Lab 2 - Deduction Machine for Predicates

1. **Introduction**: My program takes a deduction as input and outputs a demonstration of the deduction’s validity. If the program is either valid or invalid it will show the steps needed to demonstrate a proof. Some deductions do not have enough information to relate the premises to the conclusion, and so validity cannot be determined. If the validity can’t be determined an output explaining this situation is printed. ChatGPT is used to write a small part of the program but it’s main use is to compute deductions and build proofs for the validity, invalidity, or lack of proof for deductions.
2. **Process**: My program takes a user input deduction line by line. The program continues to read lines until a line entered contains R or ∴. This last line becomes the conclusion that the previous lines are compared against. All lines have their ASCII logic symbols transformed to the Unicode equivalent. Note that Unicode or ASCII characters may be used for inputs, but all printing and logic processing is done with Unicode logic characters.

After the substitution the deduction is passed to ChatGPT along with instructions guiding ChatGPT to solve the problem in accordance with the lab’s instructions. Which includes, starting at the last premises and going upwards to find matches, taking these newly created premises and further applying valid logical rules until the conclusion can be proven. The conclusion can be found to be valid if it is true for all values, invalid if it is true for no values, or the conclusion can be indeterminate.

let response: CompletionResponse = client

        .send\_message("Give a proof for deductions staring at the end and going to the top, telling me which rules are applied for each step. Don't print an introduction or conclusion. Print the output in a table. Please don't print a header for the table".to\_owned()

     + user\_deduction.as\_str())

        .await?;

*The user\_deduction string is passed along with instructions to ChatGPT*

The table ChatGPT creates is then passed into the print\_table function, which returns a beautified table with even spaced columns and rows. print\_table also removes any additional paragraphs ChatGPT returns outside the table. The formatted table is print to console followed by a “Your deduction is: (in)valid”.

1. **Testing**: I tested the example given in the lab document as well as two problems from page 44 of the book. The input is passed to ChatGPT so instead of running into syntax errors for invalid inputs, you confuse GPT and get messages asking for clarification.

**Testing examples**:

Testing the same deduction twice, we get the correct answer the first time and the incorrect answer the second time. ChatGPT also uses a different approach each time. I have not tuned ChatGPT for this assignment, but it is possible to reduce randomness and possibly reduce these inconsistent results.

A screenshot of a computer

Description automatically generatedA screenshot of a computer program

Description automatically generated

**More Testing examples**:

Page 44, problem 4-d

A computer screen shot of a black screen

Description automatically generated

Page 44, problem 4-f

A computer screen shot of a black screen

Description automatically generated

Including an unrelated question

A screenshot of a computer program

Description automatically generated

1. **Results**: The results are very good when they are correct, but unfortunately my program has no way of verifying whether the given answer is correct or not. Running the program multiple times appears to give the correct answer most of the time but it is possible that there are inputs that always produce incorrect results or at least produce incorrect results a majority of the time.

Wolfram Alpha has a plugin for ChatGPT that allows Chat to call wolfram alpha and produce deterministic answers for math and logic problems, but this functionality is not available through ChatGPT’s API.

Also, by passing a user input into the program there are potential security risks. It is very easy for the user to manipulate the output. For example, if the output of my program was output to a webpage a user could ask to have a javascript program created, and if the output was improperly handled this javascript could end up running, allowing for cross site scripting. Also, the table formatting breaks if you ask a question but don’t give a deduction which looks really ugly.

1. **Conclusion**: I used ChatGPT and programming to solve deductions, but I did it in a different way than my classmate. The common approach was to have Chat create a program that solves deductions. My approach is to have a program that has ChatGPT to solve deductions. Having heard of some of the difficulties my classmates encountered, I am glad to have taken the approach I did. ChatGPT appears to be much better at solving logic than creating programs that solve logic. I would like to point out that my program does follow the lab details as it does use chatgpt as an assistant and chat does consider the premises as specified.
2. **References and Acknowledgements**: As in Lab One. I used stack overflow, the official rust documentation, and ChatGPT to help me learn rust and complete the assignment. All code is my own, except for the remove\_paragraphs function, which is almost exactly as I got it from ChatGPT. I also consulted ChatGPT on how to format the table output, but ended up writing my own code as I didn’t think the AI’s code was very good.

fn remove\_chars\_before\_comma(input: &str) -> &str {

    if let Some(index) = input.find(',') {

        &input[index + 1..]

    } else {

        input

    }

}

ChatGPT’s code above became my function below.

fn remove\_paragraphs(input: String) -> String {

    if let Some(index) = input.find('|') {

        input[index..].to\_string()

    } else {

        input

    }

}

<https://chat.openai.com/share/f6884a86-2d21-47e7-b7d0-3e85ffe38117>