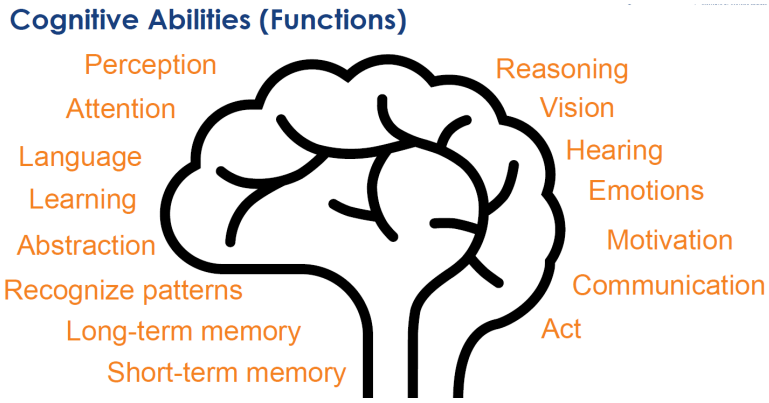
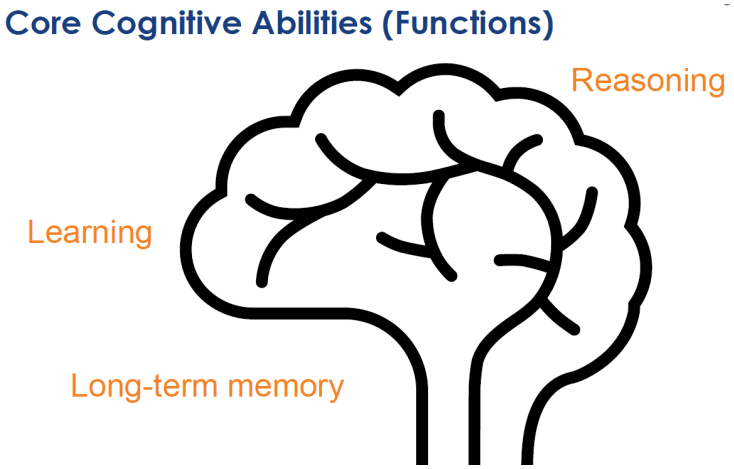
Machine Reasoning

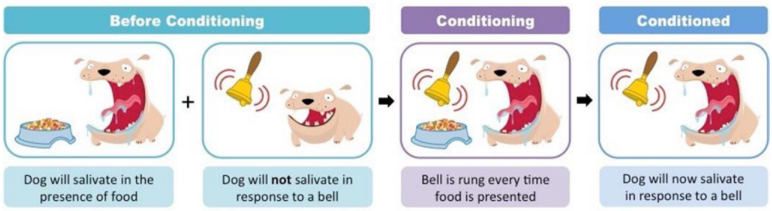
- What's intelligence?





- "Intelligence measures a human/AI's ability to achieve goals in a wide range of environments
- Creating an AI is analogical to defining goal & designing KPIs for an employee to behave autonomously (through learning and reasoning using domain knowledge) without the boss/designer's micro management (explicitly programmed)
- Learning
  - Habituation

    - Bird learned the knowledge/behaviour to ignore the fake threat;
    - Unsupervised Learning; Anomaly Detection
  - Classical (Reflex) conditioning



- Dog learned the knowledge of association/correlation: between food & bell ring;
- Dog learned the knowledge/behaviour of causation: food/bell ring → saliva

- Association (between stimuli or events); Supervised Learning

- Operant conditioning
  - Reinforcement Learning; Generate & Test; Simulation
- Observational learning

  - Imitation Learning; Unsupervised Learning; (copycat learning)

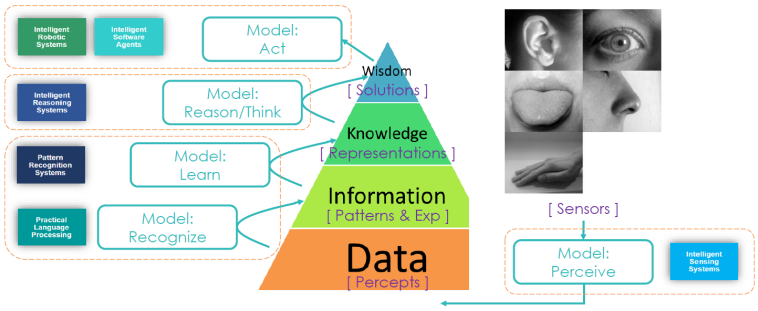
- Reasoning
  - Deductive Reasoning
    - Aristotle's syllogism; Formal logic; Knowledge Graph; If Then business rules; Declarative programing language like SQL; (Universal -> Individuals)
  - Inductive Reasoning
    - Use meta knowledge to generate new knowledge using statistical method: learning / pattern recognition algorithms; central limit theorem; regression; (Individuals -> Universal)
  - Analogical Reasoning
    - Similarity based reasoning; Case based; K nearest neighbour; (including customer profiling for recommendation; even stereotyping)
  - Abductive Reasoning
    - Probabilistic calculation; Prior/Conditional/Joint probability; Bayesian network; (Hypothesis ~ Evidence)

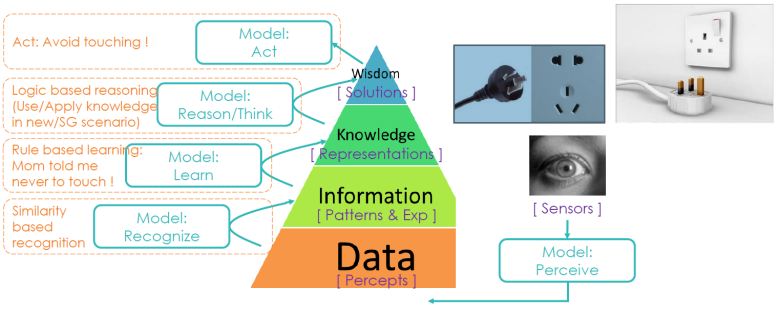
- Intelligent System
  - Model type
    - Conceptual Model

- Technically, it's a function (white or black box), which maps input(s) to output(s)

- Physical model
  - A model could be considered just as a tangible text file stored in computer/server, e.g. model.txt

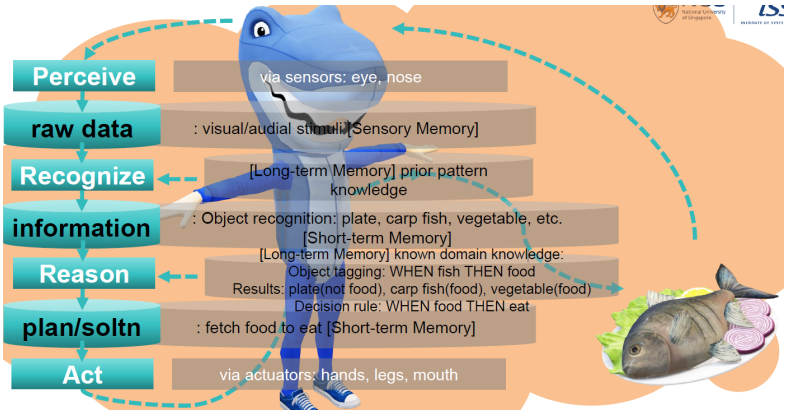
- Cognitive Processes To Solve Problem

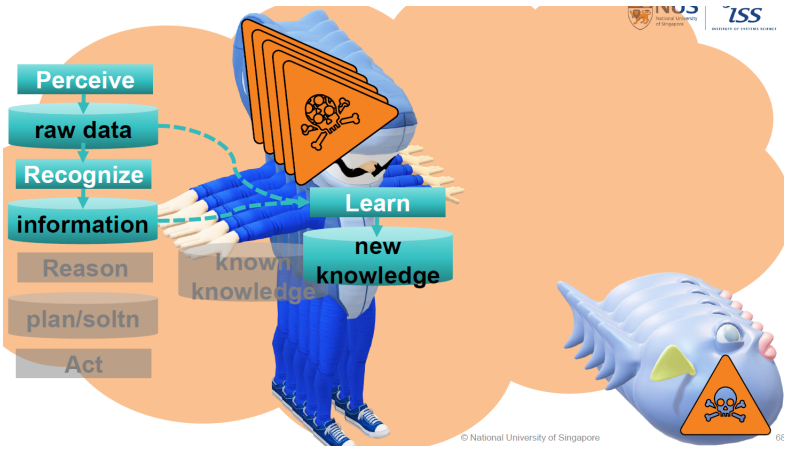


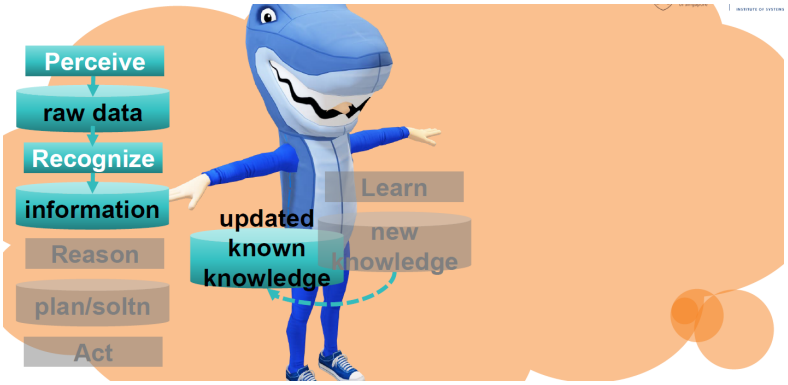


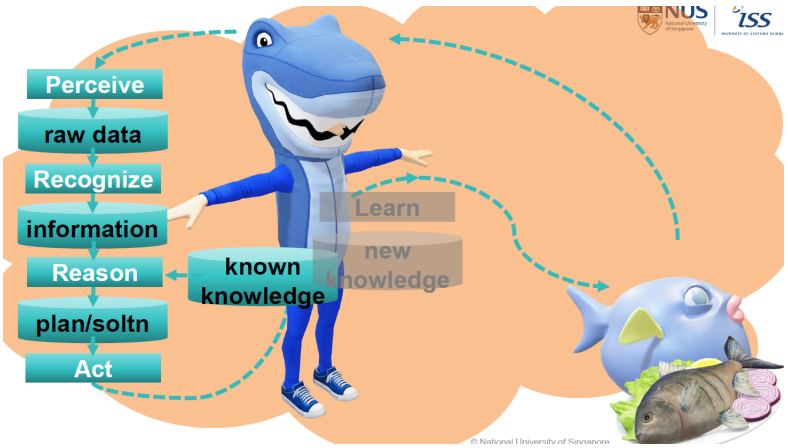
- Cognition definition
  - Cognition is "the mental action or process of acquiring knowledge and understanding through thought (recognition, learning, computation, reasoning & thinking)thinking), experience (information & data), and the senses (perceptions & It encompasses many aspects of intellectual functions and processes such as attention, the formation of knowledge, memory and working memory, judgment and evaluation, reasoning and " problem solving and decision making, comprehension and production of language.
  - Cognitive processes use existing knowledge (computation, reasoning & thinking) and generate new knowledge (learning supported by pattern recognition using experience/information from data/perceptions/sensors)

