Date 9/12/2024
Page

Lab-11

Knowledge base using proposition logic / Entails

Initilize knowledge base with proposition logic statements

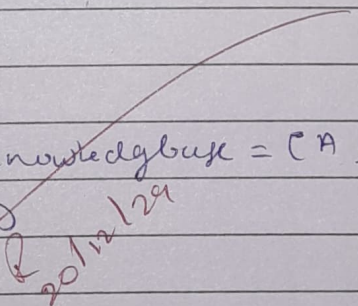
If forward_chaining (knowledge_base, query):
    print "query entailed by knowledge base"
Else
    print "query is not entailed by knowledge base"

function forward-chaining (knowledge_base, query)
    Initilize agenda with known facts from knowledge base
    while agenda is not empty
        pop fact from agenda
        If fact matches query:
            return true
        for each rule in knowledge_base
            If fact satisfies rules premise
            add the rules conclusion to agenda
    Return False

output

For the knowledge base = [A, B, A & B => C, C => D]

query D



query is established by knowledge base

tcrvwebyt

December 21, 2024

[3]:
```
from sympy import symbols, Or, Not, And
from sympy.logic import satisfiable

print("Name: Sudarshan Komar")
print("USN: 1BM22CS291")

# Define variables
x = symbols('x')
P = symbols('P(x)')
Q = symbols('Q(x)')
R = symbols('R(x)')

# Formula Set A (Premises)
A1 = Or(P, Q)         # P(x)   Q(x)
A2 = Or(Not(P), R)     # ¬P(x)   R(x)
A3 = Or(Q, Not(R))     # Q(x)   ¬R(x)

# Formula Set B (Conclusion)
B = Q

# Print the formulas
print("\nSet A (Premises):")
print(f"A1: {A1}")
print(f"A2: {A2}")
print(f"A3: {A3}")

print("\nSet B (Conclusion):")
print(f"B: {B}")

# Check entailment: A entails B if (A   ¬B) is unsatisfiable
entailment_check = satisfiable(And(A1, A2, A3, Not(B)))

if entailment_check:
    print("\nA does not entail B.")
else:
    print("\nA entails B.")
```

```
Name: Sudarshan Komar
```

USN: 1BM22CS291

Set A (Premises):
A1: P(x) | Q(x)
A2: R(x) | ~P(x)
A3: Q(x) | ~R(x)

Set B (Conclusion):
B: Q(x)

A entails B.