CS420 – Artificial Intelligence

Full name: Phạm Lê Thùy Dung

Student ID: 18125006

Lab02 - Resolution in Propositional Logic

1. **Checklist**
   - ✓ Manipulate the input and output
   - ✓ Implement the PL-Resolution
   - ✓ Implement the David-Putnam algorithm
   - ✓ Provide valid results for the PL-Resolution
   - ✓ Provide valid results for the David-Putnam
   - ✓ Report sufficient information in the document

2. Test-cases

   1) KB = {-A OR B,
          B OR -C,
          A OR -B OR C,
          -B}
     Query: A
     → *Not entail*

   2) KB = {-A OR B,
          B OR -C,
          A OR -B OR C,
          -B}
     Query: -A
     → *Entail*

   3) KB = {A,
          B,
          C,
          -A OR -B OR D,
          -B OR -D OR F,
          -F OR G,
          -A OR -E OR H,
          -A OR -C OR E}
     Query: H
     → *Entail*

   4) KB = {A,
          B,
          C,
          -A OR -B OR D,
          -B OR -D OR F,
          -F OR G,
          -A OR -E OR H,
          -A OR -C OR E}
     Query: -H
     → *Not entail*

   5) Large test. All of alphabet characters are used.
     Query: Q
     → *Entail*

	PL-Resolution	David-Putnam
1	10	9
2	10	8
3	32	47
4	26	52
5	357	378

In the average case, David-Putnam is much more efficient thanks to the elimination of old clauses including the variable after each iteration, so that we do not need to due with those clauses as in PL-Resolution.

In the other hand, after each iteration, PL-Resolution only writes the new clauses added to KB in that iteration while David-Putnam writes all clauses belonging to the KB up to that iterations. This leads to the shorter length of proof of PL-Resolution.