Week 8

FOL-prove the given query by forward chaining

Code:

```
# Define the knowledge base (KB) as a set of facts
KB = set()

# Premises based on the provided FOL problem
KB.add('American(Robert)')
KB.add('Enemy(America, A)')
KB.add('Missile(T1)')
KB.add('Owns(A, T1)')

# Define inference rules
def modus_ponens(fact1, fact2, conclusion):
    """ Apply modus ponens inference rule: if fact1 and fact2 are true,
then conclude conclusion """
    if fact1 in KB and fact2 in KB:
        KB.add(conclusion)
        print(f"Inferred: {conclusion}")

def forward_chaining():
    """ Perform forward chaining to infer new facts until no more
inferences can be made """
    # 1. Apply: Missile(x) → Weapon(x)
    if 'Missile(T1)' in KB:
        KB.add('Weapon(T1)')
        print(f"Inferred: Weapon(T1)")

    # 2. Apply: Sells(Robert, T1, A) from Owns(A, T1) and Weapon(T1)
    if 'Owns(A, T1)' in KB and 'Weapon(T1)' in KB:
        KB.add('Sells(Robert, T1, A)')
        print(f"Inferred: Sells(Robert, T1, A)")

    # 3. Apply: Hostile(A) from Enemy(A, America)
    if 'Enemy(America, A)' in KB:
        KB.add('Hostile(A)')
        print(f"Inferred: Hostile(A)")
```

```
    # 4. Now, check if the goal is reached (i.e., if 'Criminal(Robert)' can
be inferred)
    if 'American(Robert)' in KB and 'Weapon(T1)' in KB and 'Sells(Robert,
T1, A)' in KB and 'Hostile(A)' in KB:
        KB.add('Criminal(Robert)')
        print("Inferred: Criminal(Robert)")

    # Check if we've reached our goal
    if 'Criminal(Robert)' in KB:
        print("Robert is a criminal!")
    else:
        print("No more inferences can be made.")

# Run forward chaining to attempt to derive the conclusion
forward_chaining()
```

Output:

```
Inferred: Weapon(T1)
Inferred: Sells(Robert, T1, A)
Inferred: Hostile(A)
Inferred: Criminal(Robert)
Robert is a criminal!
```

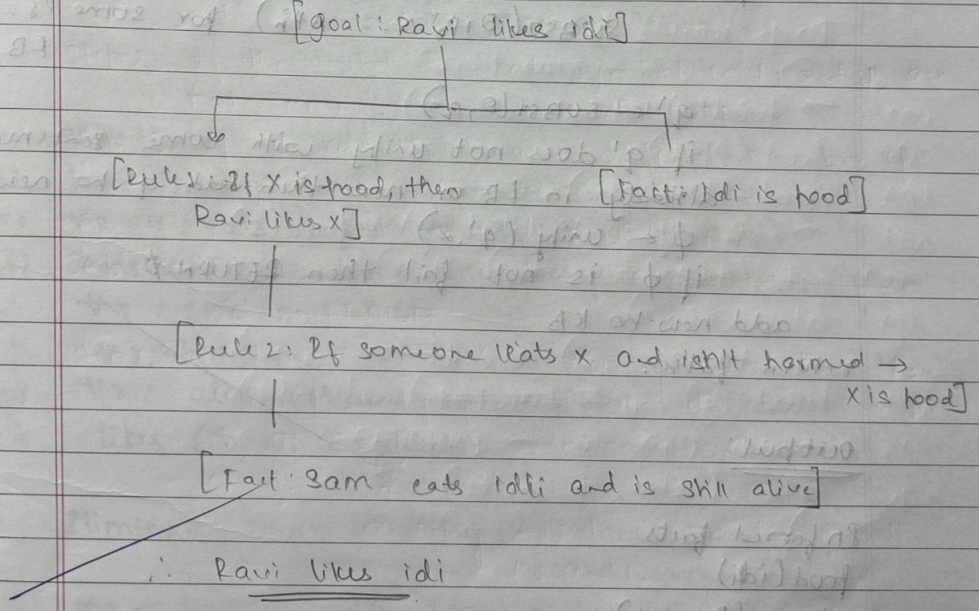
Observation:

WEEK-8

Create a knowledge base consisting of first order logic statements and prove the given query using forward reasoning.

(8) Prove using forward chaining technique

Ravi enjoys a wide variety of foods
Bananas are food
Pizza is food
A food is anything that anyone consumes and isn't harmed by
Sam eats Idli and is still alive
Bill eats everything Sam eats
Prove: Ravi like idli



∴ Ravi likes idli

Algorithm:

function FOL-FC-ASK(KB, α) returns a substitution or false

inputs : KB, the knowledge base, a set of first order definite clauses α, the query, an atomic sentence

local variables: new, the new sentence inferred on each iteration

repeat until new is empty
  new $\leftarrow$ { }
  for each rule in KB do
    $(p_1 \wedge \ldots \wedge p_n \Rightarrow q) \leftarrow$ STRANDARDIZE-VARIABLES(rule)
    for each θ such that SUBST($\theta, p_1 \wedge \ldots \wedge p_n$) = SUBST($\theta, p_1' \wedge \ldots \wedge p_n'$) for some $p_1' \ldots p_n'$ in KB
      $q' \leftarrow$ SUBST(θ, q)
      if q' does not unify with some sentence already in KB or new then add q' to new
      $\phi \leftarrow$ unify(q', α)
      if ϕ is not fail then RETURN ϕ
  add new to KB
return false.