

CSE 512 – Winter 2019 – Lab 7

Tuesdays 1:30-3:20pm in JB 359

In this lab you will be implementing **resolution for propositional logic**. At this point, consider this lab's task an exercise in algorithm implementation. *Coming up, we will use resolution "refutation" for the purpose of theorem proving by contradiction.*

Create and edit your own file `resolve_lab7.py`

The input data structure should be the following list:

```
CLS = [['notP', 'notQ', 'R'], ['P', 'R'], ['Q', 'R'], ['notR']]
```

Each sublist is a list of **literals** and the sublist itself is called a **clause**. Literals can be positive or negative; negative literals are prefixed with 'not'. Two literals are **complements** when one is positive and the other is negative, for example: 'notP' and 'P'.

Implement the following algorithm:

Given: set of clauses CLS

While True:

1. Record the current number N of clauses CLS. Set oldN to N. Random shuffle CLS.
2. For (C1,C2) in unordered pairs of clauses from CLS:

If C1 contains a literal L and C2 contains the literal (notL) that is a complement to L,

Create a new clause C3 which is a combination of C1 and C2, but without Literals L and notL.

If C3 is equal to [] ("empty clause"), return with "UNSATISFIABLE"

If C3 contains two literals that are complements, e.g., A and notA, C3 is irrelevant. Continue with the next pairs of clauses. ..

If C3 contains any literal more than once, remove any duplicates.

Now add clause C3 to CLS and increase N by 1.

Exit from the (inner) loop.

3. If N is equal to oldN (no new clauses were added), return with "SATISFIABLE".

The objective of resolution refutation is to prove that the set of clauses is "unsatisfiable". Your program should exhibit behavior similar to the following:

```

>>> res_refute(CLS)
[1.] Resolving ['P', 'R'] and ['notP', 'notQ', 'R'] ...
... new clause ['R', 'notQ']
... ADDED
[2.] Resolving ['P', 'R'] and ['notR'] ...
... new clause ['P']
... ADDED
[3.] Resolving ['notP', 'notQ', 'R'] and ['Q', 'R'] ...
... new clause ['notP', 'R']
... ADDED
[4.] Resolving ['notP', 'notQ', 'R'] and ['notR'] ...
... new clause ['notP', 'notQ']
... ADDED
[5.] Resolving ['notP', 'notQ', 'R'] and ['P'] ...
... new clause ['notQ', 'R']
[6.] Resolving ['Q', 'R'] and ['notR'] ...
... new clause ['Q']
... ADDED
[7.] Resolving ['Q', 'R'] and ['R', 'notQ'] ...
... new clause ['R']
... ADDED
[8.] Resolving ['Q', 'R'] and ['notP', 'notQ'] ...
... new clause ['R', 'notP']
[9.] Resolving ['R', 'notQ'] and ['notR'] ...
... new clause ['notQ']
... ADDED
[10.] Resolving ['R', 'notQ'] and ['Q'] ...
... new clause ['R']
[11.] Resolving ['R', 'notQ'] and ['Q', 'R'] ...
... new clause ['R']
[12.] Resolving ['notR'] and ['P', 'R'] ...
... new clause ['P']
[13.] Resolving ['notR'] and ['notP', 'notQ', 'R'] ...
... new clause ['notP', 'notQ']
[14.] Resolving ['notR'] and ['notP', 'R'] ...
... new clause ['notP']
... ADDED
[15.] Resolving ['notR'] and ['R'] ...
... new clause []
'UNSATISFIABLE :-)'
>>>

```

To receive credit for this lab:

Nothing to be handed in for this lab. Keep working ... and signup on the signup sheet.

Then extend your work to complete **Homework Assignment 3**.