# 3. Knowledge Representation & Reasoning

## Artificial Intelligence and Neural Network (AINN)

## Part I

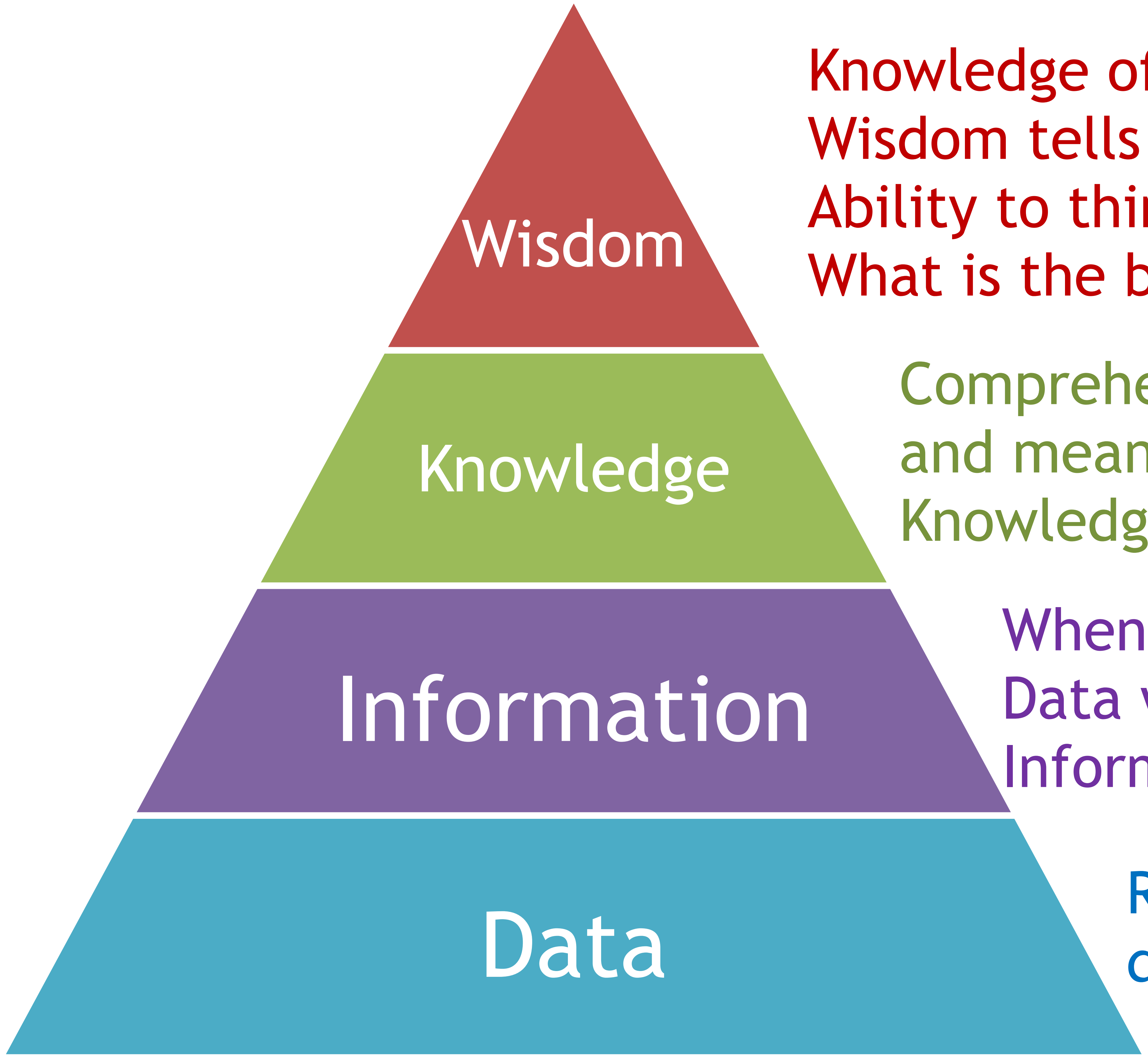
**Dr. Udaya Raj Dhungana**
Assist. Professor
Pokhara University, Nepal
Guest Faculty
Hochschule Darmstadt University of Applied Sciences, Germany
E-mail: udaya@pu.edu.np and udayas.epost@gmail.com

# Overview

- Knowledge hierarchy
- Knowledge Representation
- Requirements of Knowledge Representation
- Approaches to Knowledge representation
- Propositional logic

# Knowledge Hierarchy

- also called DIKW pyramid
- shows relations between **Data, Information, Knowledge** and **Wisdom**




Knowledge of what is true and right.
Wisdom tells us WHY.
Ability to think and act using knowledge.
What is the best?

