Keith Ecker

11/20/2017

HW8

1. KB:
   1.p=>(q=>r)
   Query:
   (p => q) => (p => r)

   Add negation of query to KB

   KB:
   1. p=>(q=>r)
   2. ¬((p=>q) => (p=>r))

   Convert to CNF:
   p => (q => r)
   Modus ponens: ¬p v (¬q v r)
   Distributive law: (¬p v q) ^ (¬p ^r)
   (¬p v q) ^ (p ^¬r)

   ¬((p=>q) => (p=>r))
   Modus pones: ¬(¬(p=>q) v (p=>r))
   Modus pones: ¬(¬(¬p v q) v (¬p v r))
   De Morgan: ¬((p ^ ¬q) v (¬p v r))
   De Morgan: ¬(p^¬q) ^ ¬(¬p v r)
   De Morgan: (¬p v q) ^ (p ^ ¬r)
   (¬p v q) ^ (p ^ ¬r)
   Resolve 1 and 2: empty

2.

1.1 For anything that is able to jump, it can jump higher than a building can.

1.2 Variablesx=something, can be anything.

1.3 Predicates: x, building.
Functions: Jump(predicate) returns true if predicate is able to jump. Examples Jump(Rabbit)=true, Jump(Keith) = true, Jump(rock) = false.
higher_Than(predicate1, building) returns true if predicate1 can jump higher than a building can. Examples higher_Than(rabbit, building) = true, higher_Than(rock, building)=false

1.4 ∀ x Jump(x) => higher_Than(x, building)

2.1 Out of the 100 politicians at the party at least one of them is honest and if you take any two politicians at least one of them is going to be a liar.

2.2 Variables: x, y. x and y = one of the 100 politicians at the party. X cannot equal y.

2.3 Predicates: x, y

Functions: Honest(predicate) returns true if predicate is honest. Example, Honest(Abraham Lincoln) = true
Liar(predicate1, predicate2) returns true if at least one of the two predicates is a liar. Example, Liar(x,y) returns true if x OR y is a liar or if x AND y are liars.

2.4 $\exists x$ Honest(x) = true
$\forall x \forall y$ Liar(x,y) = true

3.1

Cities	Madison	Seattle	Boston	Vancouver	Winnipeg	Montreal
Madison	0	1616.93	930.58	1654.14	597.39	799.51
Seattle	1616.93	0	2485.52	121.34	1152.98	2283.09
Boston	930.58	2485.52	0	2500.64	1344.25	250.36
Vancouver	1654.14	121.34	2500.64	0	1158.70	2290.58
Winnipeg	597.39	1152.98	1344.25	1158.70	0	1132.06
Montreal	799.51	2283.09	250.36	2290.58	1132.06	0

3.2

1.1 Seattle and Vancouver
1.2 Distance=121.34 miles
1.3 Five clusters: (Seattle, Vancouver) (Madison) (Montreal) (Boston) (Winnipeg)

2.1 Montreal and Boston
2.2 Distance = 250.36 miles
2.3 Four clusters: (Seattle, Vancouver) (Montreal, Boston) (Madison) (Winnipeg)

3.1 Winnipeg and Madison
3.2 Distance = 597.39 miles
3.3 Three clusters: (Seattle, Vancouver) (Montreal, Boston) (Madison, Winnipeg)

4.1 (Montreal, Boston) and (Madison, Winnipeg)
4.2 Distance = 1344.25 miles
4.3 Two clusters: (Seattle, Vancouver) (Montreal, Boston, Madison, Winnipeg)

3.3

1.1 Madison and Boston
1.2 Distance = 930.58 miles
1.3 Five clusters: (Madison, Boston) (Vancouver) (Winnipeg) (Montreal) (Seattle)

2.1 Vancouver and Winnipeg
2.2 Distance = 1158.70 miles
2.3 Four clusters: (Madison, Boston) (Vancouver, Winnipeg) (Montreal) (Seattle)

3.1 Montreal and (Vancouver, Winnipeg)
3.2 Distance = 2290.58 miles
3.3 Three clusters: (Madison, Boston) (Vancouver, Winnipeg, Montreal) (Seattle)

4.1 Seattle and (Madison, Boston)
4.2 Distance = 2485.52 miles
4.3 Two clusters: (Madison, Boston, Seattle) (Vancouver, Winnipeg, Montreal)

4.1

1.1 Cluster 1: (0,2,4)	Cluster 2: (6,7,8)
1.2 Updated Center 1: 2	Updated Center 2: 7
1.3 Energy = 20	Energy = 2
2.1Cluster 1: (0,2,4)	Cluster 2: (6,7,8)
2.2 Updated Center 1: 2	Updated Center 2: 7
2.3 Energy = 20	Energy = 2

4.2

1.1 Cluster 1: (0)	Cluster 2: (2, 4, 6, 7, 8)
1.2 Updated Center 1: 0	Updated Center 2: 5.4
1.3 Energy = 0	Energy = 23.2
2.1 Cluster 1: (0,2)	Cluster 2: (4, 6, 7, 8)
2.2 Updated Center 1: 1	Updated Center 2: 6.25
2.3 Energy = 2	Energy = 8.75
3.1 Cluster 1: (0, 2)	Cluster 2: (4, 6, 7, 8)
3.2 Updated Center 1: 1	Updated Center 2: 6.25
3.3 Energy = 2	Energy = 8.75

4.3

I believe the first k-means solution is better because it took less iterations to create two even clusters compared to the second solution which took 3 iterations to create 2 uneven clusters. The solution which is better depends on what is expected from the program.