CSE643: Artificial Intelligence
Assignment-1

Divyajeet Singh (2021529)

October 13, 2023

Theory

Problem-1

(a) The given text is composed of four sentences. Let us call them S_1, S_2, S_3, S_4 respectively. Let us define the following propositions

1. A: The universe simply exists as it is
2. B: The universe ends in a heat death
3. C: There was a big bang
4. D: The universe is expanding
5. E: The universe is accelerated

Then, the given sentences are as follows

$$S_1 = A \lor B \equiv \neg A \implies B \tag{1}$$
$$S_2 = \neg C \implies A \equiv C \lor A \tag{2}$$
$$S_3 = D \iff C \equiv (D \implies C) \land (C \implies D) \tag{3}$$
$$S_4 = (D \land E) \implies B \tag{4}$$

(b) The contrapositives of the sentences are

$$S_1 = \neg B \implies A$$
$$S_2 = \neg A \implies C$$
$$S_3 = (\neg C \implies \neg D) \land (\neg D \implies \neg C) \equiv \neg C \iff \neg D$$
$$S_4 = \neg B \implies \neg(D \land E) \equiv \neg B \implies (\neg D \lor \neg E)$$

(c) If B is False, then at least one of D or E must be False by (4). If D is False, then C must be False by (3). Finally, by (2) and (1), A must be True. Hence, **we can infer that** either the universe is not expanding or the universe is not accelerated if the universe does not end in a heat death. Also, if the universe is not expanding, then there was no big bang and the universe simply exists as it is.
If A is False, then B and C must be True by (1) and (2) respectively. Then, by (3), D must be True. But since B is True, the implication in (4) is vacuously True. So we cannot

determine the truth value of E. This means that **we cannot infer** whether the universe is accelerated or not if the universe does not simply exist as it is.

(d) The AND-OR graph of the statements is given in Figure 1. This graph is in accordance

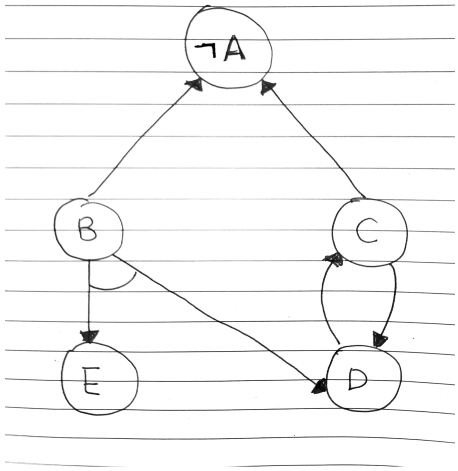

Figure 1: And-OR Graph of the Statements

with the notation given on the Wikipedia page - And-OR Tree.

Problem-2

The required semantic network is given in Figure 2. The centeral (labelled) node represents myself. The other three nodes with names represent three individuals with whom I share a relationship. The edges represent and are labelled as *MemberOf*, *SubsetOf*, etc. to denote the relationship between the two nodes.

Inheritance:
An example of Inheritance can be found in the part of the semantic diagram showing locations. The node *India* has an attribute indicating it is a Warm Country. Similarly, the node *Abroad* is labelled as Cool. So, by inheritance, the members of these classes inherit these properties - i.e. Deoband is a warm place and so is Delhi, while Paris is a cool place.

Multiple Inheritance:
An example of Multiple Inheritance can be found in the part of the semantic diagram showing the persons. The node *Ananya* belongs to the class *Female* as well as *Student*, so it can display properties of both. Similarly, the node *Ram* is both *Male* and an *Office Worker*.

Problem-3

Proof by resolution technique works by resolving complementary literals iteratively from a set of clauses until either the empty clause is obtained or no further resolution is possible. This is formally written as the inference rule

$$\frac{l_1 \vee \cdots \vee l_k, \quad m_1 \vee \cdots \vee m_n}{l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \vee \cdots \vee l_k \vee m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n}$$

where l_i and m_j form a pair of complementary literals. We now show that the method of inference by resolution is sound and complete.