- Pipeline





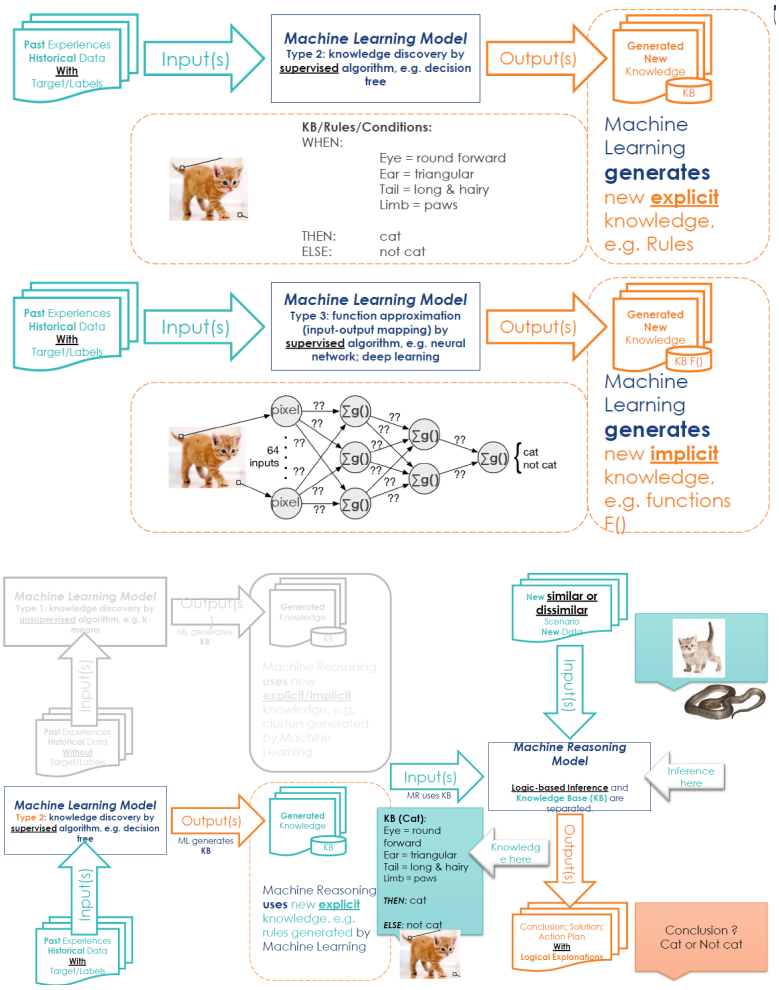


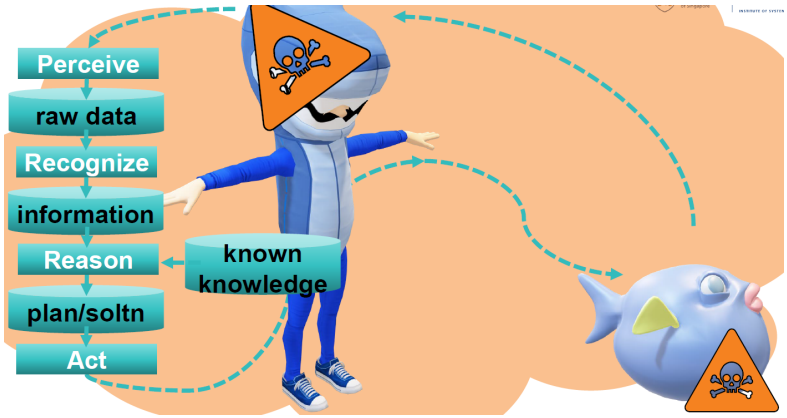


- 1.Perceive : Gather raw observable information, e.g. problem goal, type of known creatures, test creatures,decision/action required.
- 2.Recognise : Recognise patterns and decomposed relevant information, e.g. different colours, triangle#, teeth#,
- 3.Learn : Invent hypothesis / candidate theories, e.g.hypothesis 1: alien must has eyes >=2 & not in grey & no horn& target types to generalize/create new

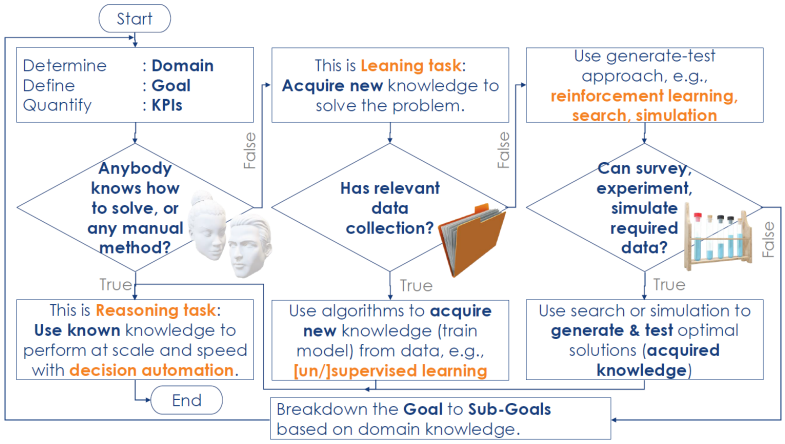
rules,

- 4.Reason : Use test data to evaluate all hypotheses /candidate theories to obtain correct knowledge.
- 5.Act : Make choice, observe/bear consequences (scores).

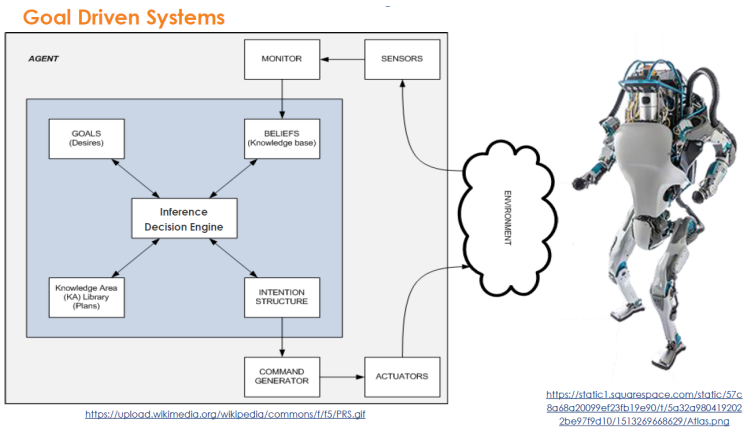
- Learning+ Reasoning
  - 

  - 

- How can I apply AI to my business problem?



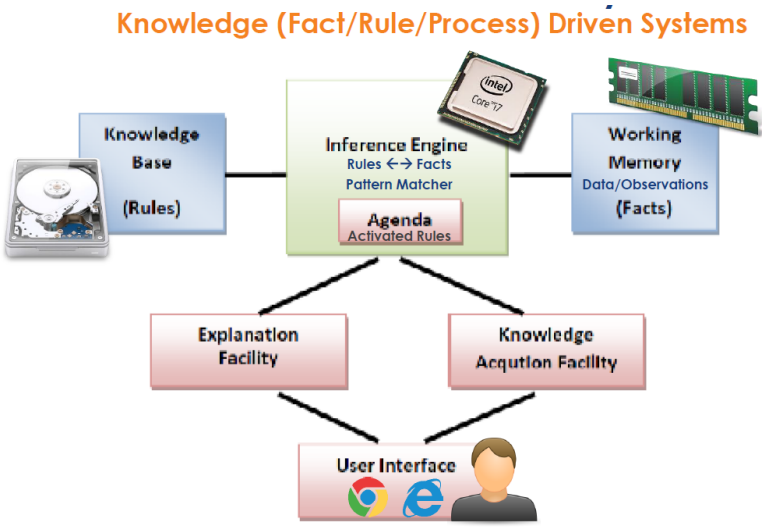
- Data driven vs. Goal driven Systems
  - Proactive Reasoning Systems (Goal Driven)
    - Autonomous Software System (Sales Chabot, Robotic ProcessAutomation)
    - Multi Agent Cooperative System (Warehouse Robotic Swarm,Coordinated Robotic Vacuum Cleaners)
    - Constrain Solver (Real time Delivery Scheduler
    - example



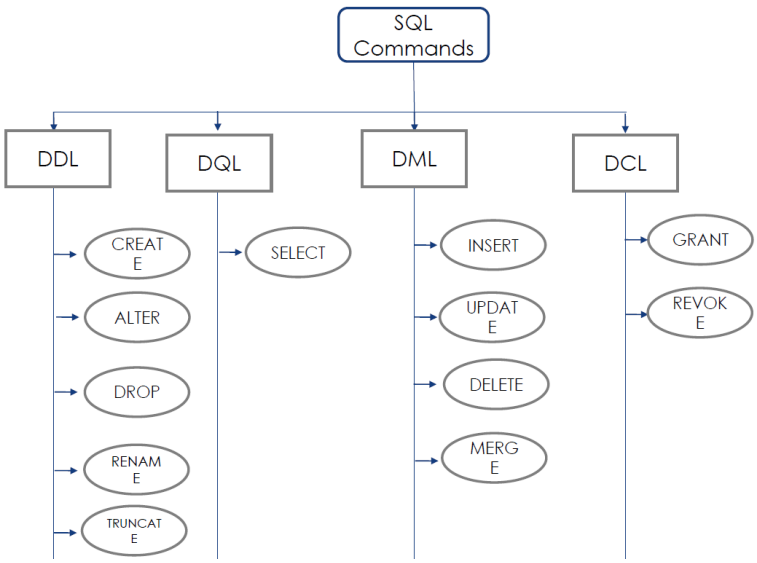
https://upload.wikimedia.org/wikipedia/commons/f/f5/PRS.gif

https://static1.squarespace.com/static/57c8a68a20099ef23fb19e90/t/5a32a980419202 2be97f9d10/1513269668629/Atlas.png

  - Reactive Reasoning Systems (Data/Event Driven)
    - Business Rule Management System (BRMS)
    - Business Process Management System(BPMS)
    - Constrain Solver (Travel Planner)
    - example

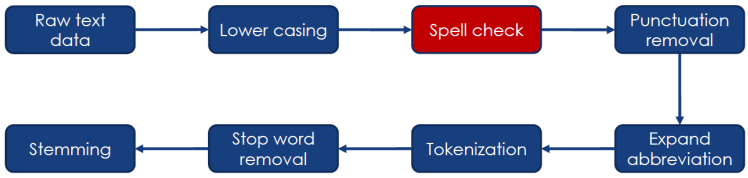


- Database
  - A database is an organized collection of structured information, or data,typically stored electronically in a computer system.
  - A database is usually controlled by a database management System.
  - What is RDBMS ?
    - A software system used to maintain relational databases is a relational database management system (RDBMS).
    - The data in a RDBMS is stored in database objects which are called tables. This table is a collection of related data entries and it consists of numerous columns and rows.
    - RDBMS is the basis (run time environment) for SQL execution.
  - SQL

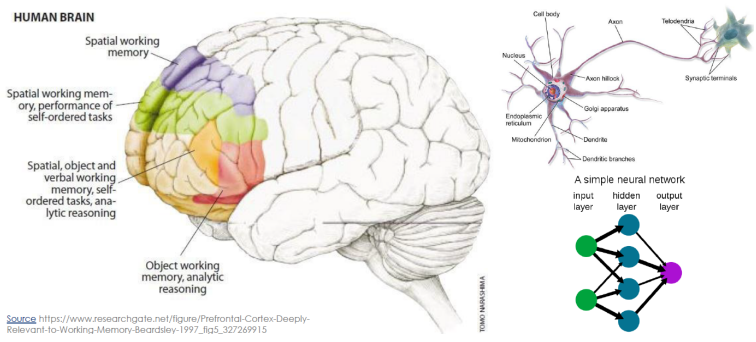


    - Operations on database is carried out using certain SQL commands. They are mainly categorized as:
      - DDL: Data Definition Language
        - DDL or data definition Language mainly focuses on creation and modification of database objects.

