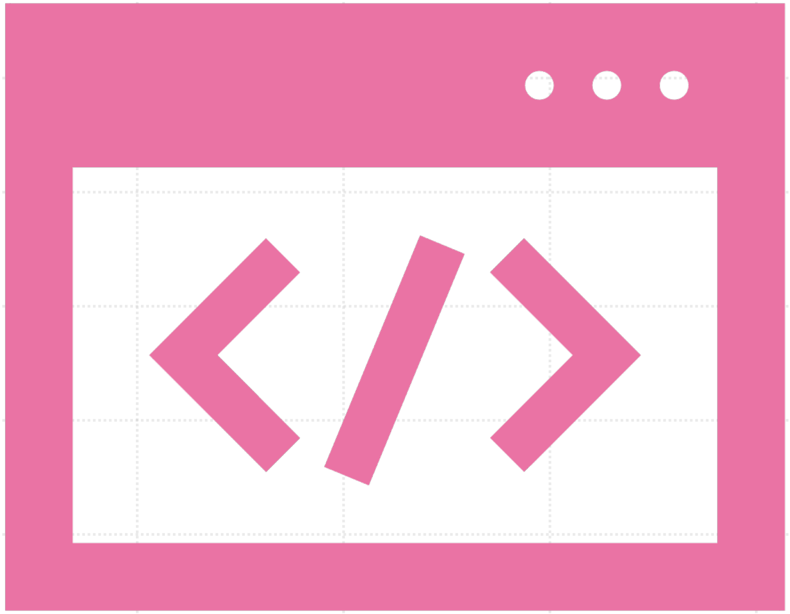
Use conflict resolution strategy to select one rule that we will use as hypothesis - we will try to prove it

Recurrently repeat the process for all attributes in the LHS of the rule, trying to prove them as goals

If at any point the process fails - use another rule at the third step.

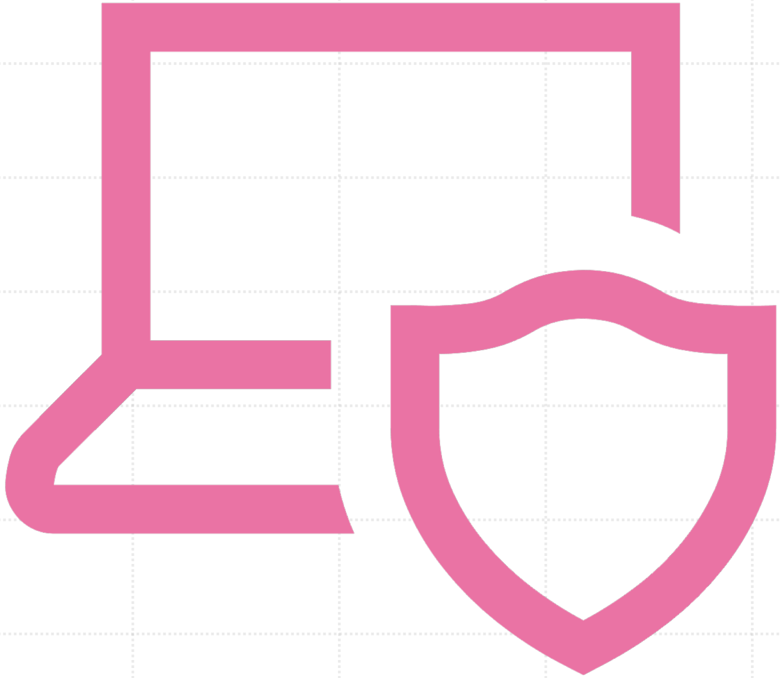
Implementation Approaches for Expert Systems

- Direct Programming in a High-Level Language
  - Involves coding the expert system in languages like Python, Java, etc.
  - Not ideal due to the conflation of knowledge and inference processes.
  - Requires understanding of both domain expertise and programming intricacies.



Implementation Approaches for Expert Systems (cont'd)

- Using Expert Systems Shells
  - Specialized software designed for developing expert systems.
  - Allows domain experts to input rules using a knowledge representation language.
  - Separates the knowledge base from the inference engine.
  - Enables experts to contribute without needing in-depth programming knowledge.



