Q1

1)

Expert systems, despite benefits, face limitations:

- Acquiring and keeping expert knowledge is hard and expensive.
- Reasoning can be limited to pre-defined rules, lacking flexibility.
- Building and maintaining them requires specialized skills and resources.
- Users may struggle to understand how they reach conclusions.
- Adapting to new situations or exceptions can be challenging.

2)

Limitations:

1. Expressiveness: Predicate logic struggles with vagueness, uncertainty, and probability, common in real-world domains.
2. Intractability: Reasoning with large knowledge bases can become computationally expensive.
3. Modularity and Knowledge Representation: Representing complex, hierarchical knowledge structures can be challenging.
4. Closed-World Assumption: Assumes all relevant information is explicitly stated, hindering incomplete knowledge handling.
5. Lack of Meta-Reasoning: Cannot directly reason about its own knowledge or reasoning processes.
6. Limited Expressiveness for Temporal and Spatial Reasoning: Not well-suited for representing and reasoning about time and space.

Solutions:

- Fuzzy Logic: Introduces degrees of truth to handle vagueness and uncertainty, overcoming limitation 1.
- Theorem Proving and Model Checking: Techniques to automate proof finding and efficiently explore states, addressing limitation 2.
- Ontologies and Semantic Networks: Provide shared vocabularies and flexible representation of complex relationships, tackling limitation 3.
- Default Logic and Open-World Reasoning: Allow for assumptions in absent information and explicitly reason about the unknown, overcoming limitation 4.

- Meta-Logic and Meta-Programming: Enable reasoning about knowledge representation and adaptive/reflective reasoning systems, addressing limitation 5.
- Temporal Logic and Spatial Logic: Extend predicate logic with operators for representing and reasoning about time and space, overcoming limitation 6.

3)

1. Shopping: Machine learning recommends products you'll love based on your past purchases.
2. Seeing: From phones to cars, machine learning helps interpret and classify images.
3. Protecting: Banks use machine learning to catch fraudsters before they strike.
4. Cleaning: Keep your inbox tidy with machine learning-powered spam filters.
5. Talking: Machine learning helps translation tools break down language barriers.

Q2

1)