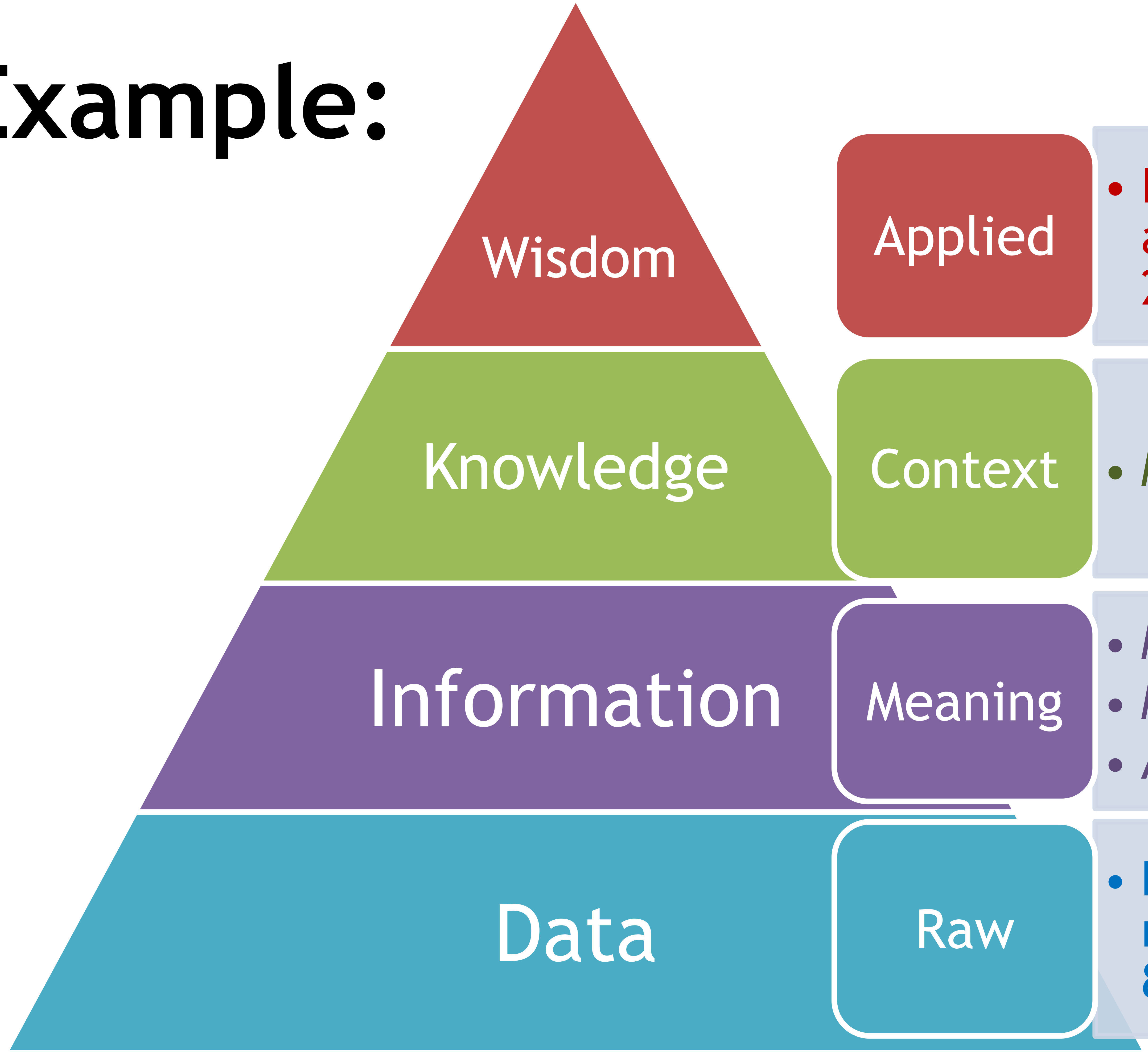
Comprehend the relationship of data to information (both their reason and meaning).
Knowledge tell us HOW.

When data is brought into context, it becomes information.
Data with analyzed relationships and connections.
Information tell us WHAT, WHO, HOW, WHERE

Raw facts, individual elements,
data by itself nothing useful.

# Knowledge Hierarchy

**Example:**




- I better to turn on the AC and set the temperature to 25°c.

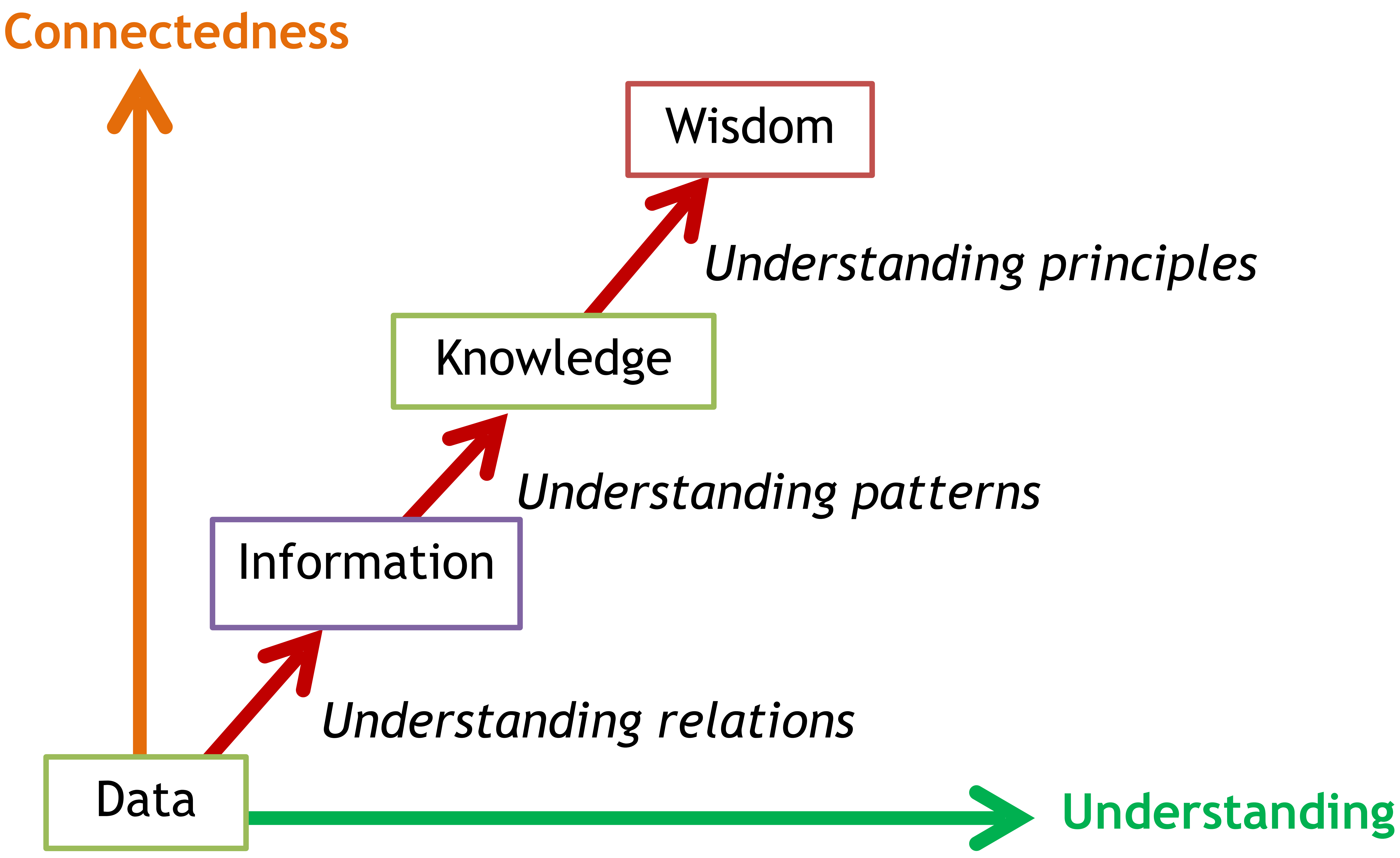
- My room temperature is low.