Advantages of Expert Systems Shells

Simplifies the process of building and updating the knowledge base.

Facilitates easier maintenance and modification of rules by domain experts.

More efficient and user-friendly for non-programmers involved in system development.

Exercise: Implementing an Animal Expert System

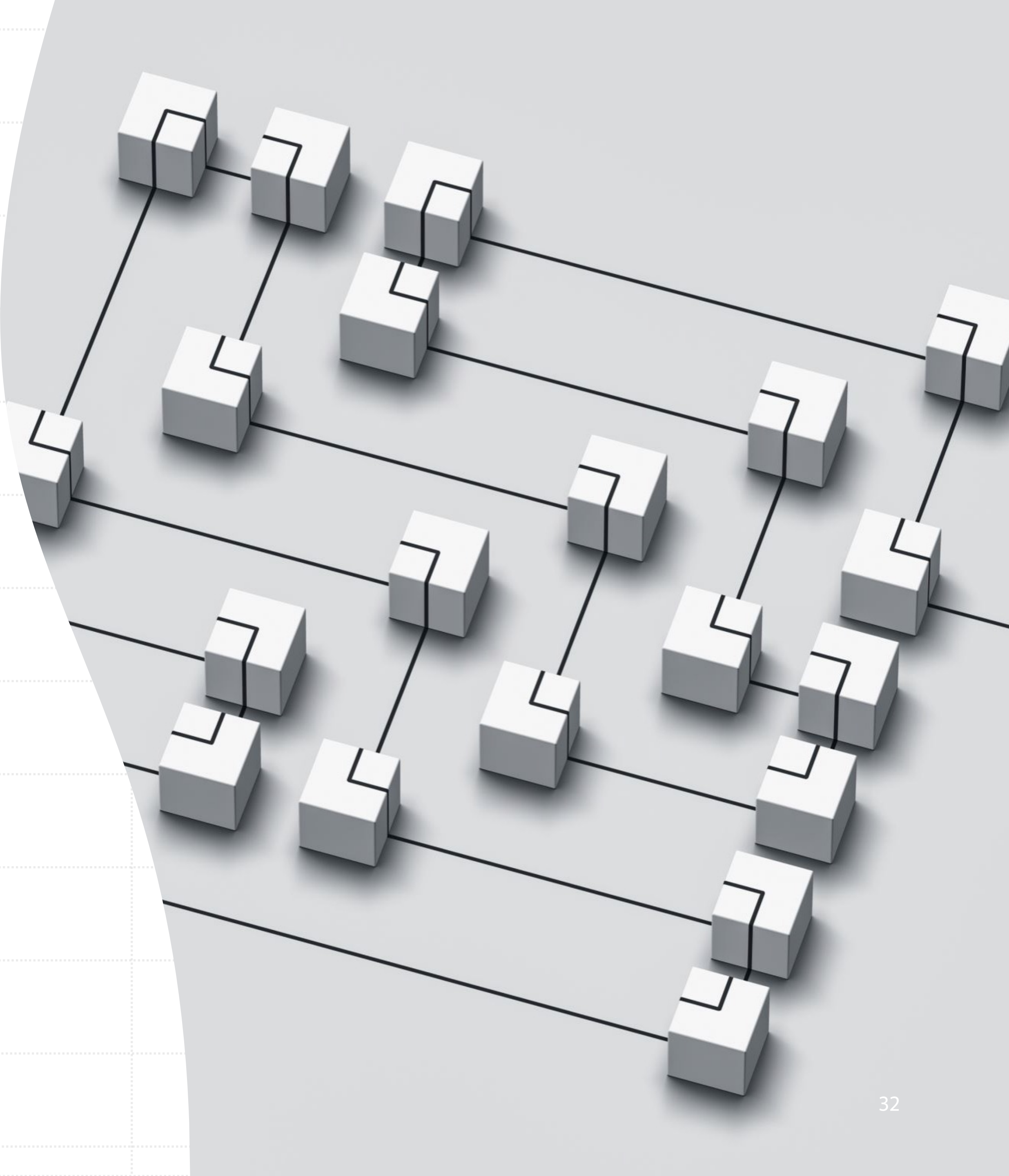
- Animals.ipynb
- - need Python 3.8.10

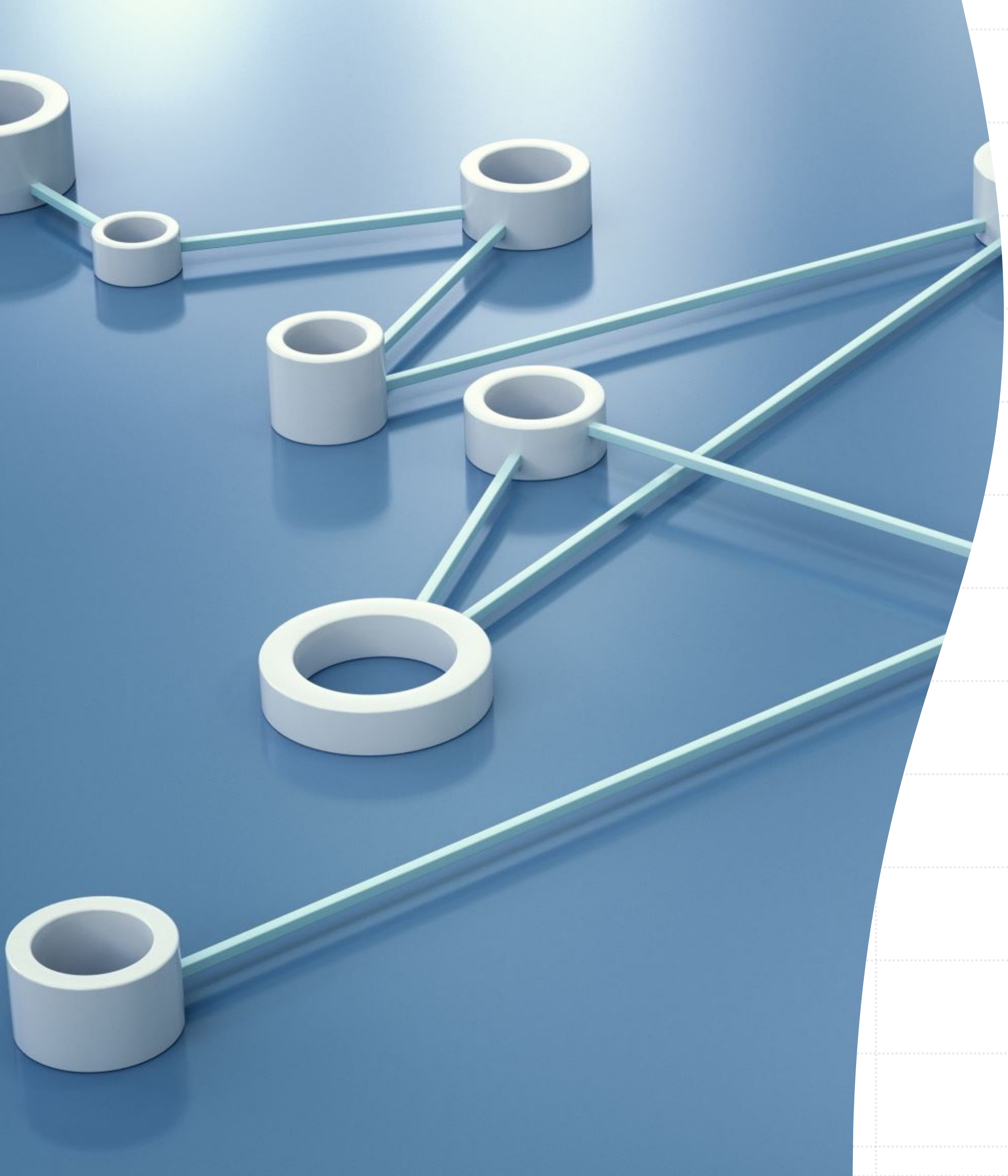
Ontologies and the Semantic Web

- Semantic Web Initiative
  - Launched at the end of the 20th century to enhance internet resource discovery.
  - Aim: Allow for more specific and accurate retrieval of web resources.