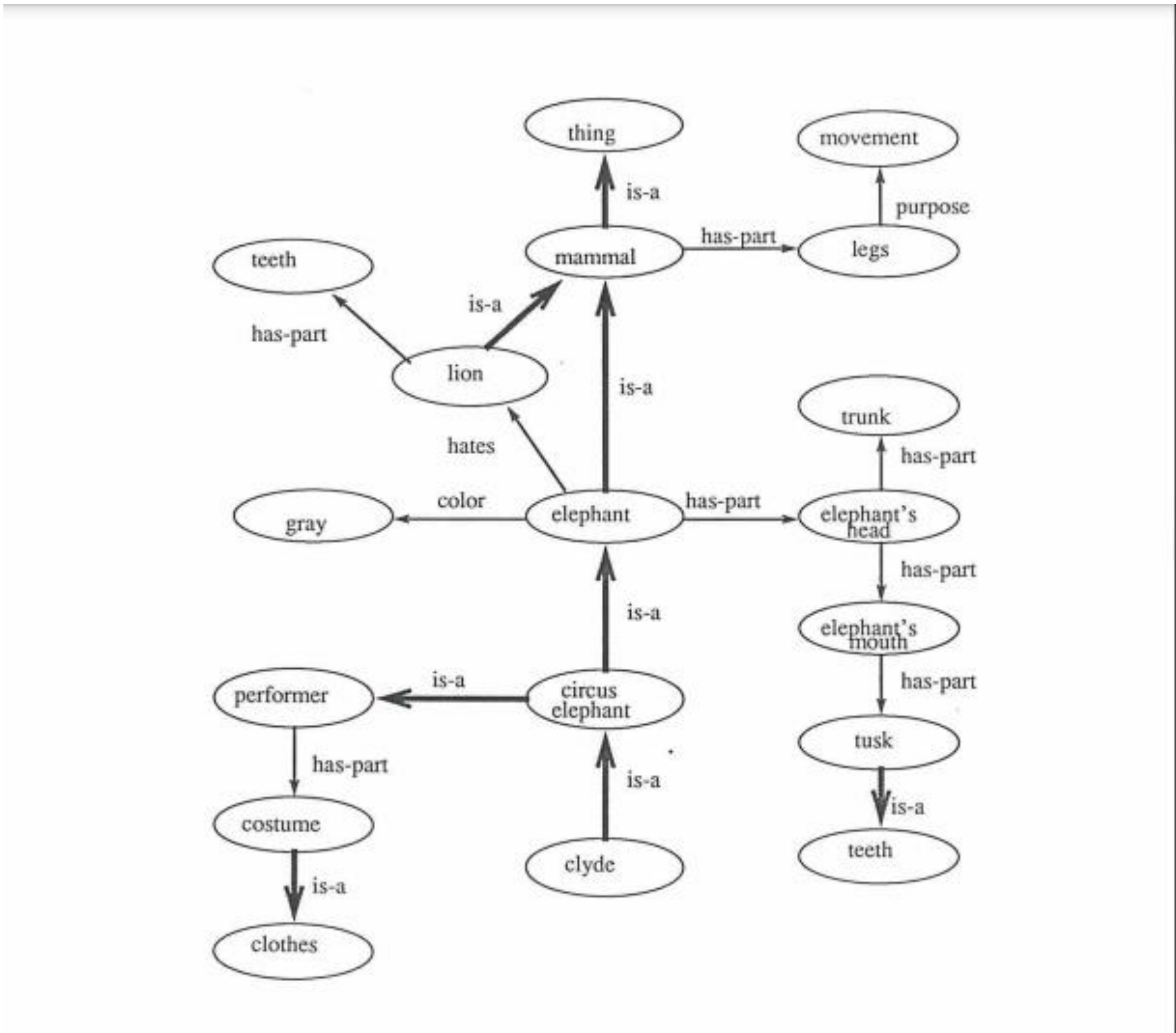


Figure 2: Semantic Network for Clyde

2)

Here are the sentences translated into predicate logic:

1. Every student in this class has visited either Lahore or Murree:

```
∀x (Student(x) ∧ InClass(x) → (Visited(x, Lahore) ∨ Visited(x, Murree)))
```

Breakdown:

- `∀x`: For all x.

- `Student(x)`: x is a student.
- `InClass(x)`: x is in this class.
- `→`: Implies.
- `(Visited(x, Lahore) ∨ Visited(x, Murree))`: x has visited Lahore or x has visited Murree.

2. Every student in this class has studied calculus:

`∀x (Student(x) ∧ InClass(x) → Studied(x, Calculus))`

Breakdown:

- Similar to the first sentence, but `Visited` is replaced with `Studied` and `Lahore/Murree` is replaced with `Calculus`.

3. Some fierce creatures do not drink coffee:

`∃x (FierceCreature(x) ∧ ¬Drinks(x, Coffee))`

Breakdown:

- `∃x`: There exists an x.
- `FierceCreature(x)`: x is a fierce creature.
- `¬`: Not.
- `Drinks(x, Coffee)`: x drinks coffee.

4. Every Pakistani loves Hockey but cricket is a familiar game:

`∀x (Pakistani(x) → Loves(x, Hockey)) ∧ FamiliarGame(Cricket)`

Breakdown:

- `∀x`: For all x.
- `Pakistani(x)`: x is a Pakistani.
- `→`: Implies.
- `Loves(x, Hockey)`: x loves hockey.
- `∧`: And.

- `FamiliarGame(Cricket)`: Cricket is a familiar game.

This assumes that a sentence like "Cricket is a familiar game" is represented as a fact about the world, not within the same scope as the quantifier.