- Max temperature = 12°c
- Min temperature = 8°c
- Avg. Temperature = 10°c

- Every hour temperature reading of my room (in °c): 8, 12, 10

# Knowledge Hierarchy

The degree of *"connectedness" and "understanding"* increase, as we progress from *data to  information to knowledge to wisdom.*




**Knowledge Model (Bellinger 1980)**

# Knowledge types

- Tacit vs. Explicit Knowledge

| Tacit/*implicit* knowledge | Explicit knowledge |
|---|---|
| Exists within a human being;<br>it is embodied. | Exists outside a human being;<br>it is embedded. |
| Difficult to articulate formally. | Can be articulated formally. |
| Difficult to share/<br>communicate. | Can be shared, copied, processed and stored. |
| Hard to steal or copy. | Easy to steal or copy |
| *Drawn from experience, action,<br>subjective insight.* | *Drawn from artifact of some type as principle, procedure, process, concepts.* |

# Knowledge Representation

- **To solve the problems in a domain, we need**
  - **Knowledge** about the domain and
  - Some mechanisms for representing that knowledge (facts) in computer so that
    - The knowledge can be manipulated to create solutions to new problems

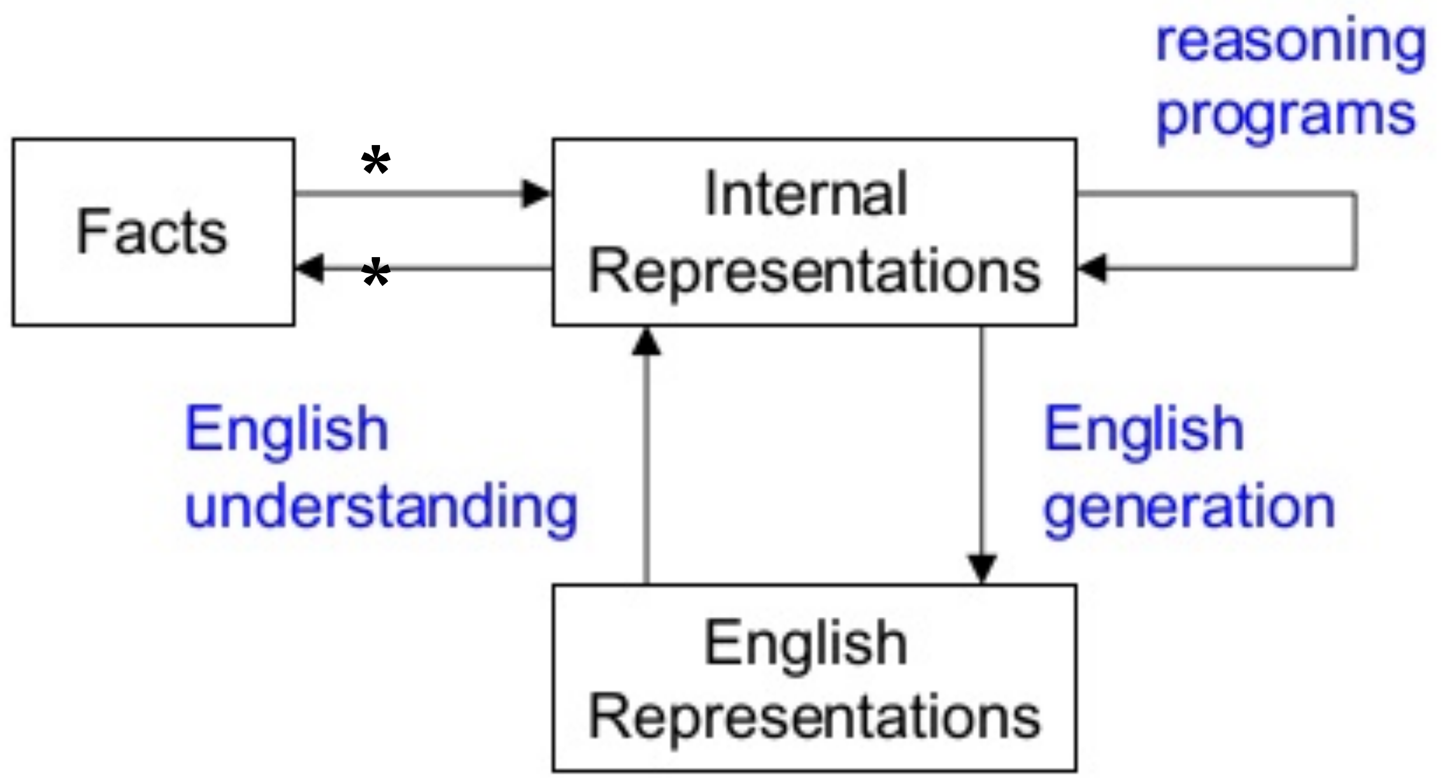
- A **knowledge base**
  - is the representation of knowledge that is stored by an agent.

