Propositional Theorem Prover using Resolution Refutation

1. How to Run

Project is developed in Python.

Steps:

1. Change directory to root of the project folder
2. make proposition_main.py exectable by command:
   chmod +x proposition_main.py
3. Exectue proposition_main.py by
   ./proposition_main.py arg0 arg1

where
arg0: path to knowledge base file (default: input.txt)
arg1 (optional): for debug (--debug)

example: ./proposition_main.py test.txt –debug

2. **Data Structures**:

1. **class Clause:**
   To have property of each clause - id, value and its two parents from which it is resplved if its from knowledge base then -1 and -1.
2. **class Clauses:**
   Have list of clauses,
   strPathToFile: to store the path to knowledge base file
   candidateDict: Dictionary which maintain signatures of each clauses to prevent duplicate clauses in future.
   lastClauseID: to keep track of last clause generated
   methods
   to parse input file and get all clauses,
   to add new clause,
   to calculate heuristic value I.e sum of length of two clauses to be resolved, which will be used to maintain priority Queue
   to check whether given pair is resolvable
   to get resolved clauses from given pair
3. **class Solver:**
   used clauses class
   candidates: priority queue (min-heap) to get clause of with best heuristic i.e. minimum length.

Methods:

1. solveByResolution: method which solves the theorem
2. getInitialCandidates: method to get initial candidates from input knowledge

base

Same **Heuristic** is used as mentioned in the problem: it is the summation of length of two clauses which is defined in class Clauses
To **check visited clauses** a separate hash table is used named candidateDict in class Clauses, here key is string format of clause. It helps in preventing duplicate clauses.

Since Heuristic takes smaller clauses first that implies it tends to find solution first.

If Set has been used then it should have removed duplicate values but there would have been trade-off of converting set into list than back to list thats why hashtable is used to prevent duplicates:

	Sammy Sport Shop	**Example1.kb**
Total Number of Iteration	22	5
Max Queue Size	148	11

Propositional rules for Sammy's Sport Shop:
L: For Label (First charcter of clause)
C: For Contains(First character of clause)
O: For Observe (First character clause)
1,2,3: For Block Number (Second character of clause)
B,Y,W: For colors:Both, Yellow, White (Third character of clause)
That implies we have
For Contains: C1W, C1Y, C1B, C2W, C2Y, C2B, C3W, C3Y, C3B
For Labels: L1W L1Y L1B L2W L2Y L2B L3W L3Y L3B
For Observation: O1Y O1W O1B O2Y O2W O2B O3Y O3W O3B
Wrong Labels

1. L1W -> -C1W
2. L1Y -> -C1Y
3. L1B -> -C1B
4. L2W ->-C2W
5. L2Y -> -C2Y
6. L2B -> -C2B
7. L3W -> -C3W
8. L3Y -> -C3Y
9. L3B -> -C3B

Each box contains either one of the colors

1. C1B C1W C1Y
2. C2B C2W C2Y
3. C3B C3W C3Y

If box contains one of colors then it does not contains any other color

1. C1B -> -C1W and -C1Y
2. C1W -> -C1B and -C1Y
3. C1Y -> -C1B and -C1W
4. C2B -> -C2W and -C2Y
5. C2W -> -C2B and -C2Y
6. C2Y -> -C2B and -C2W
7. C3B -> -C3W and -C3Y
8. C3W -> -C3B and -C3Y
9. C3Y -> -C3B and -C3W

If a box contains a particular color than the other boxes does not contain that color

1. C1W -> -C2W and -C3W
2. C1Y -> -C2Y and -C3Y
3. C1B -> -C2B and -C3B
4. C2W -> -C1W and -C3W
5. C2Y -> -C1Y and -C3Y
6. . C2B -> -C1B and -C3B
7. C3W -> -C1W and -C2W
8. C3Y -> -C1Y and -C2Y
9. C3B -> -C1B and -C2B

Based on observation

1. O1W -> C1W or C1B
2. O1Y -> C1Y or C1B
3. O2Y -> C2Y or C2B
4. .O2W -> C2W or C2B
5. O3Y -> C3Y or C3B
6. O3W -> C3W or C3B

Given Facts:

1. O1Y
2. O2W
3. O3Y
4. L1W
5. L2Y
6. L3B

Negation of query That middle box contains white

43.-C2W

CNF of Sammy's Sport Shop is given separate file named : SammyShopKnowledgeBase.txt

Transcript for Sammy's Sport Shop:

0: -O1Y v C1B v C1Y
1: -O1W v C1B v C1W
2: -O2Y v C2B v C2Y
3: -O2W v C2B v C2W
4: -O3Y v C3B v C3Y
5: -O3W v C3B v C3W
6: -C1W v -L1W
7: -C1Y v -L1Y
8: -C1B v -L1B
9: -C2W v -L2W
10: -C2Y v -L2Y
11: -C2B v -L2B
12: -C3W v -L3W
13: -C3Y v -L3Y
14: -C3B v -L3B
15: -C1B v -C1Y
16: -C1B v -C1W
17: -C1W v -C1Y
18: -C1W v -C2W
19: -C1W v -C3W
20: -C1Y v -C2Y
21: -C1Y v -C3Y
22: -C1B v -C2B
23: -C1B v -C3B
24: -C2B v -C2Y
25: -C2B v -C2W