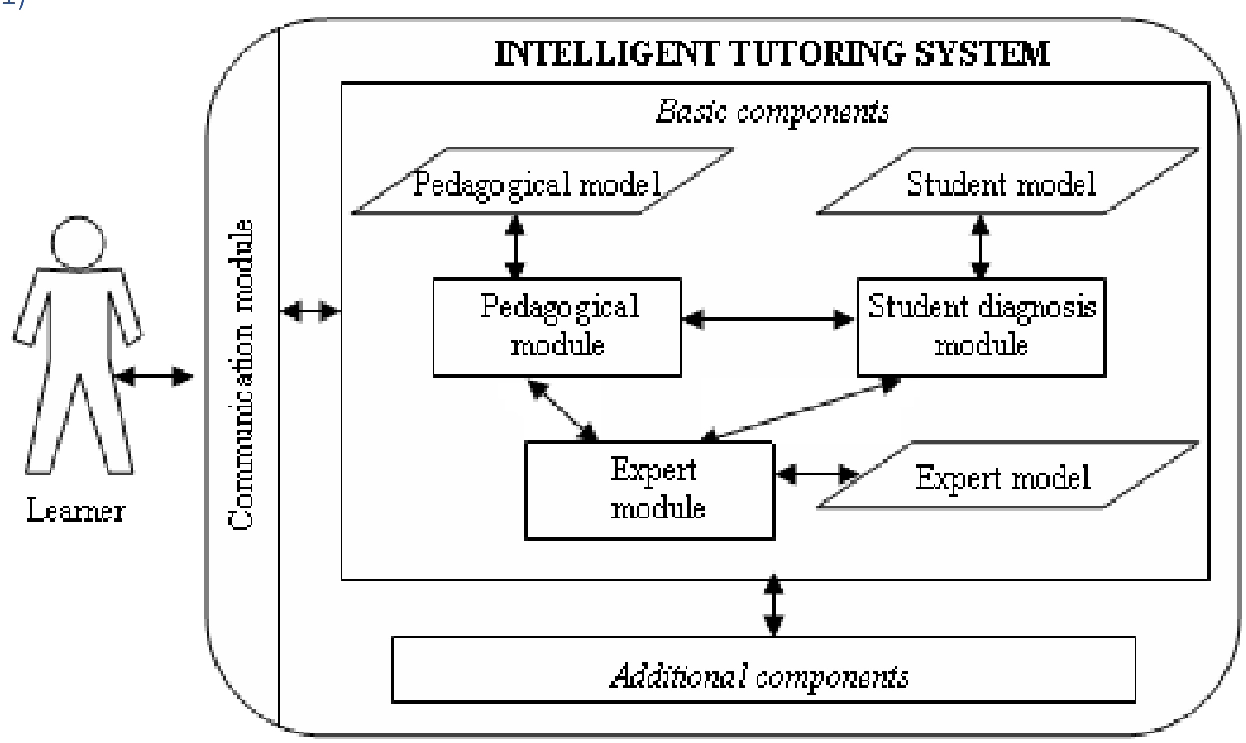
3)

Here are the steps of K-means clustering in short form:

1. Choose k (number of clusters).
2. Initialize k centroids.
3. Assign points to closest centroids.
4. Recalculate centroids as means of assigned points.
5. Repeat steps 3-4 until convergence.

Q3

1)



Knowledge:

- Domain: What's being taught (math, science, etc.).
- Pedagogy: How to teach effectively (explanations, examples).
- Student Modeling: Track your learning progress.
- Language Processing: Understand and respond naturally.

2) https://www.analyticsvidhya.com/blog/2018/08/k-nearest-neighbor-introduction-regression-python/
3)

Supervised vs. Unsupervised Learning: Key Differences and Techniques

Supervised learning: involves training a model with labeled data, where each data point has a corresponding label or output. The model learns to map the input features to the desired output. Think of it like a student learning from labeled examples with a teacher's guidance.

Unsupervised learning: deals with unlabeled data, where the model seeks to discover hidden patterns or structures within the data without any predefined labels or outputs. It's like exploring a new world to find your own meaning and connections.