# Knowledge Representation

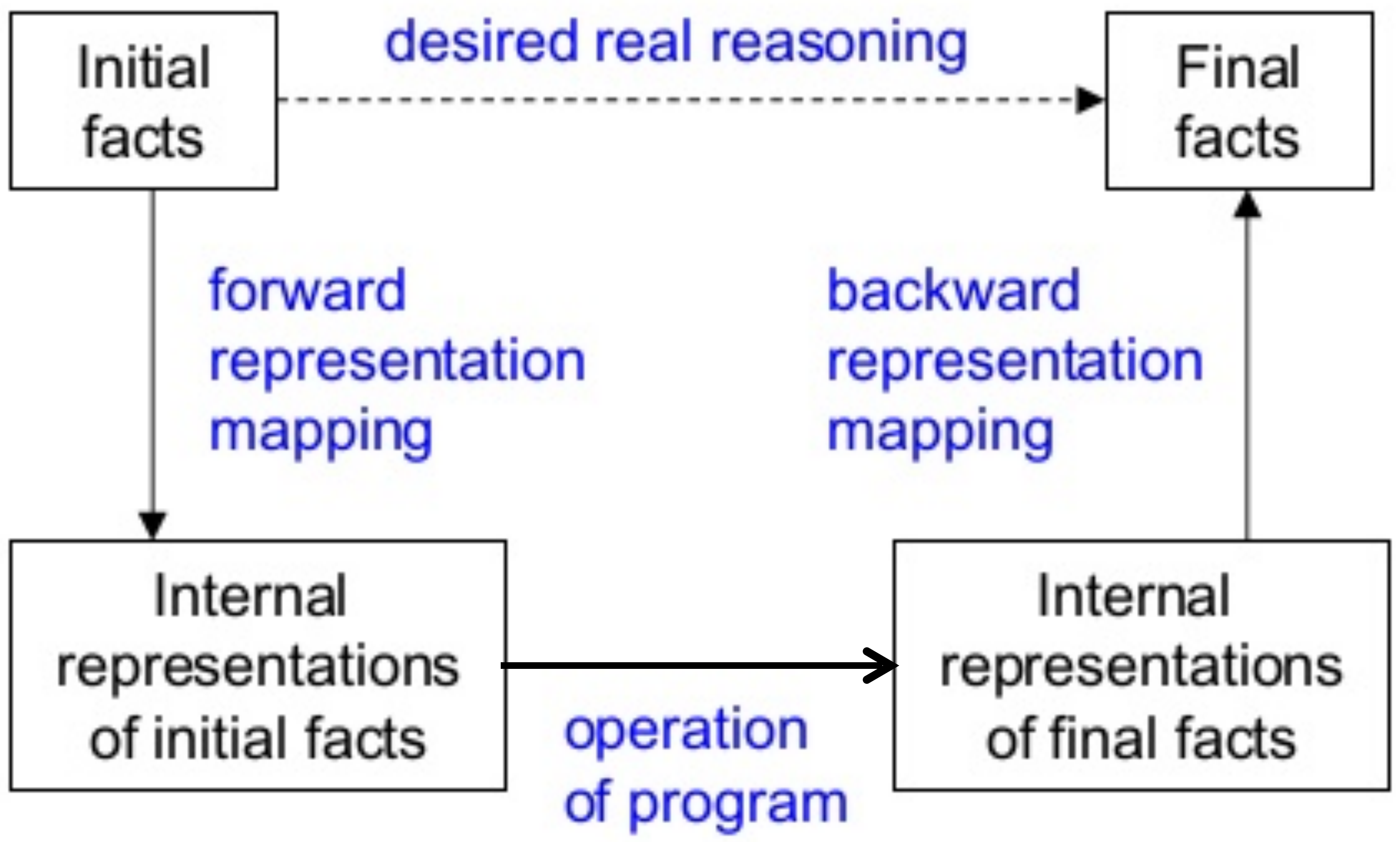
- Knowledge representation deals with
  - Facts (***knowledge level***)
    - are the truths in some relevant world
    - are need to be described first
      - these are the things we want to represent
  - Representation of facts (***symbol level***)
    - is some mechanism to encode the facts in terms of symbols
    - These are the things that can be manipulated by programs to reason and solve the problem

# Knowledge Representation

- Representation and mapping




a) Mapping between facts and representations




b) Representations of facts [stared portion of Fig a.]

# Knowledge Representation

- Representation and mapping

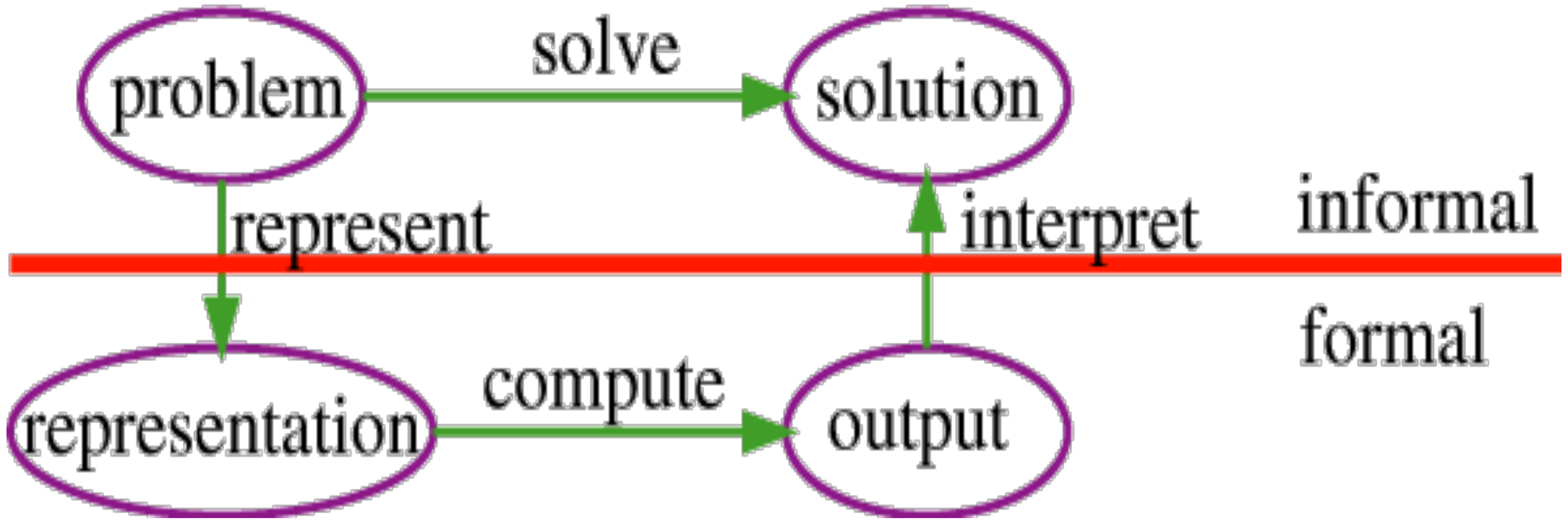
| Facts (in English) | Action applied | Representations (in Logic) |
| --- | --- | --- |
| Spot is a dog.<br>Every dog has a tail. | Forward Mapping *<br>Forward Mapping | dog(Spot)<br>$\forall x$: dog(X) → hastail(X) |
| | Deductive mechanism | hastail(Spot) |
| Spot has a tail. | Backward mapping** | hastail(Spot) |