            - Create: Create the database or its objects
            - Alter: Alter the structure of database
            - Drop: Delete database or database objects
        - DQL: Data Query Language
            - This is Used to retrieve data from the database :
            - E.g. Select field_name from database_name where condition ;
        - DML: Data Manipulation Language
            - Insert: addition of records to the existing table
            - Update: updating existing records of a table
            - Delete: deletion of a record from a table
        - DCL: Data Control Language
            - GRANT: Gives user access to DB
            - REVOKE: withdraw user's access to a particular DB that was granted access
    - Table Keys
        - Often, we have to maintain relationship with other tables and also provide uniqueness to fields in our existing table.
            - Primary Key: A Primary key uniquely identifies records in a table
            - Unique Key: same as primary key, but allows a null value
            - Foreign Key: A primary key of another table is referred to by this table as foreign key (one field/column).
- Text Pre processing



- Knowledge Representation
    - Knowledge representation in human brain (black)



    - Knowledge representation in machine (white box): A scheme/method that allows the computer system to use or manipulate it to reason and solve problems

- Unlike humans, knowledge must be 'transplanted/saved' into machine reasoning system/memory
- Representation goes hand-in-hand with reasoning/inference mechanism (computer algorithm)

  Data Structure — Processing Logic
- Large amount of knowledge is usually needed to solve complex problems

- **Explicit knowledge representation (think of documentation) enables business knowledge management and retention.**

- Formal Logic
  - Propositional Logic
    - The syntax of propositional logic expression is constructed using propositions and connectives

And

p	q	$p \cdot q$
T	T	T
T	F	F
F	T	F
F	F	F

Or

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

If . . . then

p	q	$p \supset q$
T	T	T
T	F	F
F	T	T
F	F	T

Not

p	$\sim p$
T	F
F	T

    - All propositions must be either true or false (referred to as truth value of the proposition)
  - Connectives
    - $\neg$ negation "not"
    - $\vee$ disjunction "or"
    - $\wedge$ conjunction "and"
    - $\rightarrow$ implication "if … then"
    - $\leftrightarrow$ bi-conditional "if and only if"
  - Complex expressions using connectives

    $p \wedge q$ = "Sam has flu. **AND** Sam has cough."
    $p \wedge \neg q$ = "Sam has flu. **AND** Sam has no cough."
    $p \rightarrow q$ = "**IF/WHEN** Sam has flu **THEN** Sam has cough.
  - Logical equivalence

Two sentences are logically equivalent is written as: **α ≡ β** (iff α ⊨ β and β ⊨ α)

$(\alpha \wedge \beta)$	$\equiv (\beta \wedge \alpha)$	commutativity of $\wedge$
$(\alpha \vee \beta)$	$\equiv (\beta \vee \alpha)$	commutativity of $\vee$
$((\alpha \wedge \beta) \wedge \gamma)$	$\equiv (\alpha \wedge (\beta \wedge \gamma))$	associativity of $\wedge$
$((\alpha \vee \beta) \vee \gamma)$	$\equiv (\alpha \vee (\beta \vee \gamma))$	associativity of $\vee$
$\neg(\neg\alpha)$	$\equiv \alpha$	double-negation elimination
$(\alpha \Rightarrow \beta)$	$\equiv (\neg\beta \Rightarrow \neg\alpha)$	contraposition
$(\alpha \Rightarrow \beta)$	$\equiv (\neg\alpha \vee \beta)$	implication elimination
$(\alpha \Leftrightarrow \beta)$	$\equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$	biconditional elimination
$\neg(\alpha \wedge \beta)$	$\equiv (\neg\alpha \vee \neg\beta)$	de Morgan
$\neg(\alpha \vee \beta)$	$\equiv (\neg\alpha \wedge \neg\beta)$	de Morgan
$(\alpha \wedge (\beta \vee \gamma))$	$\equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	distributivity of $\wedge$ over $\vee$
$(\alpha \vee (\beta \wedge \gamma))$	$\equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	distributivity of $\vee$ over $\wedge$

- First Order Logic (Predicate Calculus)
  - **Objects (Class)** : people, houses, numbers, colors, baseball games, …
  - **Constants (Instance)** : The White House, Sam GU Zhan, π, NUS,...
  - **Variables** : x, y, a, b, ...
  - **Relations (Predicate)** : is student, has leg, eats, is bigger than, is part of, is red, is round, prime to, come between, is one more than, …
  - **Functions (Predicate)** : father of, best friend, square root of, sum of, …
  - **Connectives** : ¬ → ∧ ∨ ↔
  - **Equality** : =
  - **Quantifiers** : ∀ ∃
  - Sentences of First Order Logic
    - Term (noun) is an expression that refers to an

      FatherOf(DiDi) : "(a person, who is) father of DiDi"

      ¬ FatherOf(DiDi) : "(a person, who is) not father of DiDi"

      FatherOf(x) : "someone's father"

      Sam, Jessie, DiDi, Machine Reasoning Course, PhD, …

      Integer: x, y, z (variables of object: all integer numbers)
    - Atomic Sentence (semantics) is formed from one predicate symbol followed by one parenthesized list of terms

      Relational predicate
      - IsFriend(Jessie, Sam) : Jessie is friend to Sam.

      Functional predicate
      - FatherOf(DiDi) : DiDi's father (Result is Sam.)

      Embedded
      - IsFriend(Jessie, FatherOf(DiDi)) : Jessie is friend to DiDi's father.
    - Complex Sentence (semantics) is made from Atomic Sentences using logical connectives

      IsClassmate(Jessie, Sam) ∧ TalksTo(Jessie, Sam) → IsFriend(Jessie, Sam)
  - Establish Truth of Predicate
    - Predicates need to be propositionalized for use in reasoning
    - Method 1 : Assign specific value to predicate expressions (similar to instantiation)

      StudyAt(x, NUS) → Smart(x) What's the scope of x?

      x = Sam

      Conclusion: StudyAt (Sam, NUS) → **Smart(Sam)**
    - Method 2A : Universal quantifier:

      Method 2A: Universal quantifier: **∀**

      We want to express "Everyone studying at NUS is smart."

      √ ∀x StudyAt(x, NUS) → Smart(x) : "For everyone, if the person is studying at NUS then the (**same**) person is smart" (For those are not studying at NUS, we don't know.)

      × ∀x StudyAt(x, NUS) ∧ Smart(x) : "Everyone (all persons in Singapore) is studying at NUS and all (**these**) persons are smart." **incorrect semantic**

- Method 2B : Existential quantifier:

  Method 2B: Existential quantifier: ∃

  We want to express "Someone studying at NUS is smart."

  √ $\exists x\ StudyAt(x, NUS) \wedge Smart(x)$: "There is someone studying at NUS and this (**same**) person is smart."

  × $\exists x\ StudyAt(x, NUS) \rightarrow Smart(x)$: "There is someone, when he/she is studying at NUS then he/she is smart." (When this (**same**) person is having a rest, then he/she may not be smart.) **incorrect semantic**

- Forms of Knowledge Representation (Rules)
  - Rules type
    - Represent problem solving knowledge as "IF/WHEN … THEN …" rules
    - Basic Rule Syntax

      IF <antecedent>
      THEN <consequent>

      IF person X is ill
      THEN person X need rest a lot

    - Multi antecedent Rule

      IF <antecedent 1>
      AND/OR <antecedent 2>
      …
      AND/OR <antecedent n>
      THEN <consequent>

      IF person X is ill
      AND person X is a lecturer
      THEN person X cannot rest at home, but go to class

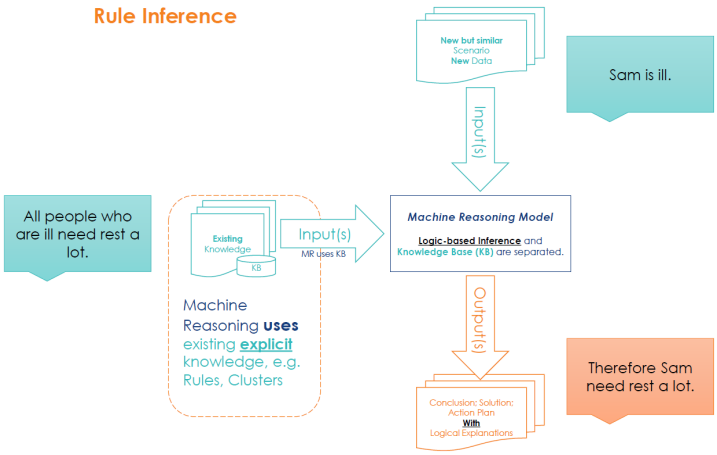
    - Multi consequent Rule

      IF <antecedent 1>
      THEN <consequent 1>
      <consequent 2>
      …
      <consequent m>

  - Rule Inference

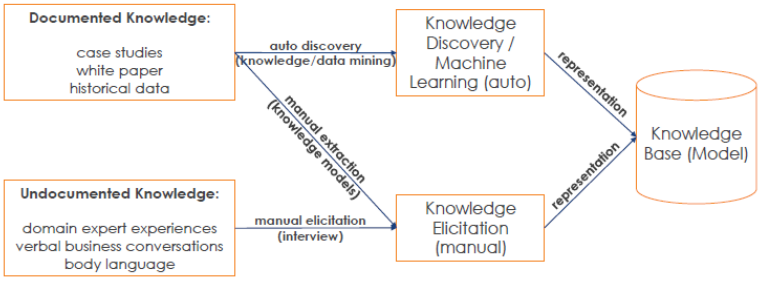
- **Knowledge/Rule** : **All people who are ill** need rest a lot.
- **Individual 1** : **Sam** is **ill**, therefore he need rest a lot.
- **Individual 2** : **Jessie** is **ill**, therefore she need rest a lot.
- **Individual …**