Key Concepts of the Semantic Web

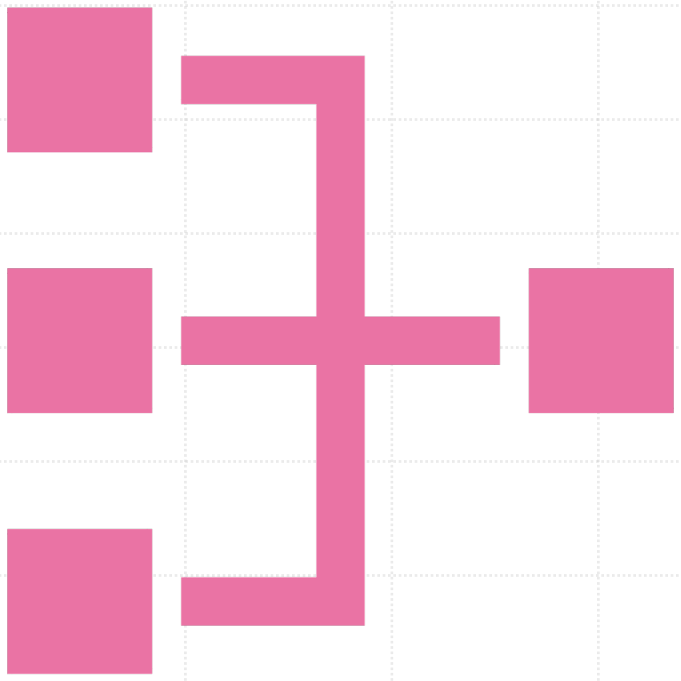
- Description Logics (DL)
  - Serves as the knowledge representation foundation.
  - Similar to frame representation with a formal logical structure and inference capabilities.
  - Balances between expressiveness and computational feasibility of inference.
  - Comprises a family of logics varying in complexity and expressiveness.





Key Concepts of the Semantic Web (cont'd)

- Distributed Knowledge Representation
  - Utilizes global URI identifiers for concepts.
  - Facilitates the creation of interconnected knowledge hierarchies across the internet.



Key Concepts of the Semantic Web (cont'd)

- XML-based Languages for Knowledge Description
  - RDF (Resource Description Framework): A standard model for data interchange.
  - RDFS (RDF Schema): Provides vocabulary for RDF data modeling.
  - OWL (Ontology Web Language): Extends RDF and RDFS for richer and more complex web ontologies.
  - All representations are based on triplets - each object and each relation are uniquely identified by the URI

Challenges and Progress in Semantic Web Development

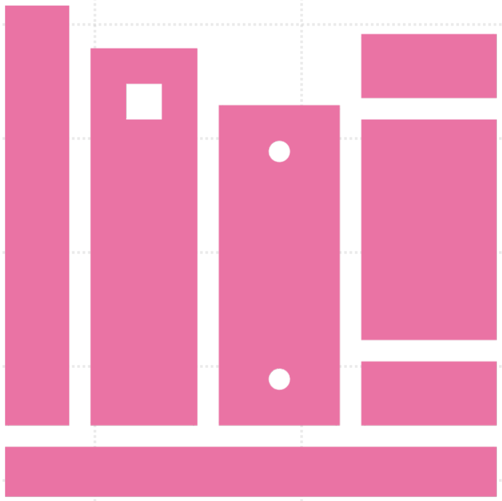
Initial momentum slowed by advancements in search engines and natural language processing (NLP).

NLP allows for efficient extraction of structured data from unstructured text.

Despite this, efforts in developing ontologies and knowledge bases continue in specific areas.

Notable Projects in Semantic Web and Ontologies

- 1. WikiData
  - A comprehensive machine-readable knowledge base linked to Wikipedia.
  - Primarily sourced from Wikipedia InfoBoxes, which offer structured information on various topics.
  - Allows querying using SPARQL, a dedicated query language for accessing Semantic Web data.
- 2. DBpedia is another effort similar to WikiData.