New fact

*Forward representation mapping function to represent fact in logic
** Backward representation mapping function to generate English sentence of that fact

# Knowledge Representation

- Framework for solving problems (Pool, 2010)




  - flesh out the task and determine what constitutes a solution
  - represent the problem in a language with which a computer can reason;
  - use the computer to compute an output, which is an answer presented to a user or a sequence of actions to be carried out in the environment; and
  - interpret the output as a solution to the problem.

# Requirements of KR System

- A good knowledge representation system should have the following properties:
  - Representational Adequacy
    - The ability to represent all kinds of knowledge that are needed in that domain.
  - Inferential Adequacy
    - The ability to manipulate the representational structures to derive new structure corresponding to new knowledge inferred from old .

# Requirements of KR System

- A good knowledge representation system should have the following properties:
  - Inferential Efficiency
    - The ability to incorporate additional information into the knowledge structure that can be used to focus the attention of the inference mechanisms in the most promising direction.
  - Acquisitional Efficiency
    - The **ability to acquire new information/knowledge** easily.

# Requirements of KR System

- Unfortunately, no single system that optimizes all of the capabilities for all kind of knowledge has yet been found.
- As a result, multiple techniques for knowledge representation exist.

# Approaches to Knowledge Representation

- Simple relational knowledge
- Inheritable Knowledge
- Inferential Knowledge
- Procedural Knowledge

# Approaches to Knowledge Representation

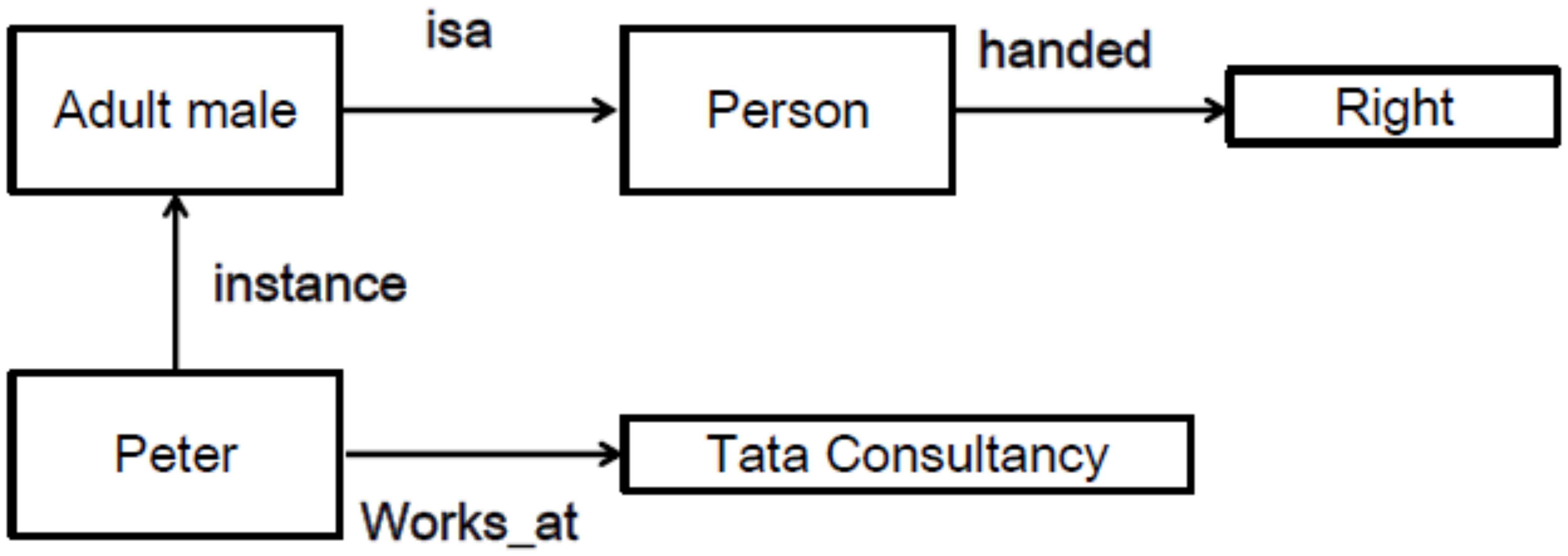
- Simple relational knowledge
  - Represents the declarative facts as a set of relations in database systems
  - Provides very weak inferential capabilities.
  - May serve as the input to powerful inference engines.

| Player | Height | Weight | handed |
|---|---|---|---|
| Peter | 6-0 | 180 | right |
| Ajay | 5-10 | 170 | left |
| John | 6-2 | 215 | left |
| Vickey | 6-3 | 205 | right |

  - Fail to answer “Who is the heaviest player?
    - But if a procedure is provided for finding heaviest player, these facts will enable the procedure to compute the answer.