☺ **Reasoning Rationality: Universal → Individual**



- Rules (business knowledge) in Rule/Process Reasoning System are designed as mutually independent
  - Each rule represents a single chunk of knowledge
    - IF pet_size = medium THEN pet_recommend = cats or small dogs
- Rules are based on a priori knowledge or heuristics
  - Rules are derived from domain experts who uses experiential knowledge and “rules of thumb”
    - IF buyer = female THEN pet_recommend = hamster
- Rules can incorporate uncertainties
  - Real life business situations are plagued with uncertainties that make decision making difficult (or flexible)
  - IF buyer_work = long_hours THEN pet_recommend = dog (30% confidence in rule conclusion)

- Knowledge Acquisition
  - Knowledge Acquisition is the transfer and transformation of problem solving knowledge into a form that can be used to build intelligent systems.
  - Methods



- Elicitation
  - Elicitation is acquisition of tacit knowledge from a subject matter expert
  - The Knowledge elicitation approach:
    - Capture knowledge using interviews
    - Interpret & Analyze the transcripts and data obtained
    - Build knowledge models (knowledge representation)
    - Use the knowledge models to guide further elicitation
    - Verify & Validate the captured knowledge
    - Stop when the knowledge model is enough for building business
    - reasoning system
  - Interview Best Practices
    - More Beneficial interviewing expert at their workplace
    - Make sure the meeting place is quiet and free from interruptions
    - Before the interview:
      - Do background research on the domain area
      - Background check on the domain expert
      - Design and phrase your questions
      - Email questions to domain expert
      - Acquire and prepare the tools for the interview
    - During the interview:
      - Introductions & Social preliminaries
      - State purpose of interview
      - Give a brief on the roles and responsibilities
      - Be courteous; Listen closely; Avoid arguments
      - Investigate each topic in detail
      - Evaluate session outcome
    - Observe confidentiality
- Knowledge Models (Templates for knowledge representation)
  - Definition

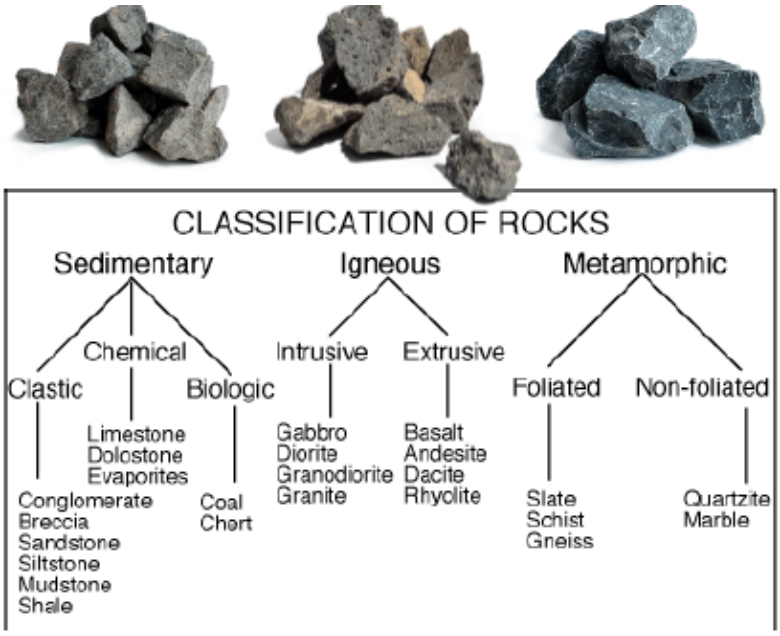
- A knowledge model is a group of structured representations of knowledge that allows us to better understand the domain and the processes involved in decision making.
- Documented models provide rich descriptions of domain knowledge that is independent of any particular software implementation.
- These models also serve as a basis for communication among stakeholders: experts , analysts, developers, and end

- Concept Dictionary

Concept	Synonyms Abbr	Meaning
Esophagus	Gullet Oesophagus	Sometimes known as the gullet. Muscular tube through which food passes from pharynx to the stomach
Duodenum		The first section of the small intestine
Peptic ulcer	PUD	Area of the gastrointestinal tract that is extremely painful. Mucosal erosions equal to or greater than 0.5cm.
Hyperacidity	Acid dyspepsia, Amalpitta	A condition of excreting more than the normal amount of hydrochloric acid in the stomach

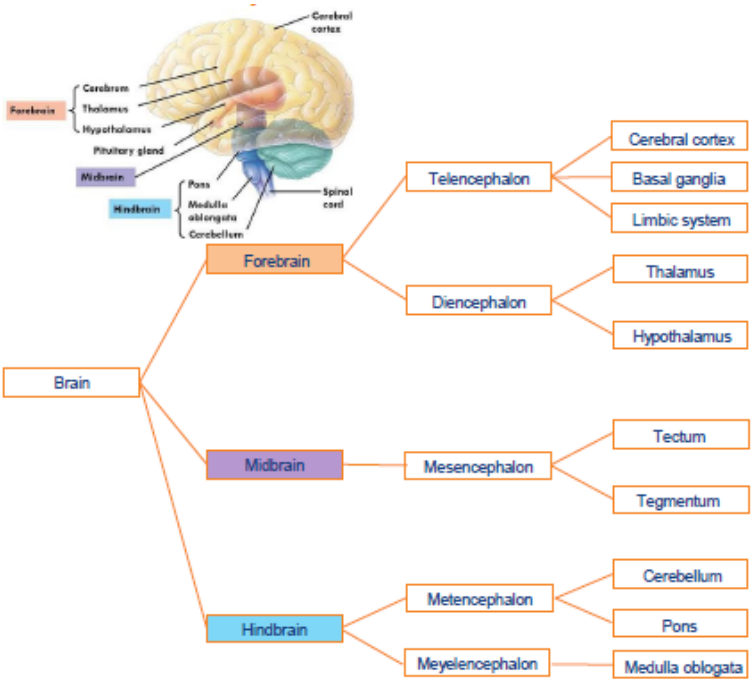
  - A concept dictionary contains the list of all relevant concepts that are used in the problem domain to solve the problem.
  - The dictionary provides a detailed explanation of the concept and can include any information that is useful for a good understanding of the concept.
  - It can be similar to a glossary
  - It does not have any particular format.

- Concept Tree



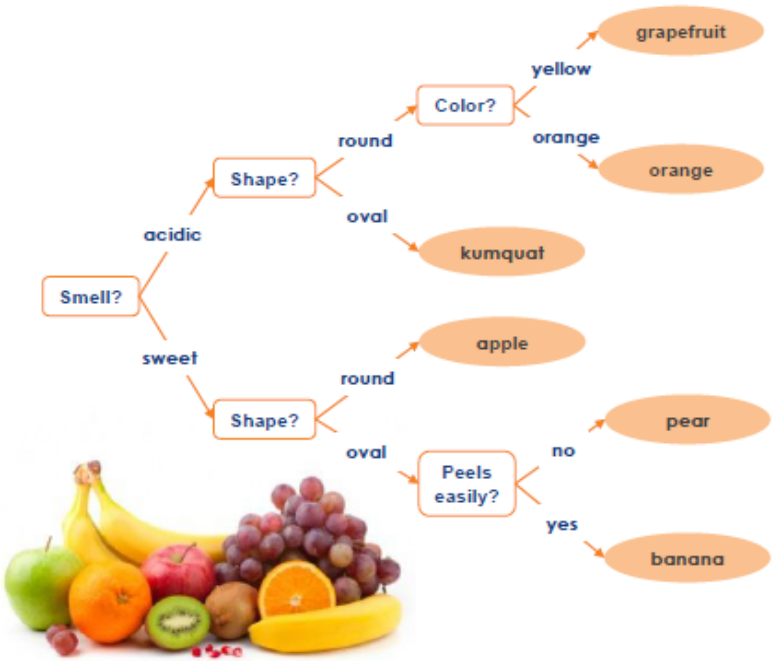
- Tree that shows concepts and the classes and sub classes.
- All relationships must be "is a"
- Check the tree by looking at the lowest and highest nodes and asking
- “is <sub-concept>a(type of) <concept>
- Nodes should have clear & complete names.
- Captured terminology and landscape of the domain.

- Composition Tree



- Detailed concept breakdowns into its constituent parts.
- All relationships must be “is part of”
- Understand things as
  - Products (parts of a machinery)
  - Organisations (your organisation chart)

    - Documents (the table of contents)
- Decision Tree



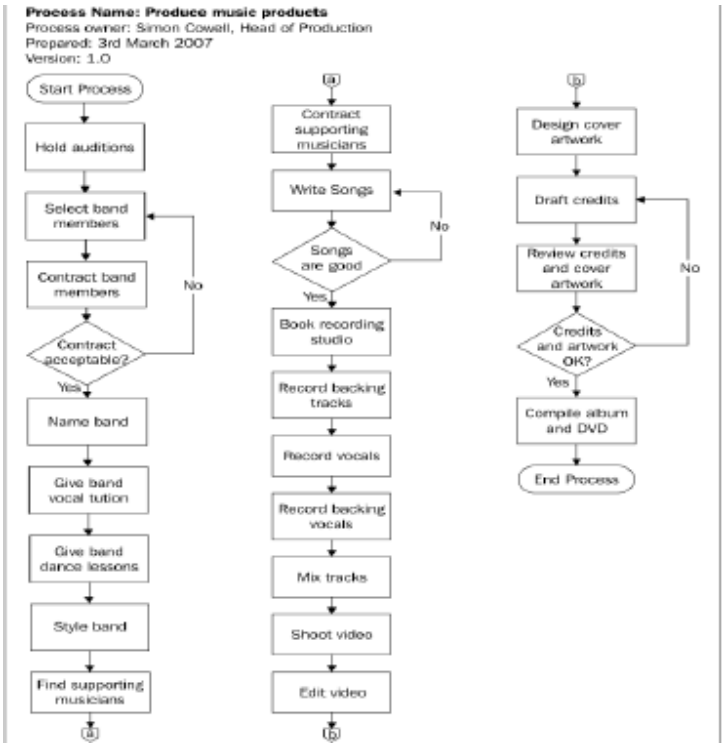
  - Tree shows the alternative courses of action or casual consequences for a particular decision.
  - Condition/Rule based domain knowledge
- Rules & Decision Table

Rule No.	Condition 1	Logical Operand	Condition 2	Sub-goal
F-1	franchise-fee ≤ threshold1	AND	royalty ≤ threshold2	Franchise = ok
F-2	franchise-fee ≤ threshold1	AND	royalty > 20% x franchise-fee	Franchise = not-ok
F-?	...	...	...	...