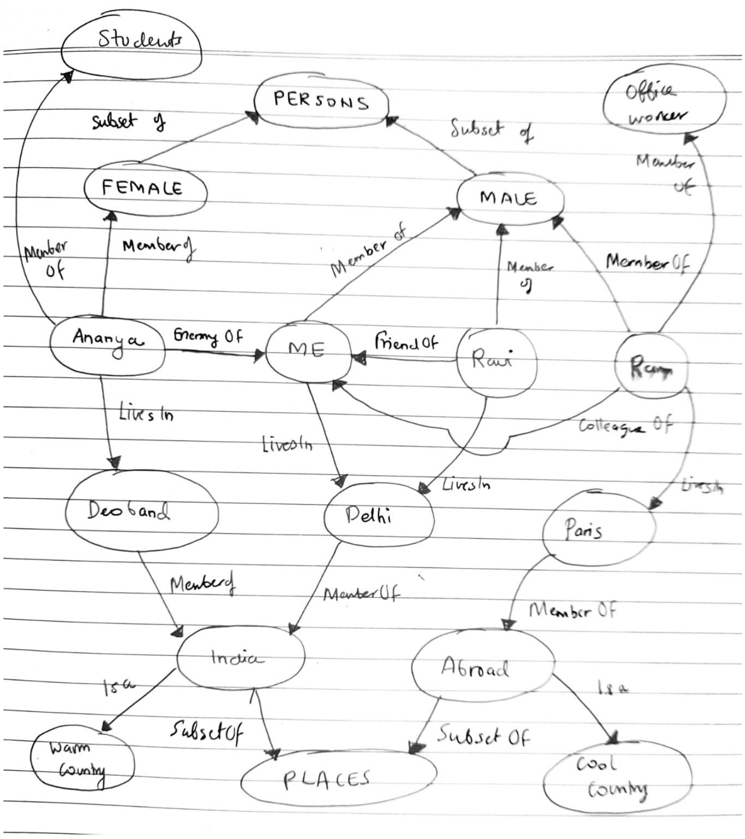

Figure 2: Semantic Network of my relationships with 3 individuals

Proof for Soundness

The *soundness* of proof by resolution can be shown easily. Let l_i and m_j be two complementary literals as above. Given the two clauses

$$l_1 \vee \cdots \vee l_k \tag{5}$$
$$m_1 \vee \cdots \vee m_n \tag{6}$$

Since l_i and m_j are complementary literals, l_i is TRUE when m_j is FALSE and vice versa. Let us consider the two cases.

1. l_i is FALSE. Then, $l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \vee \cdots \vee l_k$ must be TRUE because (5) is given. This makes the inferred clause TRUE.

2. l_i is TRUE, i.e. m_j is FALSE. Then, $m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n$ must be TRUE because (6) is given. This suggests that the inferred clause is TRUE.

Proof for Completeness

With the assumption that the knowledge base $\mathbb{K}$ is in CNF form, the *completeness* of proof by resolution can also be shown easily. Let Σ be the set of all clauses of the (say) k symbols from $\mathbb{K}$. Then, let $\Omega(\Sigma)$ be the closure of Σ under the resolution rule[1]. The *completeness* is proved using Ground Resolution Theorem, which states that if $\Omega(\Sigma)$ does not contain the empty clause, then Σ is satisfiable.
To prove this, we construct a model $\mathbf{P}_{1:k} = P_1, \ldots, P_k$ for Σ. For each symbol P_i, assign FALSE if some clause in $\Omega(\Sigma)$ uses $\neg P_i$ and all its other literals are FALSE under the model $\mathbf{P}_{1:i-1}$.

[1] $\Omega(\Sigma)$ is finite, since only a finite number of clauses can be generated from a finite Σ.

Otherwise, assign TRUE.
For the sake of contradiction, let us assume that at stage j, assigning symbol P_j causes some clause θ to become FALSE. This means that all other literals in θ must be assigned FALSE under the model $\mathbf{P}_{1:j-1}$. Therefore, θ must be one of the two forms

$$\text{FALSE} \lor \text{FALSE} \lor \cdots \lor \text{FALSE} \lor P_j \tag{7}$$

$$\text{FALSE} \lor \text{FALSE} \lor \cdots \lor \text{FALSE} \lor \neg P_j \tag{8}$$

Clearly, both clauses must be in $\Omega(\Sigma)$, otherwise the procedure would assign an appropriate value to P_j. Since $\Omega(\Sigma)$ is closed under resolution, there must be some clause θ' in $\Omega(\Sigma)$ that is the resolvent of (7) and (8). This resolvent will already have its literals $P_1, \ldots, P_{j-1}$ assigned to FALSE. This contradicts our assumption that the first falsified clause appears at stage j. So, the construction produces a model for $\Omega(\Sigma)$. Since $\Sigma \subseteq \Omega(\Sigma)$, $\mathbf{P}_{1:k}$ is also a model for Σ; and since it is a satisfying assignment, Σ is satisfiable.

Computational

The solution to the computational part of the assignment is given in the code in `main.py` in the submission. The code builds a recommender system for vacation destinations in great detail. The following attributes have been considered for decision making and wirting clauses in the code:

1. Weather of the destination
2. The budget of the user
3. The modes of connectivity to the destination
4. The tourist activities available at the destination
5. Rating of the destination

The user can also add and view feedback for places - thereby explicitly modifying the knowledge base. The knowledge base is stored in the file `database.json` in the submission. This JSON file stores information about various destinations.