Week – 8

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

Algorithm:

Bill

Week - 8.

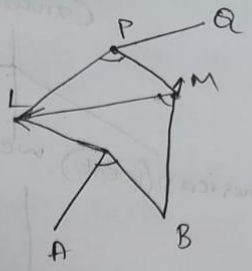
premises ↔ conclusion

$P \Rightarrow Q$

Rules:
- $L \wedge M \Rightarrow P$ ✓
- $B \wedge L \Rightarrow M$ ✓
- $A \wedge P \Rightarrow L$
- $A \wedge B \Rightarrow L$ ✓

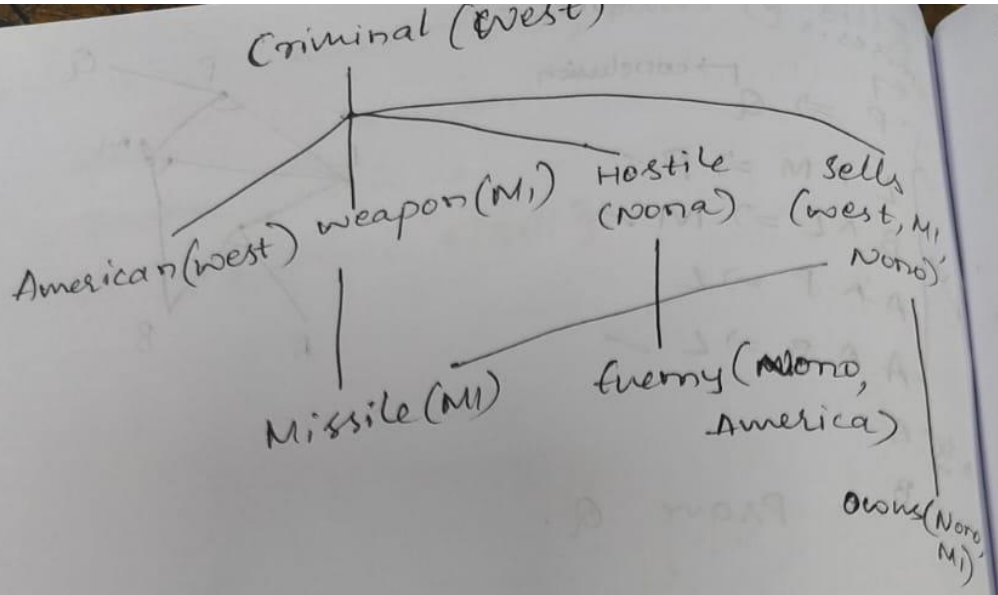
facts:
- A
- B

Prove Q.



Q) The law says that it is a crime for an American to sell weapons to hostile nations. The country Nono, an enemy of America, has some missiles, and all of its missiles were sold to it by Colonel West, who is American. An enemy of America counts as "hostile".
Prove that "West is criminal".

1. $\forall x, y, z$ $\text{America}(x) \wedge \text{Weapon}(y) \wedge \text{Sells}(x, y, z) \wedge \text{Hostile}(z) \Rightarrow \text{Criminal}(x)$
2. $\forall x$ $\text{missile}(x) \wedge \text{owns}(\text{Nono}, x) \Rightarrow \text{Sells}(\text{West}, x, \text{Nono})$
3. ~~$\forall x$~~ ~~$\text{Enemy}(x, \text{America}) \Rightarrow \text{Hostile}(x)$~~
4. ~~$\forall x$~~ $\text{missile}(x) \Rightarrow \text{weapon}(x)$
5. $\text{American}(\text{West})$
6. $\text{Enemy}(\text{Nono}, \text{America})$
7. $\text{owns}(\text{Nono}, M_1)$ and
8. $\text{Missile}(M_1)$



Algorithm:

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

inputs: KB, knowledge Base, a set of first-order definite clauses.

α, the query, an atomic sentence.

local variables: new, the new sentences inferred on each iteration.