- Data Model

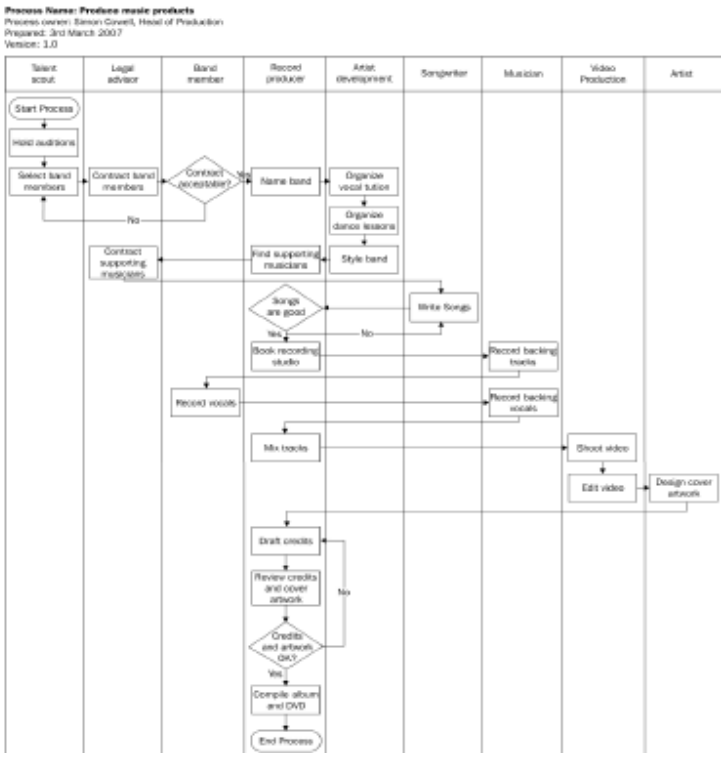
Sub-goal	Attribute	Inferable or Observable	KIE Field Type & Value			English Translation
KIE Data Model: Data Object	KIE Data Model: Object Field	KIE Form: User Interface	String, Integer, Float, Boolean, Date, etc.	Value Range	Value Unit	KIE Data Model: Comments
Franchise	franchise-fee	Observable	Float	1 - 50,000	SGD $	The price to be paid for the franchise
	royalty	Observable	Float	1 - 10,000	SGD $	The monthly fee payable to franchisor
	lease-period	Observable	Integer	1 - 5	years	The Franchise Lease period
Profit	**Income**	**Inferable**	Float	1 – 1,000,000	SGD $	Annual income from sale of goods
	Expenditure	Inferable	Float	1 – 1,000,000	SGD $	Annual expenditure from sale of goods
Income	goods	Observable	Float	1 – 1,000,000	SGD $	The sales proceeds from goods sold
	services	Observable	Float	1 – 1,000,000	SGD $	Sales proceeds from services rendered
Location	in-city	Observable	Boolean	True or False	N.A.	Planned shop in city area
	in-suburb	Observable	Boolean	True or False	N.A.	Planned shop in suburb area

- Flowchart



- The key to develop Flowchart to model/capture business process workflow, is to define the sequence of activities , and to identify those points where the flow can go two ways, depending on the circumstances.
- Write the activity name in as few words as possible, e.g. Verb + Noun pairs
- Write decision points as clear questions to which the answer is either “yes/true” or “no/false”.

- Activity Flow Diagram



- Activity Flow Diagram captures who does what (activity) activity)” in the workflow.
- Identify roles and responsibilities (Swimlanes)

    - A single activity should map to a single role . If this doesn't seem possible, then consider whether the activity should actually be split out into multiple activities.
    - Expand flowchart by drawing swimlanes for each activity under the identified roles/teams.
  - RACI Matrix

Process step / Role	Hold auditions	Select band members	Contract band members	Name band	Organise vocal tuition	Organise dance lessons	Stylise band	Find supporting musicians	Contract supporting musicians	Write songs	Book recording studio	Record backing vocals	Record vocals	Record backing vocals	Mix tracks	Shoot video	Edit video	Design cover artwork	Draft credits	Review credits and cover artwork	Compile album and DVD
Note	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Talent Scout	AR	AR	R					C													
Legal Adviser			AR						AR											R	
Band Member				I	I	I	I				I		R			R				R	
Record Producer			C	AR			C	AR	R	C	AR	AR	AR	AR	AR	R	C	R	AR	R	AR
Artist Development				C	AR	AR	AR			C								C		R	
Songwriter										AR										R	
Musician											I	R		R						R	
Video Production																AR	AR			R	R
Artist																		AR		R	

Key	
R - Responsible	Actually completes the activity - responsibility can be shared. Degree of responsibility is determined by the "A".
A - Accountable	Has Yes/No authority - there can only be one "A" per activity
C - Consulted	Involved prior to decision or action - two-way communication
I - Informed	Needs to know of the decision or action - one-way communication.

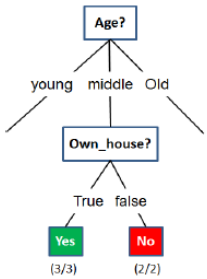
    - Responsible; Accountable; Consulted; Informed
      - R for responsible means, "the person who actually does the activity". Responsibility for an activity can be shared, if necessary.
      - A for accountable means, "the buck stops here", and the role has ultimate yes/no authority.
      - C for consulted means, "kept in the loop", and implies two-way communication prior to the activity.
      - I for informed means, "kept in the picture", and implies one-way communication after the activity.
    - Best Practices:
      - There can only be one accountability (A) per activity.
      - Recommend one responsibility (R) only per activity.
      - Roles can combine both accountability and responsibility for activities.
      - Minimize the number of consults (C) and informs (I).
      - Authority must accompany accountability.
      - Don't map decision points on the RACI matrix, only activities.
- Data Mining
  - Deductive Reasoning (Decision Rule/Tree Model)

- **Knowledge/Rule** : **All people** who are **ill**, they rest a lot.
- **Individual 1** : **Sam** is **ill**, therefore he rest a lot.
- **Individual 2** : **Jessie** is **ill**, therefore she rest a lot.
- **Individual ...**

- DT

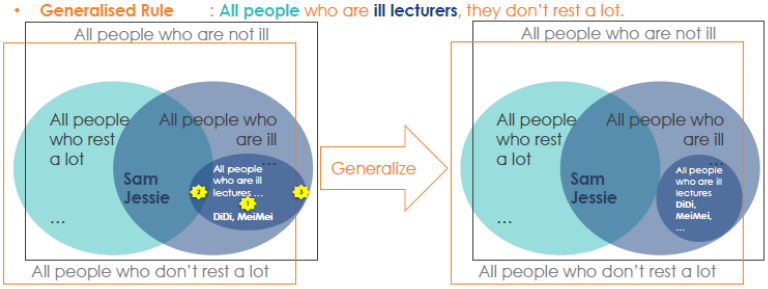
Decision Tree (DT) is a knowledge discovery method (supervised learning) used for classification and regression. The goal is to create a model (knowledge/rules) that predicts the value of a target variable by learning decision tree/rules inferred from the data features, through statistical calculation.

ID	Age	Has_job	Own_house	Credit_rating	Outcome
1	young	False	False	fair	No
2	young	False	False	good	No
3	young	True	False	good	Yes
4	young	True	True	fair	Yes
5	young	False	False	fair	No
6	middle	False	False	fair	No
7	middle	False	False	good	No
8	middle	True	True	good	Yes
9	middle	False	True	excellent	Yes
10	middle	False	True	excellent	Yes
11	old	False	True	excellent	Yes
12	old	False	True	good	Yes
13	old	True	False	good	Yes
14	old	True	False	excellent	Yes
15	old	False	False	fair	No

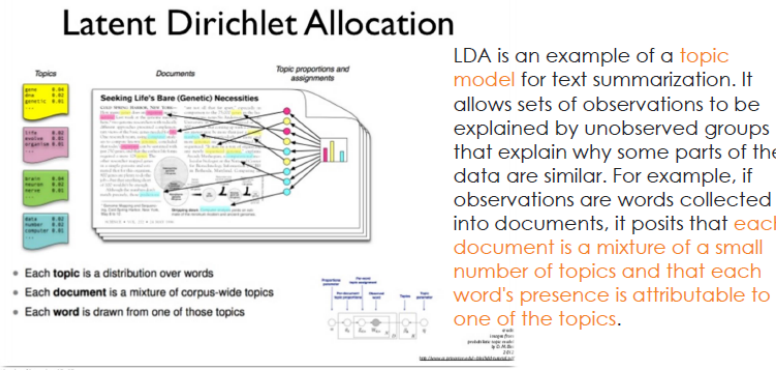


- Inductive Reasoning (Topic Summarization/Pattern Recognition)

- **Individual 1** : When **DiDi** is **ill** AND he is **lecturer**, he doesn't rest a lot.
- **Individual 2** : When **MeiMei** is **ill** AND she is **lecturer**, she doesn't rest a lot.
- **Generalised Rule** : **All people** who are **ill lecturers**, they don't rest a lot.



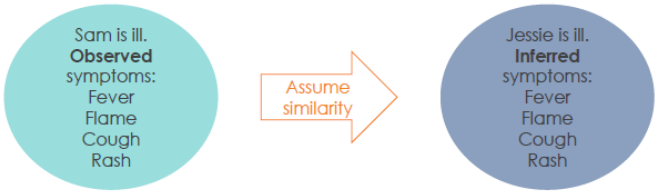
- Topic model



- documents. Topic modeling is a frequently used text mining tool for discovery of hidden semantic structures in a text body. Intuitively, given that a document is about a particular topic, one would expect particular words to appear in the document more or less frequently: "dog" and "bone" will appear more often in documents about dogs, "cat" and "meow" will

appear in documents about cats, and "the" and "is" will appear approximately equally in both.

- Analogical Reasoning (Similarity) (FAQ Bot: Approximate String Matching of Queries)
  - **Known case** : **Sam** is ill with his symptoms (features): fever, flame, cough, and rash.
  - **Inferred case** : **Jessie** is ill too, therefore she **would** have **same** symptoms as **Sam**: fever, flame, cough, and rash.

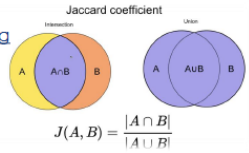


☺ **Reasoning Rationality: Known case → Inferred case**

  - Approximate String Matching

Approximate String Matching: In computer science, approximate string matching (often colloquially referred to as fuzzy string searching) is the technique of finding strings that match a pattern approximately (rather than exactly). A variation: similarity between two given (text) strings.