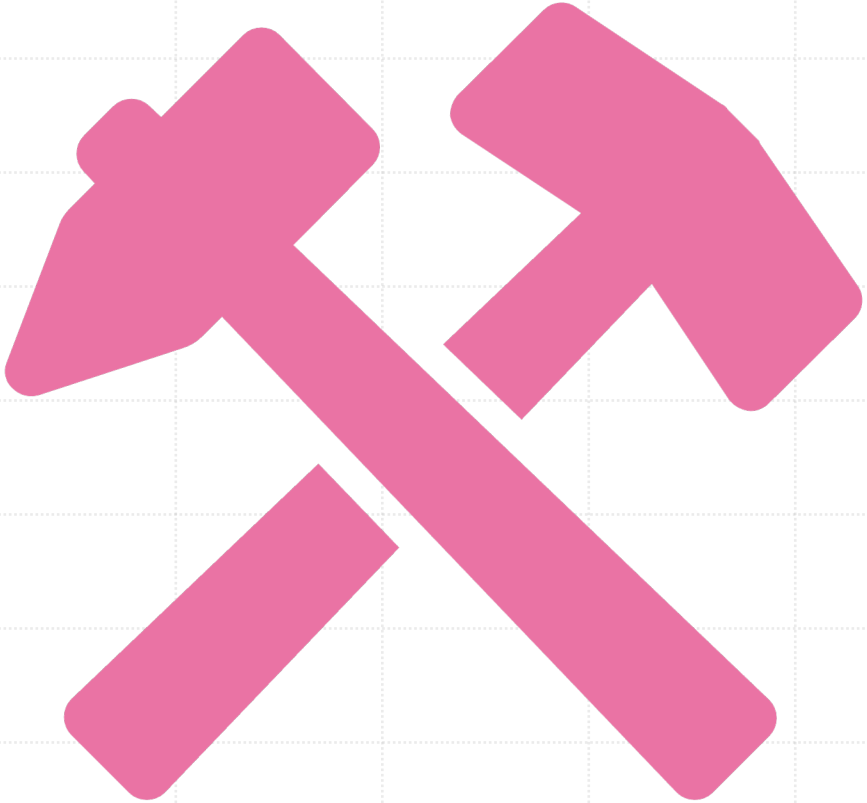


A sample query that displays most popular eye colors among humans

```
#defaultView:BubbleChart
SELECT ?eyeColorLabel (COUNT(?human) AS ?count)
WHERE
{
  ?human wdt:P31 wd:Q5.       # human instance-of homo sapiens
  ?human wdt:P1340 ?eyeColor. # human eye-color ?eyeColor
  SERVICE wikibase:label { bd:serviceParam wikibase:language "en". }
}
GROUP BY ?eyeColorLabel
```

Microsoft Concept Graph

- Ontology Creation and Mining
  - Traditionally, ontologies are meticulously crafted by hand.
  - An alternative approach involves mining ontologies from unstructured data, such as natural language texts.



Microsoft Concept Graph (cont'd)

A project by Microsoft Research exemplifying ontology mining.

Features:

- A vast collection of entities connected by "is-a" relationships.
- Organizes entities into groups based on common characteristics or classes.
- Capable of providing probabilistic classifications of entities (e.g., Microsoft as "a company" or "a brand" with respective probabilities).

Accessibility:

- Available through a REST API for integration and querying.
- Can also be downloaded as a comprehensive text file listing all entity relationships.

AI and Its Current Perception

Commonly equated with Machine Learning (ML) or Neural Networks (NNs) in contemporary discourse.

These areas, particularly NNs, have seen substantial advancements and attention.

The Role of Explicit Reasoning in AI

Human cognition involves explicit reasoning, a facet not fully replicated by NNs or ML models.

Explicit reasoning involves logical processing and deductive thinking, crucial for understanding and explaining decisions.

Application in Real-World Projects

Explicit reasoning remains vital for tasks requiring detailed explanations or controlled behavioral modifications.

Used in systems where understanding the 'why' behind decisions or actions is as important as the outcome itself.

Balancing Neural Networks and Explicit Reasoning

Many advanced AI systems integrate NNs for pattern recognition and predictive analysis with explicit reasoning for clarity and control.

This hybrid approach aims to leverage the strengths of both methodologies for more robust, understandable, and adaptable AI solutions.