Supervised techniques:

- Linear Regression: predicts numbers based on patterns.
- Decision Trees: classifies data by asking questions about features.
- Support Vector Machines: finds best borders between different data groups.

Unsupervised techniques:

- K-means Clustering: groups similar data points together.
- Principal Component Analysis: simplifies data without losing much information.
- Anomaly Detection: finds unusual data points that don't fit the pattern.

Q4

1)

Example of Bayes Rule

- Givens
  - A doctor knows that meningitis causes stiff neck 50% of the time
  - Prior probability of any patient having meningitis is 1/50,000
  - Prior probability of any patient having stiff neck is 1/20

- If a patient has stiff neck, what's the probability he/she has meningitis?

$$P(M \mid S) = \frac{P(S \mid M)P(M)}{P(S)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002$$

datasciencedojo
unleash the data scientist in you

2)

Single perceptron Neural network

In the given problem, there is 3 input sensors . Shape, texture and weight.

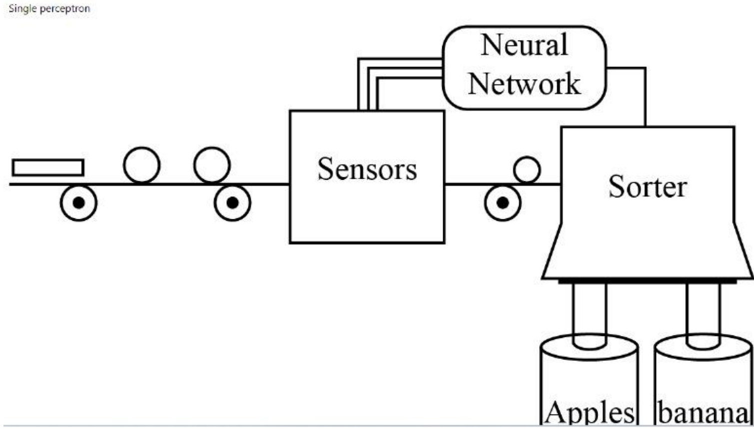
Shape sensor: -1 if the fruit is round and 1 if it is more elliptical

Texture sensor: -1 if the surface is smooth, 1 if is rough

Weight sensor: -1 if the fruit >500g, 1 if is<500g

Network architecture

Single perceptron



Here, 3 sensor output are input to the neural network as shown in the figure. The network decides which fruit is directed to correct storage bin. Here taking 2 fruits, apple and banana. The sensor recognize the fruit by using 3 vectors. They are shape, texture and weight.

$$P = \begin{bmatrix} Shape \\ Texture \\ Weight \end{bmatrix}$$

Given dataset for apple and banana:

Therefore Apple prototype represented as:

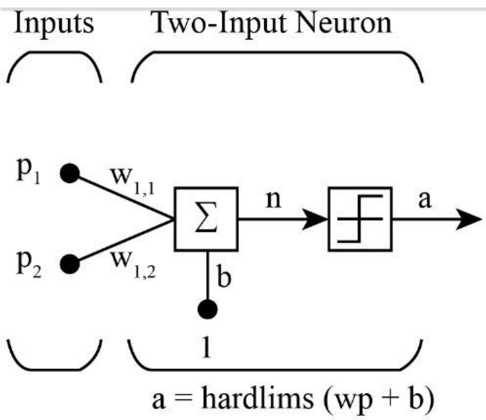
$$P_a = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}$$

Therefore Banana prototype represented as:

$$P_b = \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix}$$

Two input single perceptron network

Two input (Apple and banana) perceptron is represented in the figure



Vector input R=3

So, Perceptron equation is:

$$a = hardlims\left([W1,1\ w1,2\ w1,3]\begin{bmatrix} P1 \\ P2 \\ P3 \end{bmatrix} + b\right)$$

Choose bias b and weight matrix elements, so that the perceptron will able to distinguish the 2

Choose bias b and weight matrix elements, so that the perceptron will able to distinguish the 2 fruits.