# Approaches to Knowledge Representation

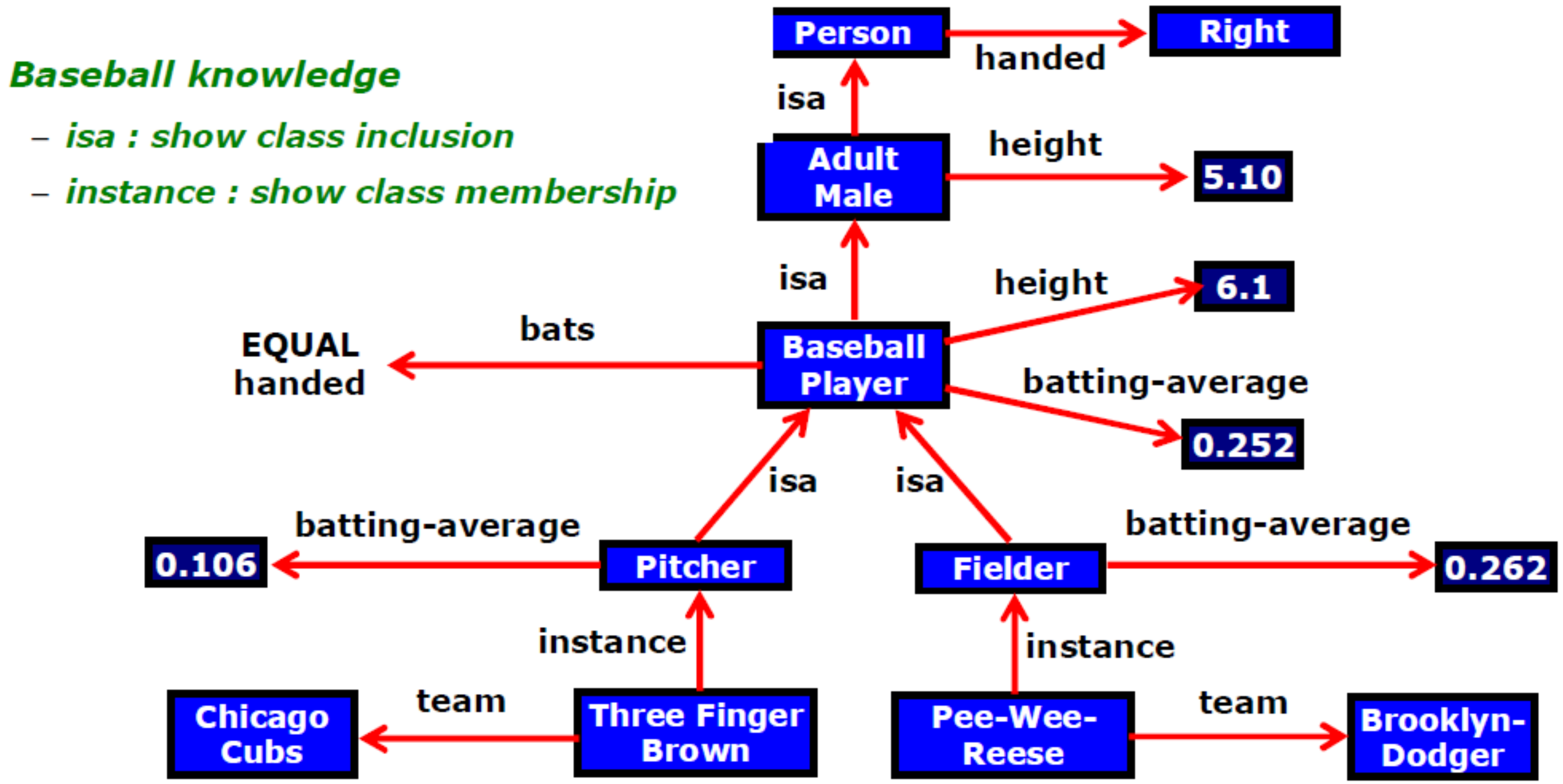
- Inheritable Knowledge
  - Objects are organized into classes and classes are organized in a generalization hierarchy.
  - Inheritance is a powerful form of inference, but not adequate.



# Approaches to Knowledge Representation

*(refer book -> AI by Elaine Rich , Kevin Knight and S B Nair, 3rd ed, page no 84)*

- Inheritable Knowledge




**Queries and ans:**

- team (Pee-Wee-Reese) = Brooklyn-Dodger
- batting-average(Three-Finger-Brown) = 0.106
- height (Pee-Wee-Reese) = 6.1

# Approaches to Knowledge Representation

- Inferential Knowledge
  - Facts represented in a logical form, which facilitates reasoning.
    - e.g. Predicate logic
  - An inference engine (procedure) is required.
    - Some of which reason **forward** from given facts to conclusions
    - Others reason **backward** from desired conclusions to given facts
    - Most commonly used procedure:
      - **Resolution (Proof by contradiction)**

# Approaches to Knowledge Representation

- Inferential Knowledge

| Facts | Representation in Predicate logic |
|---|---|
| Wonder is a dog. | ***dog (wonder)*** |
| All dogs belong to the class of animals. | $\forall$ ***x : dog (x) → animal(x)*** |
| All animals either live on land or in water. | $\forall$ ***x : animal(x) → live (x, land) V live (x, water)*** |

- We can infer from these three statements that:

***" Wonder lives either on land or in water."***

# Approaches to Knowledge Representation

- Procedural Knowledge
  - Specifies "what to do when" and the representation is of "how to make it" rather than "what it is".
  - Can be represented in programs in many ways
    - E.g. writing **Prolog** Code for doing something
  - May have inferential efficiency, but
    - No inferential adequacy
      - Because it is very difficult to write a program that can reason about another program's behavior
    - No acquisitional efficiency.
      - Because the process of updating and debugging large pieces of code becomes unwieldy

# Issues in Knowledge Representation

- The fundamental goal of KR
  - is to facilitate inferencing (conclusions) from knowledge.

- Some issues that arise while using KR techniques are:
  - Important attributes
  - Relationship among attributes
  - Choosing the granularity of representation
  - Representing sets of objects
  - Finding the right structures as needed

# Knowledge Representation Schemes

- four general schemes for representing knowledge
  1. logic
  2. production rules
  3. semantic networks
  4. frames

# KR using Logic

- Logics
  - are formal languages for representing information such that conclusions can be drawn
  - concerned with the truth of statements about the world
  - each statement is either *TRUE or FALSE*.
  - includes :
    - Syntax
    - Semantics
    - Inference Procedure.

