AI Assignment - 6

Inference from Knowledge Base

Sabarivasan Velayutham

205001085

CSE-B

1. Show that the hypotheses:

♣ It is not sunny this afternoon and it is colder than yesterday. **¬s ∧ c**

♣ We will go swimming only if it is sunny. **w → s**

♣ If we do not go swimming, then we will take a canoe trip. **¬w → t**

♣ If we take a canoe trip, then we will be home by sunset. **t → h**

lead to the conclusion:

♣ We will be home by sunset. **h**

a) Translate the statements into propositional logic. (Already given in RED)

b) Write a formal proof, a sequence of steps that state hypotheses or apply inference rules to previous steps.

c) Write the same above proof based on resolution also.

Code:

```
def modus_ponens(p1, p2):
  # p->q, p
  # => q
  if '->' in p2 and '->' not in p1:
    p1, p2 = p2, p1

  p1 = p1.split('->')
  if p2 == p1[0]:
    return p1[1]

def modus_tollens(p1, p2):
  # p->q, ~q
  # => ~p
  if '->' in p2 and '->' not in p1:
    p1, p2 = p2, p1

  p1 = p1.split('->')
  p2 = p2.split('~')
  if p2[1] == p1[1]:
    return '~' + p1[0]

# print(modus_ponens("w", "w->s"))
# print(modus_tollens("~s", "w->s"))

def conjunction(p1, p2):
```

```
  return p1 + '^' + p2

# print(conjunction("s","r"))

def simplification(p):
  p = p.split('^')
  return p[0], p[1]

# print(simplification("s^r"))

def addition(s1, s2):
  return s1 + 'v' + s2

# print(addition("S","R"))

# main
print("Premises:-")
kb = ["~s^c", "w->s", "~w->t", "t->h"]
goal = "h"

for i in range(len(kb)):
  print(i+1, ".", kb[i])

print("\nGoal: ", goal)

print("\nProof using inference rules:-\n")
```

```
print("Step\t\tPremise\t\tInference rule")
print("1.\t\t\t", kb[0], "\t\tpremise")
kb.extend(simplification(kb[0]))
print("2.\t\t\t", kb[4], "\t\tsimplification of 1")
print("3.\t\t\t", kb[1], "\t\tpremise")
kb.append(modus_tollens(kb[4],kb[1]))
print("4.\t\t\t", kb[-1], "\t\tmodus tollens of 2,3")
print("5.\t\t\t", kb[2], "\t\tpremise")
kb.append(modus_ponens(kb[-1],kb[2]))
print("6.\t\t\t", kb[-1], "\t\t\tmodus ponens of 4,5")
print("7.\t\t\t", kb[3], "\t\tpremise")
kb.append(modus_ponens(kb[-1],kb[3]))
print("8.\t\t\t", kb[-1], "\t\t\tmodus ponens of 6,7")
print("goal found!")
# print(kb)

def resolution(p1, p2):
  if 'v' in p2 and 'v' not in p1:
    p1,p2 = p2,p1

  p1 = p1.split('v')
  if '~' not in p2:
    if '~' + p2 == p1[0]:
      return p1[1]
    elif '~' + p2 == p1[1]:
      return p1[0]
```

```
  else:
    p2 = p2.split('~')[1]
    if p2 == p1[1]:
      return p1[0]
    elif p2 == p1[0]:
      return p1[1]
# print(resolution('~rvw','~w'))

print("\nConverted to CNF")
kb = ["~s", "c", "~wvs", "wvt", "~tvh"]

for i in range(len(kb)):
  print(i+1, ".", kb[i])

print("\nGoal: ", goal)
print("\nProof using resolution:-")
print("\nStep\t\tPremise\t\tInference rule")
print("1.\t\t\t", kb[0], "\t\tpremise")
print("2.\t\t\t", kb[2], "\t\tpremise")
kb.append(resolution(kb[0],kb[2]))
print("3.\t\t\t", kb[-1], "\t\tresolution of 1,2")
print("4.\t\t\t", kb[3], "\t\tpremise")
kb.append(resolution(kb[-1], kb[3]))
print("5.\t\t\t", kb[-1], "\t\t\tresolution of 3,4")
print("6.\t\t\t", kb[4], "\t\tpremise")
kb.append(resolution(kb[-1], kb[4]))
```

print("7.\t\t\t", kb[-1], "\t\t\tresolution of 5,6")

print("goal found!")

Output:

```
~/AI$ python week6.py
Premises:-
1 . ~s^c
2 . w->s
3 . ~w->t
4 . t->h

Goal:  h

Proof using inference rules:-

Step        Premise     Inference rule
1.           ~s^c       premise
2.           ~s         simplification of 1
3.           w->s       premise
4.           ~w         modus tollens of 2,3
5.           ~w->t      premise
6.           t          modus ponens of 4,5
7.           t->h       premise
8.           h          modus ponens of 6,7
goal found!

Converted to CNF
1 . ~s
2 . c
3 . ~wvs
4 . wvt
5 . ~tvh
```

```
Goal:  h

Proof using resolution:-

Step        Premise     Inference rule
1.           ~s         premise
2.           ~wvs       premise
3.           ~w         resolution of 1,2
4.           wvt        premise
5.           t          resolution of 3,4
6.           ~tvh       premise
7.           h          resolution of 5,6
goal found!
~/AI$
```