Given initial weight matrix as $\begin{bmatrix} 0.5 & -1.0 & -0.5 \end{bmatrix}$ and bias b=0.5

Then,

Decision boundary calculated using $W_p + b = 0$

In case of apple and banana it is 0 , since boundary is linear

Apple: $a = hardlims\left(\begin{bmatrix} 0.5 & -1.0 & -0.5 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} + 0\right)$

$= 1\left(apple\right)$

Banana: $a = hardlims\left(\begin{bmatrix} 0.5 & -1.0 & -0.5 \end{bmatrix}\begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} + 0\right)$

$= -1\left(banana\right)$

Hence, any input vector that is closer to the banana prototype vector than to the apple prototype vector (in Euclidean distance) will be classified as an banana (and vice versa).

3)

Choosing the optimal value of k in k-nearest neighbors (kNN) is a crucial step in optimizing its performance. Both high and low values of k have their own advantages and disadvantages, which we'll explore here:

Pros of Low k:

- **Reduced model complexity**: With fewer neighbors, the model becomes simpler and easier to interpret. This can be beneficial for scenarios where interpretability is important.
- **Reduced variance:** Lower k helps mitigate the influence of outliers and noisy data points, potentially leading to a more stable model.
- **Sharper decision boundaries:** A low k can create sharper decision boundaries between classes, especially when the data is linearly separable.

Cons of Low k:

- **Higher sensitivity to noise:** Low k models are more susceptible to noise in the data, as a single noisy neighbor can have a significant impact on the prediction.
- **Overfitting:** If k is too low, the model might overfit the training data, leading to poor performance on unseen data.
- **Less robust to data sparsity:** In sparse datasets, low k might not find enough relevant neighbors, leading to unreliable predictions.

Pros of High k:

- **Reduced variance:** Higher k values can average out the influence of noisy data points, leading to a more robust model.
- **Smoother decision boundaries:** With more neighbors, the decision boundaries become smoother, potentially improving accuracy for non-linearly separable data.
- **Less sensitive to outliers:** A high k helps mitigate the influence of outliers, leading to more stable predictions.

Cons of High k:

- **Increased model complexity:** With more neighbors, the model becomes more complex and computationally expensive to train and predict.
- **Increased risk of overfitting:** A high k can lead to overfitting, especially with noisy data or small datasets.

- **Loss of interpretability:** As k increases, the model becomes less interpretable, making it difficult to understand the reasoning behind its predictions.

Convert the following sentences into FOL: I. Everyone who loves all animals is loved by someone. II. Anyone wo kills an animal is loved by no one. III. Jack loves all animals. IV. Either Jack or Curiosity killed the cat, who is names Tuna. V. It is a crime for an American to ell weapons to hostile nations.

Here are the sentences converted into First-Order Logic (FOL):

I. ∀x (Loves(x, AllAnimals) → ∃y (Loves(y, x)))

- `∀x`: For all x (universal quantifier)
- `Loves(x, AllAnimals)`: x loves all animals
- `→`: implies
- `∃y`: There exists a y (existential quantifier)
- `Loves(y, x)`: y loves x

II. ∀x (Kills(x, Animal) → ∀y (¬Loves(y, x)))

- `∀x`: For all x (universal quantifier)
- `Kills(x, Animal)`: x kills an animal
- `→`: implies
- `∀y`: For all y (universal quantifier)
- `¬`: not
- `Loves(y, x)`: y loves x

III. Loves(Jack, AllAnimals)

- `Loves(Jack, AllAnimals)`: Jack loves all animals

IV. (Kills(Jack, Tuna) ∨ Kills(Curiosity, Tuna)) ∧ Cat(Tuna)

- `∨`: Or
- `Kills(Jack, Tuna)`: Jack kills Tuna
- `Kills(Curiosity, Tuna)`: Curiosity kills Tuna
- `∧`: And

- `Cat(Tuna)`: Tuna is a cat

V. Crime(American(x) ∧ Sells(x, Weapons, HostileNation))

- `Crime`: It is a crime
- `American(x)`: x is an American
- ∧: And
- `Sells(x, Weapons, HostileNation)`: x sells weapons to a hostile nation

2)

<u>Ex(2):</u> We now present an example of a resolution refutation for the predicate calculus. We wish to prove that "Fido will die" from the statements that "Fido is a dog. All dogs are animals. All animals will die."