# KR using Logic

- Syntax
  - defines the sentences in the language
  - Specifies the *symbols in the language about how they can be combined* to form sentences.
- Semantics
  - define the "meaning" of sentences
  - Specifies how to assign a truth value to a sentence based on its *meaning in the world*
  - it may be *TRUE or FALSE*
- Inference Procedure.
  - Specifies *methods for computing new sentences from an existing* sentences.

# KR using Logic

- Logics are of different types
  - Propositional logic
  - Predicate logic
  - Higher order predicate logic
  - Fuzzy logic
  - Temporal logic
  - Modal logic

# Propositional Logic

- is fundamental to all logic that tells
  - the ways of joining and/or modifying entire propositions to form more complicated propositions
  - the logical relationships and properties that are derived from the methods of combining or altering statements

- *A proposition*
  - *is a statement (in English, a declarative sentence)*
  - *Every proposition is either* TRUE *or* FALSE

# Propositional Logic

- Deals with statements or proposition and connections between them

- Propositional logic represents a complete statement by a single symbol
  - It does not consider smaller parts of statements and treat simple statements as indivisible (atomic)

# Propositional Logic

- Syntax
  - **The symbols in this logic are**
    1. **Uppercase letter** (may also contain other letters or subscripts) such as P, Q, R, $W_{1,3}$ and North
       - which represent propositions such as: "It is raining" and "I am wet“
    2. **connectives**
       - and ($\wedge$), or ($\vee$), implies ($\rightarrow$), not ($\neg$), if and only if ($\leftrightarrow$)
    3. **Parenthesis:** ()
    4. **Logical constants:** True and False

# Propositional Logic

- Syntax
  - Truth tables for and (∧), or (∨), implies (→), not (¬), if and only if (↔)

| P | $\neg P$ |
|---|---|
| T | F |
| F | T |

| P | Q | $P \vee Q$ |
|---|---|---|
| T | T | T |
| T | F | T |
| F | T | T |
| F | F | F |

| P | Q | $P \wedge Q$ |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | F |
| F | F | F |

# Propositional Logic

- Syntax
  - Truth tables for and (∧), or (∨), implies (→), not (¬), if and only if (↔)

| P | Q | $P \to Q$ |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | T |
| F | F | T |

| P | Q | $P \leftrightarrow Q$ |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | F |
| F | F | T |

# Propositional Logic

- Semantics
  - are rules about how to assign truth values to a sentence if we know whether the propositions mentioned in the sentence are true or not
  - For instance, the sentence $P \wedge Q$ is true only in the situation when both $P$ and $Q$ are true
  - In this logic, the semantics or meanings of a sentence or formula is just a value TRUE or FALSE

# Propositional Logic

- Syntax

| $P$ | $Q$ | $\neg P$ | $P \wedge Q$ | $P \vee Q$ | $P \Rightarrow Q$ | $P \Leftrightarrow Q$ |
|---|---|---|---|---|---|---|
| *false* | *false* | *true* | *false* | *false* | *true* | *true* |
| *false* | *true* | *true* | *false* | *true* | *true* | *false* |
| *true* | *false* | *false* | *false* | *true* | *false* | *false* |
| *true* | *true* | *false* | *true* | *true* | *true* | *true* |

For complex sentences, we have five rules, which hold for any subsentences P and Q in any model m (here "iff" means "if and only if"):

- ¬P is true iff P is false in m.
- P ∧ Q is true iff both P and Q are true in m.
- P ∨ Q is true iff either P or Q is true in m.
- P → Q is true unless P is true and Q is false in m.
- P ↔ Q is true iff P and Q are both true or both false in m.

# Propositional Logic

- Inference rules

$$
\begin{array}{rcll}
(\alpha \land \beta) & \equiv & (\beta \land \alpha) & \text{commutativity of } \land \\
(\alpha \lor \beta) & \equiv & (\beta \lor \alpha) & \text{commutativity of } \lor \\
((\alpha \land \beta) \land \gamma) & \equiv & (\alpha \land (\beta \land \gamma)) & \text{associativity of } \land \\
((\alpha \lor \beta) \lor \gamma) & \equiv & (\alpha \lor (\beta \lor \gamma)) & \text{associativity of } \lor \\
\neg(\neg\alpha) & \equiv & \alpha & \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) & \equiv & (\neg\beta \Rightarrow \neg\alpha) & \text{contraposition} \\
(\alpha \Rightarrow \beta) & \equiv & (\neg\alpha \lor \beta) & \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) & \equiv & ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)) & \text{biconditional elimination} \\
\neg(\alpha \land \beta) & \equiv & (\neg\alpha \lor \neg\beta) & \text{De Morgan} \\
\neg(\alpha \lor \beta) & \equiv & (\neg\alpha \land \neg\beta) & \text{De Morgan} \\
(\alpha \land (\beta \lor \gamma)) & \equiv & ((\alpha \land \beta) \lor (\alpha \land \gamma)) & \text{distributivity of } \land \text{ over } \lor \\
(\alpha \lor (\beta \land \gamma)) & \equiv & ((\alpha \lor \beta) \land (\alpha \lor \gamma)) & \text{distributivity of } \lor \text{ over } \land
\end{array}
$$

Standard logical equivalences. The symbols α, β, and γ stand for arbitrary sentences of propositional logic.