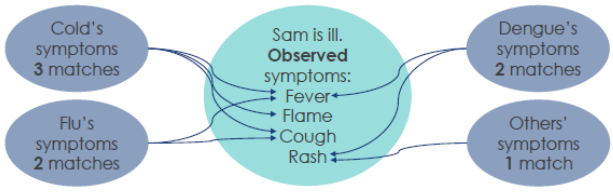
https://en.wikipedia.org/wiki/Approximate_string_matching



Sentence A:	This is machine reasoning course				
Sentence B:	This is reasoning systems course				
A ∩ B = 4 (unique tokens)	This	is	machine	reasoning	course
	This	is	reasoning	systems	course
A ∪ B = 6 (unique tokens)	This	is	machine	reasonoing	course
	This	is	reasoning	systems	course

J(A, B) = 66.67% similarity

- Abductive Reasoning:Probabilistic calculation; Prior/Conditional/Joint probability; Bayesian network; (Hypothesis ~ Evidence)
  - **Known observations** : **Sam** is ill with his symptoms (evidences): fever, flame, cough, and rash.
  - **Inferred root cause** (Hypothesis) : **Cold? Flu? Dengue? Others?**



☺ **Reasoning Rationality: Observations → Causes likelihood**

- Reasoning under Uncertainty
  - Analogical Reasoning (Similarity) (k Nearest Neighbors)
  - Approximate Reasoning (Fuzzy Logic)

Long Hair Group ← Hair length ≥ 10 cm

Hair length < 10 cm → Short Hair Group

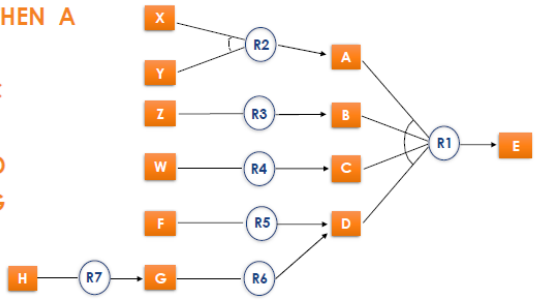
Long Hair Group ← Hair length is long

Hair length is short → Short Hair Group

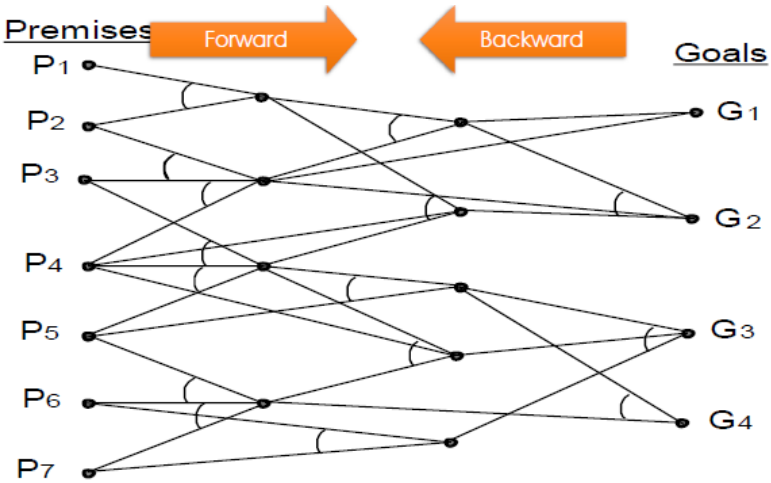
What if the hair length is both long and short → Which Group?

- Abductive Reasoning (Probability) (Bayes Model)

- Deductive Reasoning by Logical Inference
  - Rules & Logical Inference
    - Three main components of Rule Based System (RBS)
      - A set of domain rules/facts (Knowledge Base) [long term memory
        - Rules/Facts represent a chunk of problem solving knowledge
      - A working memory (WM) [short term memory
        - Contains new observation's Data (current state of program execution)
      - A rule/inference engine [pattern matching controller]
        - A computational system that implements the control strategy and applies (fire) the rules
        - The patterns in WM are matched against the conditions of the rules
        - Matched rules are called the conflict set
        - The control strategy determines the order in which the rules are fired and
        - resolves any rule conflicts
        - Uses the Recognise Act cycle
    - Rules form a Search Tree
      - R1 IF A and B and C and D THEN E
      - R2 IF X and Y THEN A
      - R3 IF Z THEN B
      - R4 IF W THEN C
      - R5 IF F THEN D
      - R6 IF G THEN D
      - R7 IF H THEN G



      - Inference strategy is known as "chaining".
    - Forward Chaining & Backward Chaining



- Forward Chaining(data-driven)
  - Begin with a set of facts in the Working Memory, then apply rules to generate new facts until the desired goal is reached.
  - This process continues until no more rules can be triggered/fired. The system then reports its conclusions.
  - The focus of attention starts form known data &business rules
    - e.g., object recognition, routine decisions
    - May do lots of work that is irrelevant to the goal
  - Forward Chaining Example

    Rule 1:
    IF the patient has a sore throat
    AND we suspect a Bacterial infection
    THEN we believe the illness is a strep throat

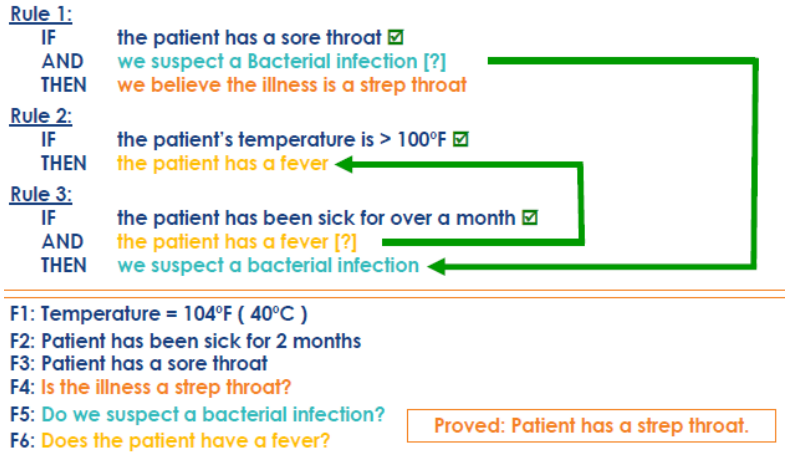
    Rule 2:
    IF the patient's temperature is > 100
    THEN the patient has a fever

    Rule 3:
    IF the patient has been sick for over a month
    AND the patient has a fever
    THEN we suspect a bacterial infection

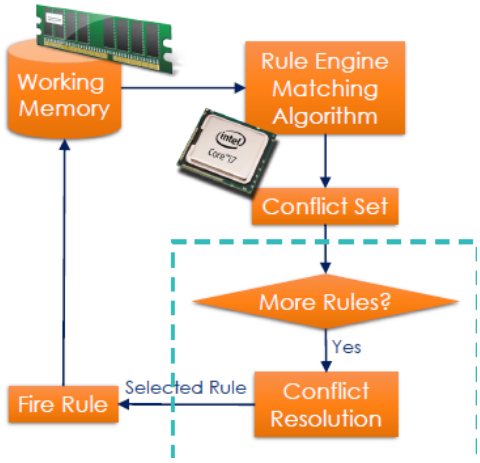
Facts:	Temperature = 104°F (40°C)	F1
	Patient has been sick for 2 months	F2
	Patient has a sore throat	F3
	patient has a fever	F4
	suspect a bacterial infection	F5
	illness is a strep throat	F6

    - Initial state of working memory WM: [F1, F2, F3]
    - Rule-2 [F1]
      - Add F4: “patient has a fever”
    - Rule-3 [F2, F4]
      - Add F5: “suspect a bacterial infection”
    - Rule-1 [F3, F5]
      - Add F6: “illness is a strep throat”
    - No more rules to fire -> halt
    - Conclusion-> illness is a strep throat

- Backward Chaining(goal-driven)
  - starts from a goalor hypothesis
  - It works through the rules trying to match the goal with the action clauses (THEN part) of a rule.
  - When a match is found, the condition clauses(IF part) of the matching rule become "sub goals".
  - The cycle is repeated until a verifiable set of condition clauses is found.
  - Appropriate for problem solving & investigation
    - e.g., Where are my keys? How do I get into a PhD program?
  - Backward Chaining Example



    - Initial state of working memory WM: [Facts + Hypothesis]
    - Rule-1 [Hypothesis F4]
      - Goal: we suspect a strep throat?{new goal to pursue Add F4}
      - Check: patient has a sore throat?{proved by F3}
      - Check: we suspect bacterial infection?{new goal to pursue Add F5}
    - Rule-3 [F5]
      - Check: patient has been sick for over a week?{proved by F2}
      - Check: patient has a fever?{new goal to pursue Add F6}
    - Rule-2 [F6]
      - Check: patient's temperature is >100°F (37.8°C) C)? {proved by F1}
    - No more rules to fire
    - {all proved} => halt
    - Conclusion -> yes, illness is a strep throat
- Conflict Resolution



- Conflict Set
  - More than one rule can fire based on the facts in WM. That is, the fact in WM can match more than one rule at a specific time. The matched activated rules represents a conflict set.
- Conflict Resolution
  - A method for choosing a rule to fire when more than one rule can be fired in a given cycle.
- Example

Rule 1: **IF** I have at least $20 AND Man-United is playing today
THEN I should go to the football game.
Rule 2: **IF** it is raining today AND I don't have school
THEN I should stay home
Rule 3: **IF** I have at least $20
THEN I should go to the cinema.
Rule 4: **IF** I should go to the cinema
THEN I should call my friends

Initial Facts: **Fact-1**: I have $20
Fact-2: Man-United is playing today
Conflict Set: **<R1: Fact-1, Fact-2>, <R3: Fact-1>**

- Strategy
  - The order in which the rule fires depends on facts in Working Memory, (in general) not the order of rules.
  - Rule firing priority:
    - Default is last in first out (LIFO) based on facts in WM
    - Attach priority to rules, and select the rule with the highest priority (Salience)
    - Order the facts by the length of time they have been in working memory, and select the most recent. (Recency)
    - Select rules which required the lowest/highest number of facts/rule-conditions (Specificity).
  - Strategy : Drools inference engine control parameters

Salience / Specificity / Recency / LIFO / FIFO

Individual rule's priority in Agenda

AgendaGroup

It allow you to place rules into groups, and to place those groups onto a stack (rule set's priority). The stack has push/pop behaviour.

ActivationGroup

It is a set of rules bound together by the same "activation-group" rule attribute. In this group only one rule can fire, and after that rule has fired all the other rules are cancelled from the agenda.

