

CSC14003 – Artificial Intelligence

LAB 02: PL RESOLUTION

Problem description

function PL-RESOLUTION(KB, α) **returns** *true* or *false*
inputs: KB , the knowledge base, a sentence in propositional logic
 α , the query, a sentence in propositional logic

$clauses \leftarrow$ the set of clauses in the CNF representation of $KB \wedge \neg\alpha$
 $new \leftarrow \{ \}$
loop do
 for each pair of clauses C_i, C_j **in** $clauses$ **do**
 $resolvents \leftarrow$ PL-RESOLVE(C_i, C_j)
 if $resolvents$ contains the empty clause **then return** *true*
 $new \leftarrow new \cup resolvents$
 if $new \subseteq clauses$ **then return** *false*
 $clauses \leftarrow clauses \cup new$

You are required to implement the PL-RESOLUTION function to determine if a given knowledge base (KB) entails a query α . Both the KB and α are sets of propositional clauses in CNF. As part of the submission, include 5 non-trivial test cases to validate your program's functionality. Additionally, the report should cover an analysis of the efficiency of the PL resolution algorithm and propose potential solutions to overcome its limitations.

Specifications

Input: The file contains the knowledge base (KB) and query α in the following format:

- First line contains a positive integer M , which is the number of clauses in query α .
- M next lines represent the clauses in query α , one clause per line.
- $M + 2$ line contains a positive integer N , which is the number of clauses in the KB.
- N next lines represent the clauses in KB, one clause per line.

A positive literal is denoted by an uppercase letter (A-Z), while a negated literal is denoted by a minus symbol ('-') preceding the literal. Literals are connected using the OR keyword. There may be multiple whitespace characters between literals and keywords.

Output: The file contains the generated clauses and the conclusion in the following format:

- The first line indicates the number of clauses generated in the first loop (M_1).
- The subsequent M_1 lines represent the newly generated clauses in the first loop, including the empty clause represented as "{}".
- Subsequent loops (M_2, \dots, M_n clauses) follow the same format as mentioned above.
- The last line displays the conclusion, indicating whether the KB entails α . Print "YES" if the KB entails α ; otherwise, print "NO".
- Duplicate clauses, meaning clauses that are identical to previously appeared clauses (in the current/previous loop or in the original KB), are disregarded.

Code: The main function should perform the following tasks:

- Read the input data and store it in appropriate data structures.
- Call the PL-Resolution function, which implements the PL-Resolution algorithm.
- Write the output data to the output file in the specified format.

Important notes

- Pay close attention to the definitions of true and false as outlined in the textbook for the PL-RESOLUTION function. Remember to include the negation of the query α .
- The literals within each clause, both in the input data and output data, should be sorted in alphabetical order.
- The entailment check is performed at the end of each loop, rather than after generating a clause.
- The clause $(A \vee B \vee \neg B)$ is considered equivalent to $(A \vee \text{True})$ and thus equivalent to True. Deriving that True is true does not provide useful information. Therefore, any clause that contains two complementary literals can be disregarded.
- It is assumed that the input data is in a valid format.

Here is an example of the provided KB and query α in the input.txt file.

Input.txt	Output.txt	Note
1 -A 4 -A OR B B OR -C A OR -B OR C -B	3 -A B -C 4 -B OR C A OR C A OR -B {} YES	 (-A OR B) resolves with (-B) (-A OR B) resolves with (negative of -A) (B OR -C) resolves with (-B) (A OR -B OR C) resolves with (-A) (A OR -B OR C) resolves with (B) (A OR -B OR C) resolves with (-C) (-B) resolves with (B) KB entails α since an empty clause exists in KB.
1 A 4 -A OR B B OR -C A OR -B OR C -B	2 -C -B OR C 2 -A OR C A OR -B 1 A OR -C 0 NO	 (A OR -B OR C) resolves with (-A) (-A OR B) resolves with (-B OR C) (A OR -B OR C) resolves with (-C) (B OR -C) resolves with (A OR -B) No new sentence found KB does not entail α since no new clause is created and no empty clause is found.

Requirements

No.	Specifications	Scores
1	Read the input data and successfully store it in some data structures.	10%
2	The output file strictly follows the lab specifications.	10%
3	Implement the propositional resolution algorithm.	20%
4	Provide a complete set of clauses and exact conclusion.	30%
5	Five test cases: both input and output files.	10%
6	Discussion on the algorithm's efficiency and suggestions.	20%
Total		100%

Submission preparation

This assignment is to be completed **individually**. To prepare for it, you will need to create a folder that contains various subfolders, including:

- Source: This folder should contain all of your Python files.
- Input: This folder contain five non-trivial test cases of different KB- α pairs.
- Output: This folder contain the corresponding output files to the input files.
- Document: Create a PDF-formatted file that includes a checklist of what you have/have not done and a brief description of the main functions.

When naming the main folder, use your Student ID and compress it in a common format. Contact this email if you have any questions about this lab (quochuyy2000@gmail.com).