29/12/23

# Lab-7

## ① Knowledge Base Entailment

→ 'KB' logically entails 'S' If all the models that evaluate KB to True also evaluate S to True.

→ Denoted by KB |= S

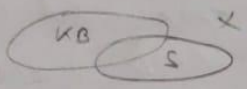
eg:- KB: A ∨ B, ¬C ∨ A  S: A ∧ C

eg:-

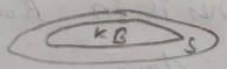
| A | B | C | KB | S |
|---|---|---|---|---|
| F | F | F | F | F |
| F | F | T | F | F |
| F | T | F | T | F |
| F | T | T | F | F |
| T | F | F | T | F |
| T | F | T | T | T |
| T | T | F | T | F |
| T | T | T | T | T |

KB |= S

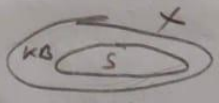
→ KB |= S




KB ⊭ S




KB |= S




KB ⊭ S

## Algo

① So, first will define knowledge base query in code

② then we will define the query_entails in code to satisfies the conditions

③ So, the KB entails 'S', if and only if the KB is 'true' & 'S' is also true.

④ As we have seen in the Truth table & Logically graph.

⑤ So, base on this we say the 'KB' is Logically entails S' or not.

~~from~~

## code

```
from sympy import Symbols, And, Not, Implies, Satisfiable
def create_KB():
    p = Symbols('p')
    q = Symbols('q')
    r = Symbols('r')                    } define

    knowledge_base = And(
        Implies(p, q),                  } p => q
        Implies(q, r)                   } q => r
        Not(r)                          } ~r
    )                                   } KB
    return KB

def query_entails(KB, query):                condition for entails 'S'
    entailment = satisfiable(And(knowledge_base, Not(query)))
    return not entailment

if __name__ == "__main__":
    KB = create_KB()
    query = Symbols('not p')
    result = query_entails(KB, query)        } computation

    print("Knowledge Base:", KB)
    print("query:", query)
    print("query entails KB:", result)
```

O/P

KB : ~r & (Implies(P, q)) & (Implies(q, r))
Query: (not, P)
Query entails KB: False.

Output:

```
Knowledge Base: ~r & (Implies(p, q)) & (Implies(q, r))
Query: p
Query entails Knowledge Base: False
```

```
[ ]  Start coding or generate with AI.
```