RuleFlowGroup

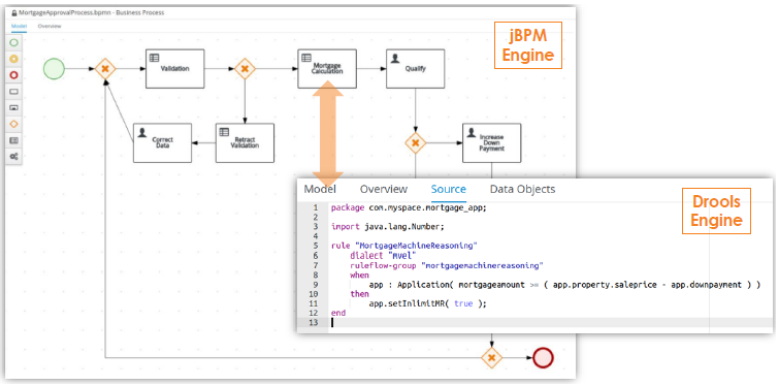
It is a group of rules associated by the "ruleflow-group" rule attribute. These rules can only fire when the group is activated. (jBPM Business Rule Task)

```
rule "Print balance for AccountPeriod"
    salience -50
  when
    ap : AccountPeriod()
    acc : Account()
  then
    System.out.println( acc.accountNo + " : " + acc.balance );
end
```

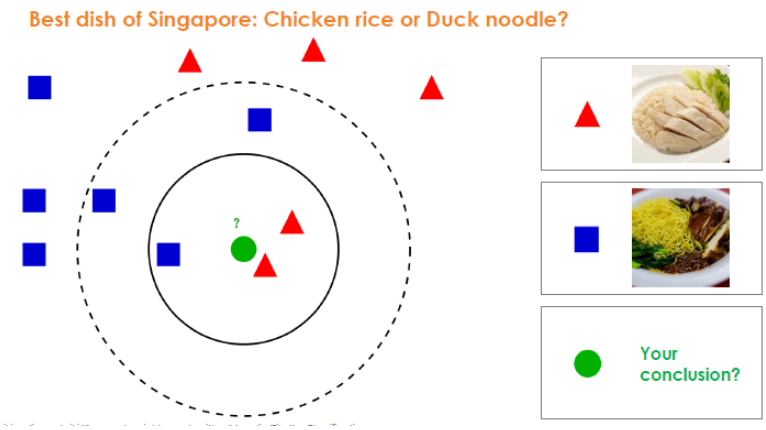
```
rule "increase balance for credits"
  agenda-group "calculation"
when
  ap : AccountPeriod()
  acc : Account( $accountNo : accountNo )
  CashFlow( type == CREDIT,
            accountNo == $accountNo,
            date >= ap.start && <= ap.end,
            $amount : amount )
then
  acc.balance  += $amount;
```

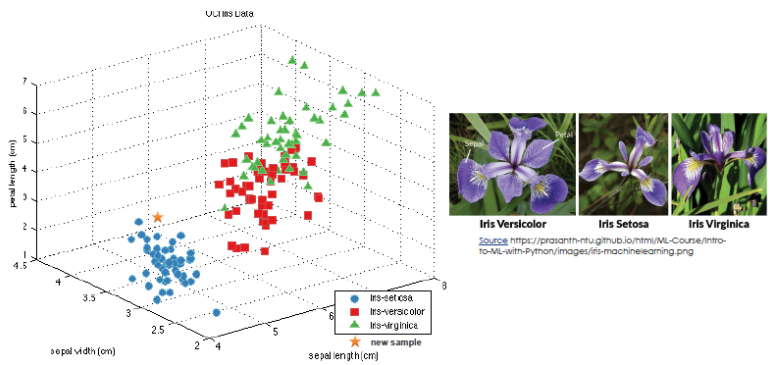
```
rule "Print balance for AccountPeriod"
  agenda-group "report"
when
  ap : AccountPeriod()
  acc : Account()
then
  System.out.println( acc.accountNo +
                      " : " + acc.balance );
end
```

- KIE BPMS/BRMS Suite Mortgage application systems



- Reasoning under Uncertainty
  - Analogical Reasoning (Similarity based method)
    - example

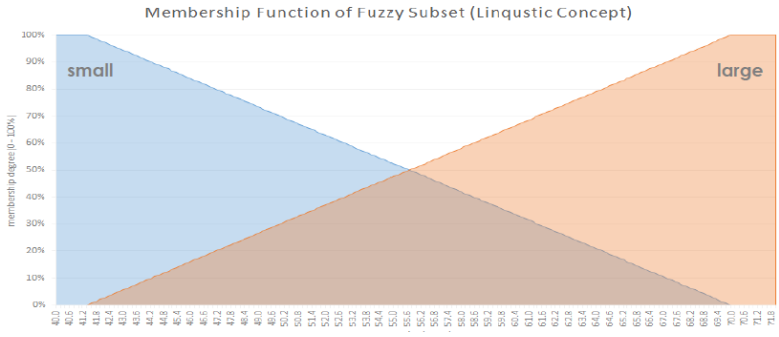




  - Fuzzy Logic
    - Fuzzy Logic are measures of inclination (degree of truth) towards a linguistic concept/word, which lacks a rigorous definition
    - 3 Steps of Fuzzy Reasoning

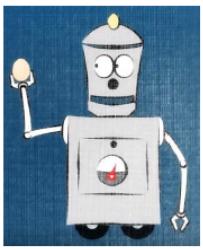
- **Egg-boiling Fuzzy Ruleset**
  - **IF** egg size is **small** **THEN** boil **less than 5 minutes**
  - **IF** egg size is **large** **THEN** boil **more than 5 minutes**

- Fuzzification



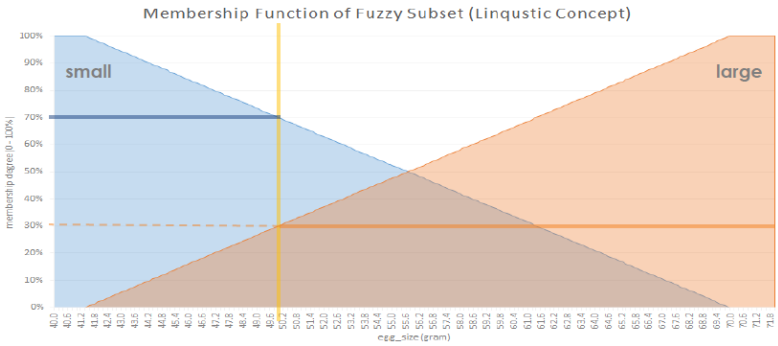
- Inference

  - **New fact: The egg is 50 grams**

    Link https://www.youtube.com/watch?v=J_GSX0nTmrA

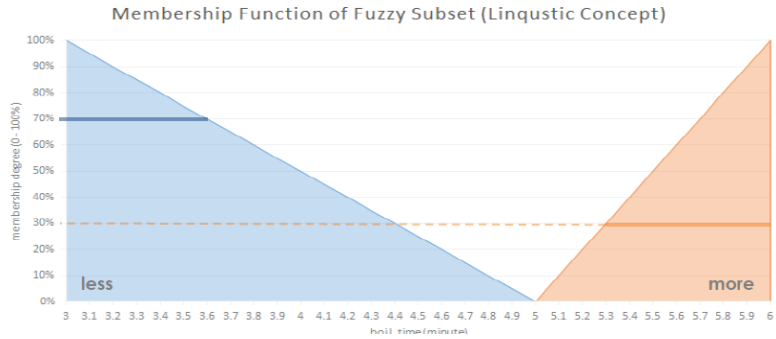
    Egg-boiling Fuzzy Ruleset
    - **IF** egg size is **small** **(%)** **THEN** boil **less than 5 minutes** **(%)**
    - **IF** egg size is **large** **(%)** **THEN** boil **more than 5 minutes** **(%)**

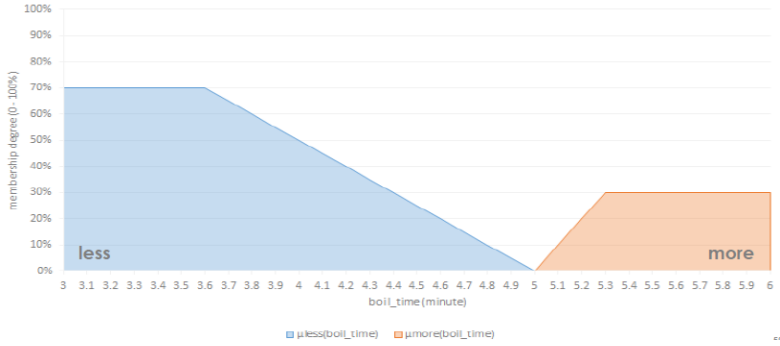
  - 

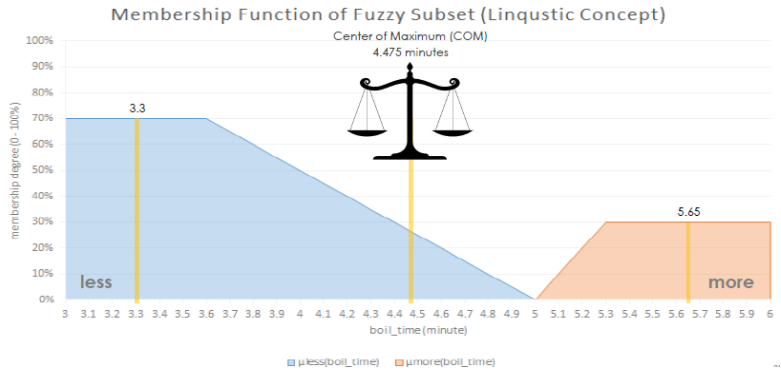
    Egg-boiling Fuzzy Ruleset

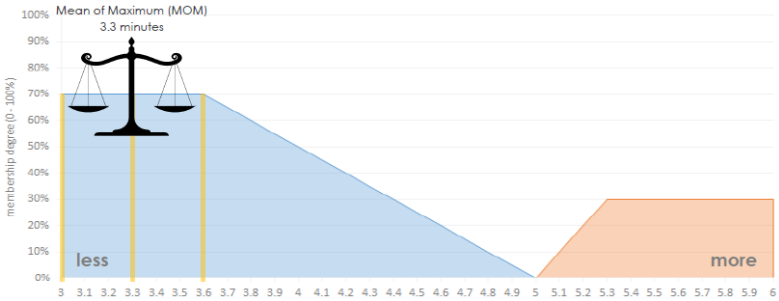
    - **IF** egg size is **small** **(70%)** **THEN** boil **less than 5 minutes** **(70%)**
    - **IF** egg size is **large** **(30%)** **THEN** boil **more than 5 minutes** **(30%)**