repeat until new is empty

new $\leftarrow$ {}

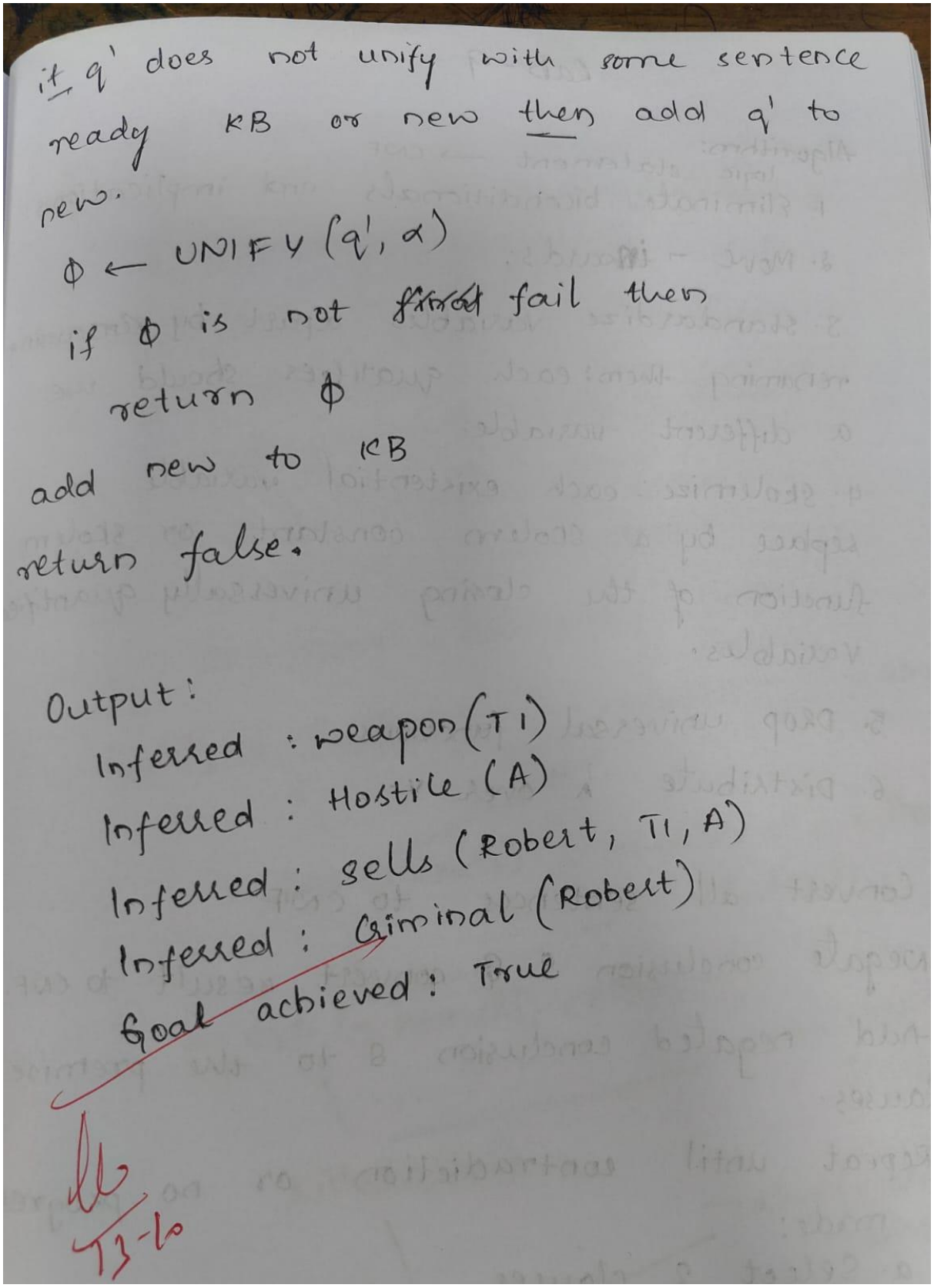
for each rule in KB do

$(p_1 \wedge p_2 \wedge \dots \wedge p_n \Rightarrow q) \leftarrow$ Standardize-Variable(rule)

for each θ such that SUBST(θ, $p_1 \wedge \dots$ p_n) = SUBST(θ, $p'_1 \wedge \dots p'_n$)

for some $p'_1, \dots p'_n$ in KB

$q' \leftarrow$ SUBST(θ, q)


if q' does not unify with some sentence
ready KB or new then add q' to
new.
ϕ ← UNIFY (q', α)
if ϕ is not fail then
return ϕ
add new to KB
return false.
Output:
Inferred : weapon (T1)
Inferred : Hostile (A)
Inferred : sells (Robert, T1, A)
Inferred : Criminal (Robert)
Goal achieved : True

Output:

```
New fact inferred: Criminal(West)
New fact inferred: SoldWeapons(West, Nono)

Final facts:
American(West)
Hostile(Nono)
Missiles(Nono)
Criminal(West)
SoldWeapons(West, Nono)
```

Code:

```
facts = {
    'American(West)': True,
    'Hostile(Nono)': True,
    'Missiles(Nono)': True,
}
def rule1(facts):
    if facts.get('American(West)', False) and facts.get('Hostile(Nono)', False):
        return 'Criminal(West)'
    return None


def rule2(facts):
    if facts.get('Missiles(Nono)', False) and facts.get('Hostile(Nono)', False):
        return 'SoldWeapons(West, Nono)'


def forward_chaining(facts, rules):
    new_facts = facts.copy()
    inferred = True
    while inferred:
        inferred = False
        for rule in rules:
            result = rule(new_facts)
            if result and result not in new_facts:
                new_facts[result] = True
```

```
            inferred = True
            print(f"New fact inferred: {result}")
    return new_facts
rules = [rule1, rule2]

inferred_facts = forward_chaining(facts, rules)

print("\nFinal facts:")
for fact in inferred_facts:
    print(fact)
```