Fido is a dog: dog (fido).

All dogs are animals: $\forall(X)$ (dog (X) $\rightarrow$ animal (X)).

All animals will die: $\forall(Y)$ (animal (Y) $\rightarrow$ die (Y)).

Converts these predicates to clause form:

PREDICATE FORM	CLAUSE FORM
1. $\forall(X)$ (dog) (X) $\rightarrow$ animal (X))	$\neg$ dog (X) $\vee$ animal (X)
2. dog (fido)	dog (fido)
3. $\forall(Y)$ (animal (Y) $\rightarrow$ die (Y))	$\neg$ animal (Y) $\vee$ die (Y)

Negate the conclusion that Fido will die:

4. $\neg$ die (fido)	$\neg$ die (fido)

Resolve clauses having opposite literals, producing new clauses by resolution as in Figure (3-2).

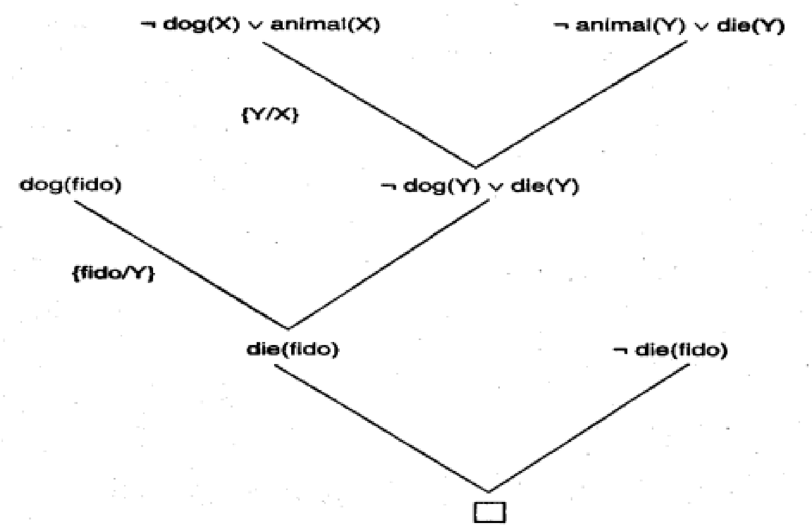


Figure (3-2): Resolution proof for the "dead dog" problem.

9

3)

Answer the following with one [2 x 5 = 10] example of each. I) Write down advantages of production rules. 2) What is decision tree and how it works? 3) Enlist the steps of Nearest Neighbor Algorithm. 4) Write down the syntax of Propositional Logic. 5) What is knowledge engineering? Write down steps involves in knowledge engineering.

Ans: **Production rules:** Modular, transparent, flexible. Example: Medical diagnosis adding new disease rule.

Decision tree: Tree-like structure, asking questions to classify data. Example: Loan approval based on income, credit score, etc.

Nearest Neighbor: Measure distances, find closest neighbors, assign class based on their majority. Example: Movie recommendations similar to what you liked.

Propositional logic: Basic propositions, connectives like AND, OR, NOT. Example: Rain AND red car IMPLIES take bus.

Knowledge engineering: Building knowledge-based systems. Steps: Acquire, represent, infer, validate. Example: Medical diagnosis system with expert rules.