# Propositional Logic

- Inference rules

| | | | |
|---|---|---|---|
| Modus Ponens | $p \implies q$<br>$p$<br>$\therefore q$ | Modus Tollens | $p \implies q$<br>$\sim q$<br>$\therefore \sim p$ |
| Elimination | $p \vee q$<br>$\sim q$<br>$\therefore p$ | Transitivity | $p \implies q$<br>$q \implies r$<br>$\therefore p \implies r$ |
| Generalization | $p \implies p \vee q$<br>$q \implies p \vee q$ | Specialization | $p \wedge q \implies p$<br>$p \wedge q \implies q$ |
| Conjunction | $p$<br>$q$<br>$\therefore p \wedge q$ | Contradiction Rule | $\sim p \implies F$<br>$\therefore p$ |

# Propositional Logic

- Inference rules
  - **Horn clause**
    - A *Horn clause* is a clause with at most one positive literal.
    - Example: $(P \vee \neg Q) \wedge (\neg P \vee S)$
    - In **Resolution**,

$$\frac{A \vee B, \quad \neg B \vee C}{A \vee C}$$

# Propositional Logic

- Example 1

| Propositions | Symbols |
|---|---|
| It is raining. | R |
| I am wet. | W |
| I am annoyed | A |

Then "I always get wet and annoyed when it rains“ is represented in this logic as

$$R \rightarrow W \land A$$

# Propositional Logic

- Example 2
- Suppose a statement

$$((P \land \neg Q) \to R) \lor Q$$

- Let the interpretation for this statement assigns ‘T’ to P and ‘F’ to Q and ‘F’ to R, then the semantics of the given statement is determined as:

$$
\begin{aligned}
&((P \land \neg Q) \to R) \lor Q \\
&\equiv ((T \land \neg F) \to F) \lor F \\
&\equiv ((T \land T) \to F) \lor F \\
&\equiv (T \to F) \lor F \\
&\equiv F \lor F \\
&\equiv F
\end{aligned}
$$

# Propositional Logic

- Example 3
- Suppose there are two restaurants. The first has a signboard saying that “good food is not cheap food” and other has signboard saying that “cheap food is not good”. Do they saying the same?
- Solution

Let

G: food is good.

C: Food is cheap.

Then

Good food is not cheap: $G \rightarrow \neg C$

Cheap food is not good: $C \rightarrow \neg G$

# Propositional Logic

- Example 2
  - Truth table for *inference*

| G | C | ¬ G | ¬ C | G → ¬ C | C → ¬ G |
|---|---|---|---|---|---|
| F | F | T | T | | |
| F | T | T | F | | |
| T | F | F | T | | |
| T | T | F | F | | |

  - The two statements have the same truth value
    - Both statements are saying same

# Propositional Logic

- Example 2
  - Truth table for *inference*

| G | C | ¬ G | ¬ C | G → ¬ C | C → ¬ G |
|---|---|---|---|---|---|
| F | F | T | T | T | T |
| F | T | T | F | T | T |
| T | F | F | T | T | T |
| T | T | F | F | F | F |

  - The two statements have the same truth value
    - Both statements are saying same

# Propositional Logic

- Example 3
  - Write each sentence in symbols, assigning propositional variables to statements as follows:
    - Today is hot but it is not sunny
    - It is neither hot nor sunny.
    - You can use the microlab only if you are a cs major or not a fresh- man.
    - If it snows or rains today, I will not go for a walk.

# Propositional Logic

- Example 4
  - Consider a propositional language with three propositional constants - ***mushroom*, *purple*, and *poisonous*** - each indicating the property suggested by its spelling. Using these propositional constants, encode the following English sentences as Propositional Logic sentences.
    - *All purple mushrooms are poisonous.*
    - *A mushroom is poisonous only if it is purple.*
    - *A mushroom is not poisonous unless it is purple.*
    - *No purple mushroom is poisonous.*

# Propositional Logic

- Example 4
  - Consider a propositional language with three propositional constants - ***mushroom*, *purple*, and *poisonous*** - each indicating the property suggested by its spelling. Using these propositional constants, encode the following English sentences as Propositional Logic sentences.

    - *All purple mushrooms are poisonous.*
      - ***purple*** ∧ **mushroom** → **poisonous**
    - *A mushroom is poisonous only if it is purple.*
      - **mushroom** ∧ **poisonous** → ***purple***
    - *A mushroom is not poisonous unless it is purple.*
      - **mushroom** ∧ **poisonous** → ***purple***
    - *No purple mushroom is poisonous.*
      - ¬ **(*purple*** ∧ **mushroom** ∧ **poisonous)**

# Propositional Logic

- **Some terminologies**
  - **Tautology**
    - A proposition that is always true is called a *tautology*.
    - e.g., ***(P V ¬P) is always true regardless of the truth value of the*** proposition ***P***.
  - **Contradictions**
    - A proposition that is always false is called a *contradiction*.
    - e.g., ***(P ∧ ¬P) is always false regardless of the truth value of*** the proposition ***P***.
  - **Contingencies**
    - A proposition is called a *contingency, if that proposition is neither* a *tautology nor a contradiction*
    - e.g., ***(P V Q) is a contingency.***

, ≡

# Propositional Logic

- **Some terminologies**
  - **Equivalent**
    - Two statements are *logically equivalent "if and only if" their* truth table columns are identical;
    - Example: $P \rightarrow Q$ and $\neg P \vee Q$ are equivalent.

| P | Q | $P \rightarrow Q$ | $\neg P$ | $\neg P \vee Q$ |
|---|---|---|---|---|
| T | T | T | F | T |
| T | F | F | F | F |
| F | T | T | T | T |
| F | F | T | T | T |

# Propositional Logic

- **Examples**
  - Show that $(P \wedge Q) \rightarrow P$ is a tautology.
  - Show $\neg P \rightarrow (P \rightarrow Q)$ is a tautology
  - $P \rightarrow \neg Q$ and $Q \rightarrow \neg P$ are equivalent

, ≡

# Assignment #2

Question 1: You are given a note on propositional theory proving method which you can find in the lecture materials (File name: Resolution and Refutation in propositional logic.pdf).

Suppose the following propositions:

, ≡

S: I study

G: I get good grades

E: I enjoy

1. If I study I make good grades.
2. If I do not study I enjoy.

Using resolution, prove that either I make good grades or I enjoy.

# Questions

1. Define and differentiate data, information, knowledge and wisdom with an example
2. Differentiate between tacit and explicit knowledge.
3. What are requirements of a good knowledge representation system?
4. What are the different approaches of knowledge representation?
5. What may be the issues in knowledge representation?
6. What are the general schemes for representing knowledge?
7. What are the drawbacks of propositional logic?

THANK YOU

End of Chapter