26: -C2W v -C2Y
27: -C2W v -C3W
28: -C2Y v -C3Y
29: -C2B v -C3B
30: -C3B v -C3Y
31: -C3B v -C3W
32: -C3W v -C3Y
33: C1B v C1W v C1Y
34: C2B v C2W v C2Y
35: C3B v C3W v C3Y
36: L1W
37: L2Y
38: L3B
39: O1Y
40: O2W
41: O3Y
42: -C2W
Iteration 1, queue size 98, resolution on 6 and 36
Resolving -C1W v -L1W and L1W
43: -C1W generated from 6 and 36
Iteration 2, queue size 99, resolution on 10 and 37
Resolving -C2Y v -L2Y and L2Y
44: -C2Y generated from 10 and 37
Iteration 3, queue size 100, resolution on 14 and 38
Resolving -C3B v -L3B and L3B
45: -C3B generated from 14 and 38
Iteration 4, queue size 102, resolution on 0 and 39
Resolving -O1Y v C1B v C1Y and O1Y
46: C1B v C1Y generated from 0 and 39
Iteration 5, queue size 110, resolution on 4 and 41
Resolving -O3Y v C3B v C3Y and O3Y
47: C3B v C3Y generated from 4 and 41
Iteration 6, queue size 119, resolution on 45 and 47
Resolving -C3B and C3B v C3Y
48: C3Y generated from 45 and 47
Iteration 7, queue size 123, resolution on 13 and 48
Resolving -C3Y v -L3Y and C3Y
49: -L3Y generated from 13 and 48
Iteration 8, queue size 122, resolution on 21 and 48
Resolving -C1Y v -C3Y and C3Y
50: -C1Y generated from 21 and 48
Iteration 9, queue size 124, resolution on 28 and 48

Resolving -C2Y v -C3Y and C3Y
Iteration 10, queue size 123, resolution on 30 and 48
Resolving -C3B v -C3Y and C3Y
Iteration 11, queue size 122, resolution on 32 and 48
Resolving -C3W v -C3Y and C3Y
51: -C3W generated from 32 and 48
Iteration 12, queue size 123, resolution on 46 and 50
Resolving C1B v C1Y and -C1Y
52: C1B generated from 46 and 50
Iteration 13, queue size 127, resolution on 8 and 52
Resolving -C1B v -L1B and C1B
53: -L1B generated from 8 and 52
Iteration 14, queue size 126, resolution on 15 and 52
Resolving -C1B v -C1Y and C1B
Iteration 15, queue size 125, resolution on 16 and 52
Resolving -C1B v -C1W and C1B
Iteration 16, queue size 124, resolution on 22 and 52
Resolving -C1B v -C2B and C1B
54: -C2B generated from 22 and 52
Iteration 17, queue size 126, resolution on 23 and 52
Resolving -C1B v -C3B and C1B
Iteration 18, queue size 125, resolution on 33 and 50
Resolving C1B v C1W v C1Y and -C1Y
55: C1B v C1W generated from 33 and 50
Iteration 19, queue size 134, resolution on 43 and 55
Resolving -C1W and C1B v C1W
Iteration 20, queue size 133, resolution on 34 and 42
Resolving C2B v C2W v C2Y and -C2W
56: C2B v C2Y generated from 34 and 42
Iteration 21, queue size 143, resolution on 44 and 56
Resolving -C2Y and C2B v C2Y
57: C2B generated from 44 and 56
Iteration 22, queue size 148, resolution on 54 and 57
Resolving -C2B and C2B
success! empty clause found
58: [] [54, 57]
  54: -C2B [22, 52]
    22: -C1B v -C2B [input]
    52: C1B [46, 50]
      46: C1B v C1Y [0, 39]
        0: -O1Y v C1B v C1Y [input]
        39: O1Y [input]

50: -C1Y [21, 48]
21: -C1Y v -C3Y [input]
48: C3Y [45, 47]
45: -C3B [14, 38]
14: -C3B v -L3B [input]
38: L3B [input]
47: C3B v C3Y [4, 41]
4: -O3Y v C3B v C3Y [input]
41: O3Y [input]
57: C2B [44, 56]
44: -C2Y [10, 37]
10: -C2Y v -L2Y [input]
37: L2Y [input]
56: C2B v C2Y [34, 42]
34: C2B v C2W v C2Y [input]
42: -C2W [input]

Transcript for example1.kb:

0: -P v -Q v R v S
1: -A v -R
2: A
3: P
4: Q
5: -S
Iteration 1, queue size 5, resolution on 1 and 2
Resolving -A v -R and A
6: -R generated from 1 and 2
Iteration 2, queue size 5, resolution on 0 and 5
Resolving -P v -Q v R v S and -S
7: -P v -Q v R generated from 0 and 5
Iteration 3, queue size 8, resolution on 3 and 7
Resolving P and -P v -Q v R
8: -Q v R generated from 3 and 7
Iteration 4, queue size 10, resolution on 4 and 8
Resolving Q and -Q v R
9: R generated from 4 and 8
Iteration 5, queue size 11, resolution on 6 and 9
Resolving -R and R
success! empty clause found
10: [] [6, 9]
6: -R [1, 2]

1: -A v -R [input]
2: A [input]
9: R [4, 8]
4: Q [input]
8: -Q v R [3, 7]
3: P [input]
7: -P v -Q v R [0, 5]
0: -P v -Q v R v S [input]
5: -S [input]