Lab-8

FOL → unification

Statements :-

① Doctors treat patients who are sick.

② John is a doctor

③ Mary is sick

④ Doctors work in hospitals

⑤ General Hospital is a hospital

⑥ John works at General Hospital.

Quantification Statements :-

① $\forall x \forall y (Doctor(x) \wedge Sicky(y) \rightarrow Treats(x, y))$

② Doctor (John)

③ Sick (Mary)

④ $\forall x (Doctor(x) \rightarrow \exists h (Hospital(h) \wedge WorksAt(x, h)))$

⑤ Hospital (General Hospital)

⑥ Works At (John, General Hospital)

Unify Statements :-

$\exists x$ (Treats (x, Mary))

→ (1) From statement (1), unify Treats (x, y) with Treats (x, Mary), binding y = Mary

(2) Statements (3), confirm Sick (Mary), is true, activating statement (1).

(3) Use statement (2) to deduce that Doctor (John) holds, so x = John satisfies the query $\exists x$ (Treats (x, Mary)).

∴ X → John

Y → Mary

Lab 8 - code

```
knowledge_base = [
    {"type": "rule", "rule": "∀x ∀y (Doctor(x) ∧ Sicky(y) → Treats(x, y))"},
    {"type": "fact", "fact": Doctor(John)"},
    {"type": "fact", "fact": Sick(Mary)"},
    {"type": "fact", "fact": ∀x(Doctor(x) → ∃h(hospital(h) ∧ Works At(x, h)))"},
    {"type": "fact", "fact": Hospital(GeneralHospital)"},
    {"type": "fact", "fact": "Works At(John, General Hospital)"},
]

query = {"predicate": "Treats", "arguments": ["?", "Mary"]}

def unify(kb, query):
    predicate = query["predicate"]
    target_arg2 = query["arguments"][1]
    result = None

for item in kb:
    if item["type"] == "rule" and predicate in item["rule"]:
        rule = item["rule"]
        if "Doctor(x)" in rule and "Sick(y)" in rule:
            doctor = None
            sick_person = None
```

```
for fact in kb:
    if fact["type"] == "fact" and "Doctor(" in fact["fact"]:
        doctor = fact["fact"].split("(")[1][:-1]
    if fact["type"] == "fact" and "Sick(" in fact["fact"]:
        sick_person = fact["fact"].split("(")[1][:-1]
    if sick_person == target_arg2:
        result = doctor
        break
if result:
    return f"The query '{query['predicate']}({result}, {target_arg2})' is
    unified: {result} treats {target_arg2}."
else:
    return f"The query '{query['predicate']}({query['args'][0]},
    {target_arg2})' could not be unified with knowledge base."
result = unify(knowledge_base, query)
print(result).
```

Output:-

The query 'Can Access ? Project X' is unified : John Treats Mary