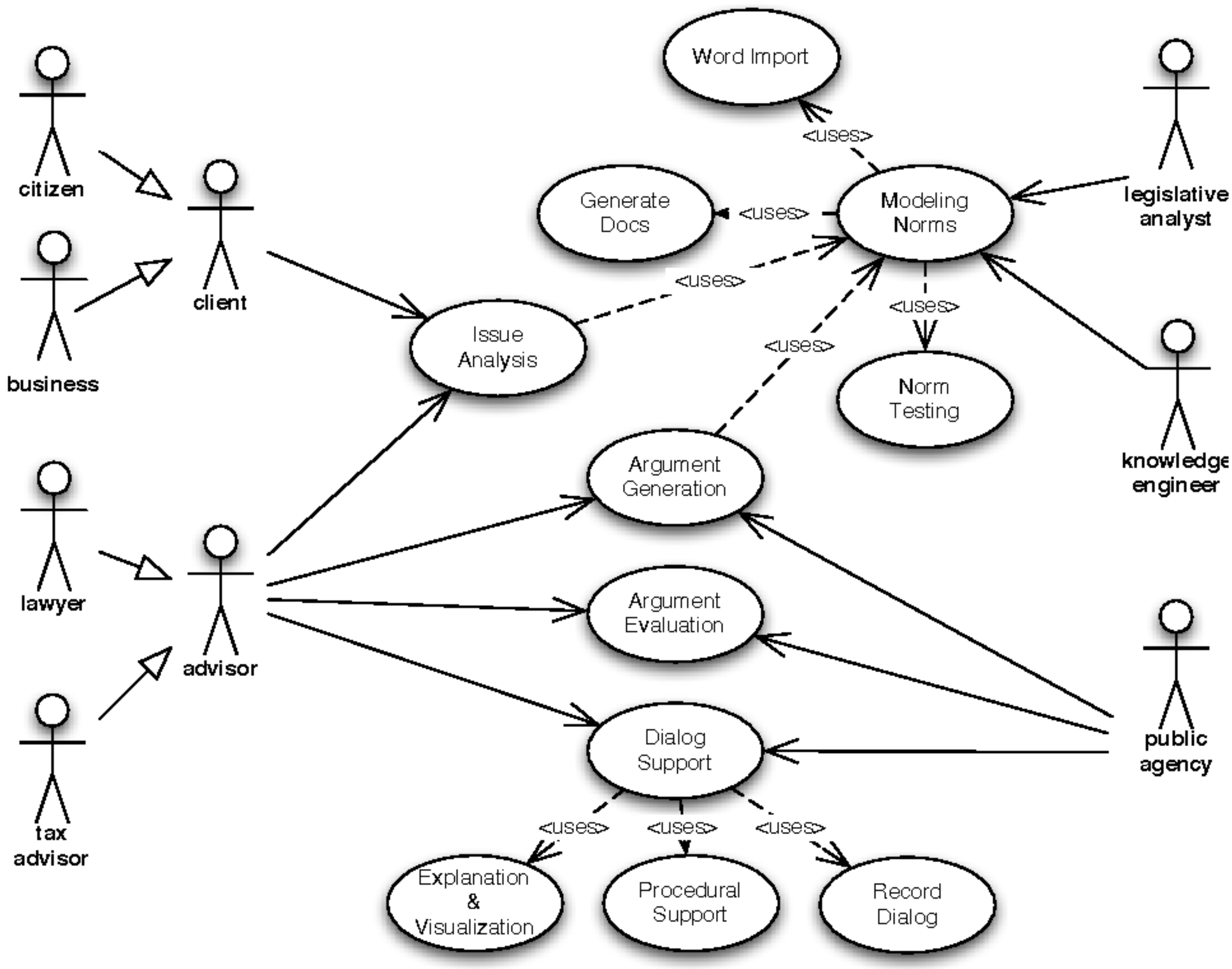
5)

B) The T-shirt size ofa customer is determined from the height and weight as shown in Table l.

A New customer has height 161 cm and weight 61kg. Predict the T-Shirt size.

Ans: Solution

6) Elaborate the structure of a legal advisor system only with the help ofa figure.



citizen
business
client
lawyer
advisor
tax advisor
Word Import
Generate Docs
Modeling Norms
Norm Testing
Issue Analysis
Argument Generation
Argument Evaluation
Dialog Support
Explanation & Visualization
Procedural Support
Record Dialog
legislative analyst
knowledge engineer
public agency
<uses>