  -

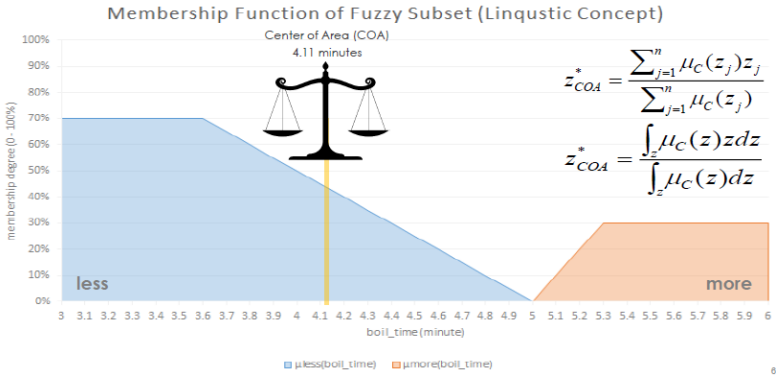




- 



- Defuzzificaiton



$$z^*_{COA} = \frac{\sum_{j=1}^{n} \mu_C(z_j) z_j}{\sum_{j=1}^{n} \mu_C(z_j)}$$

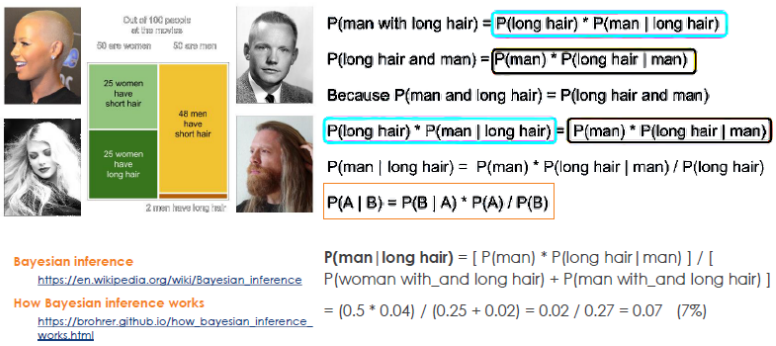
$$z^*_{COA} = \frac{\int_z \mu_C(z) z dz}{\int_z \mu_C(z) dz}$$

- Extension

- More than one conditions
  - IF egg size is small (70%) OR very hungry (90%) THEN boil less than 5 minutes (%)
  - IF egg size is large (45%) AND slightly hungry (75%) THEN boil more than 5 minutes (%)
- Composition operation
  - AND Min()
  - OR Max()

- Fuzzy Logic vs. Probability
  - Fuzziness and randomness deal with different types of uncertainty in our life
- Summary
  - The theory of fuzziness is to build models for entities which lack a rigorous definition.
  - The concept of " graded membership " belongs to a class which could be subjective in different business context
  - It is not compatible with a concept suitable for the lack of information , which is with probability

- Abductive Reasoning(Probability)
  - Probabilistic Reasoning (Bayesian inference)
    - Bayesian inference is a method of statistical inference in which Bayes' theorem is used to update the probability for a hypothesis as more evidence or information becomes available.
  - Bayesian Inference



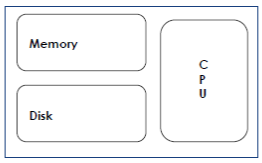
- Knowledge Discovery by Machine Learning (Big Data)
  - Big data is a field that treats ways to analyze, systematically extract information from, or otherwise deal with data sets that are too large or complex to be dealt with by traditional data processing application software.
  - It is also a term used to describe large volumes of data which is hard to process using traditional database management systems (DBMS)
  - 5V
    - 1.Volume : refers to the huge amount of data
    - 2.Velocity : refers to the speed at which data is

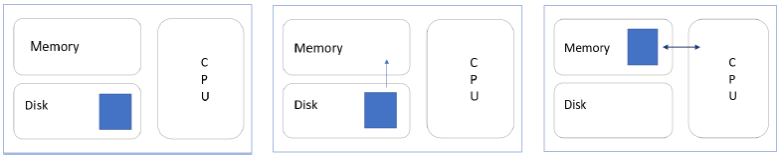
- 3.Variety : The variety of data available/generated, which includes(structured/semi structured/unstructured data);
- 4.Veracity : It refers to the uncertainties in data produced, i.e., real world data is messy and can be hard to analyze;
- 5. Value : Insights generated must be based on accurate data so that the knowledge discovered is of value to the company;

- traditional database management system(DBMS)

Data resides on disk and loaded into memory when needed, we do processing by spending CPU cycles on the server and then the data is presented to the user for analysis, which in turn spends/uses User CPU and memory.

Resources (**memory size**) are limited and its **slow (I/O: inputs/outputs)** to load data from disk. Big data or extremely large data sets that could be analyzed computationally to reveal insights like patterns, trends and associations is unlikely to fit in to memory **all at once**.



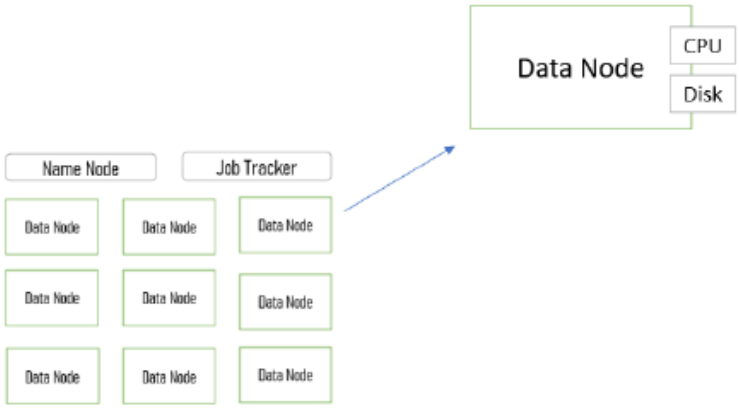


- How to Handle Big Data?
  - The solution is to scale out and use distributed computing by connecting many servers to perform jobs.
    - Distributed Computing
      - Recover from failures: With more resources things are bound to fail more frequently, because of that they must be able to recover from failure in a scalable manner;

      - Shared nothing architecture

- Shared Nothing Architecture: In order to scale out efficiently, we do not burden the master servers / name nodes, like coordination and communication with other data nodes: reduced networking communication overheads;
- Each data node is unaware of the other nodes in a cluster;
- The Coordination task is left to the master servers: Name Node and Job Tracker;
- Each Data node is unaware of the other nodes in a cluster;

- Hadoop distributed file system (HDFS)
  - Hadoop is an opensource implementation of the Hadoop distributed file System. It masks the complexities when with working with distributed systems;
  - Provides fault tolerant distributed file system on top of these nodes; (servers/storages) carrying forward all the characteristics as well as an API that makes it easier to work on Hadoop;
- Map Reduce (Divide Conquer method)



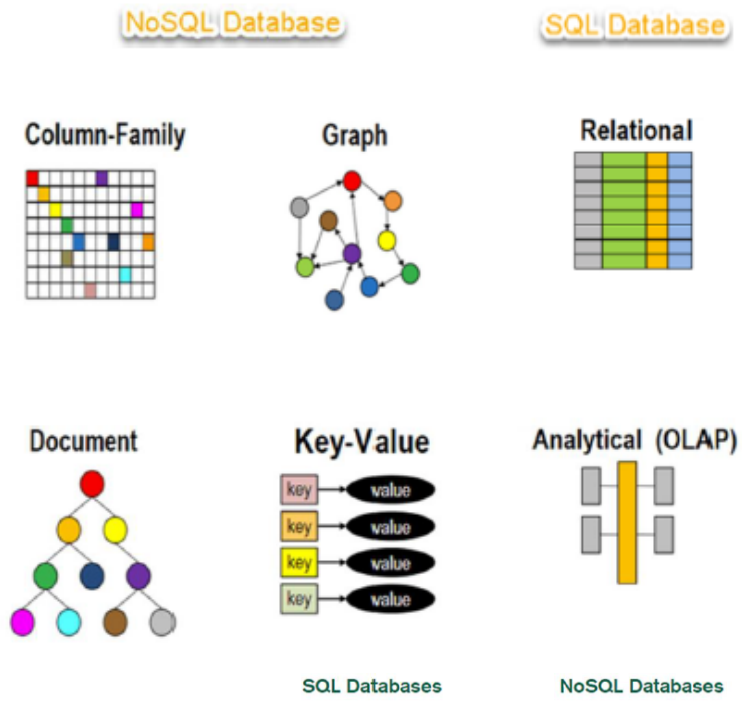
  - Takes compute against data stored on disk instead of data loaded in memory

- A distributed system is a system whose components are located on different networked computers , which communicate and coordinate their actions by passing messages to one another, via Remote Procedure Calls.

- NoSQL (Not Only SQL)
  - Types
    - Column family / Wide column stores store data in tables, rows, and dynamic columns;
    - Graph databases store data in nodes and edges;
    - Document databases store data in documents similar to JSON Java S cript Object Notation) objects. Each document contains pairs of (embedded) fields and values;

- Key value databases are a simpler type of database where each item contains keys and values;

- SQL VS NOSQL



	SQL Databases	NoSQL Databases
Data Storage Model	Tables with fixed rows and columns	Document: JSON documents, Key-value: key-value pairs, Wide-column: tables with rows and dynamic columns, Graph: nodes and edges
Development History	Developed in the 1970s with a focus on reducing data duplication	Developed in the late 2000s with a focus on scaling and allowing for rapid application change driven by agile and DevOps practices.
Examples	Oracle, MySQL, Microsoft SQL Server, and PostgreSQL	Document: MongoDB and CouchDB, Key-value: Redis and DynamoDB, Wide-column: Cassandra and HBase, Graph: Neo4j and Amazon Neptune

- Data Structure in SQL DB

  Example: Employee DB

  Each row in each table is a single value(employee), and each column has fixed attributes:

Emp_ID	Emp_firstname	Emp_lastname	Emp_phone
1	Geet	Jethwani	81815656

- Data Structure in NoSQL DB

  Example: document database – JSON;

  Each of the SQL column attributes would be fields;

  The details of an employee's record would be the data values associated with each field;

  For Example: Emp_firstname: "Geet", Emp_lastname: "Jethwani"

Experience data represented in semi-structured format;

We have looked at the structured (tabular) formats;

However, we must understand JSON representation as in real world big data application, one may come across this very frequently;

When setting up a data store, your first task is to answer the question: "What data would I like to store and how do the fields relate to each other"?

name	quantity	size	status	tags	rating
journal	25	14x21,cm	A	brown, lined	9
notebook	50	8.5x11,in	A	college-ruled,perforated	8

- JSON is formatted as name-value pairs;
- In JSON documents, fieldnames and values are separated by a colon, fieldname and value pairs are separated by commas, and sets of fields are encapsulated in "curly brackets" ({});
- Example:

```
{          Key          Value
 "name"       : "notebook",
 "quantity"   : 50,      Key      Value
 "size"       : { "length": 11, "width": 8.5, "unit": "in" },
 "status"     : "A",
 "tags"       : [ "college-ruled", "perforated" ],
 "rating"     : 8
}
```