Automatic Theorem Prover

CS386-Assignment

Sahil Jindal	Abhishek Gupta	Rohan Gyani	Mridul Ravi Jain
110020043	110040067	110040001	110040083

Assignment Specifications

- You will have to create an automatic theorem prover for propositional logic.
- The input is any well formed formula in PL.
- The output is yes/no depending on the formula being a theorem or not
- The proof must be SYNTACTIC. You cannot use a truth table.
- Go from 1st principles OR use Deduction theorem
- After outputting the result, you have to DISPLAY the proof path
- You can take human help if stuck in between in the proof. For example, you can ask for a hint as to which axiom will be needed.

Theorem Input Format

- Elements are *propositions* : Capital letters
- Operator is only one : $\rightarrow$ (called implies)
- Special symbol ***f*** (called 'false')
- Two other symbols : '***(***' and '***)***'
- Well formed formula is constructed according to the grammar

  $$WFF \rightarrow P \mid f \mid WFF \rightarrow WFF$$

- Inference rules:
- Modes Ponens
- Axioms:
  - A1: $(A \rightarrow (B \rightarrow A))$
  - A2: $((A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C)))$
  - A3: $(((A \rightarrow f) \rightarrow f) \rightarrow A)$

Techniques Applied

- Every theorem to be proved is brought down to deriving WFF1, WFF2, WFF3….. $\vdash \boldsymbol{f}$

  by applying Deduction Theorem

- Put all existing hypothesis in the proof vector

- All subsequent statements are pushed into this proof vector.

Techniques Applied

- Then in a loop continuously check if one of the following conditions can be applied until proof is reached:
  - Check if Modem Ponens can be applied on any two quantities in the proof vector
  - If there are 2 statements S1 & S2 such that
    - S2: (L$\rightarrow$ S1) $\rightarrow$ R
    - Then apply Axiom1 with **A** as S1 and **B** as L
  - If the LHS of any hypothesis is of the form A $\rightarrow$ (B $\rightarrow$A), then apply Axiom1 on it
  - If any statement is of the form A $\rightarrow$ (B$\rightarrow$C) apply Axiom2
  - If any statement is of the form ((A $\rightarrow$ *f*)$\rightarrow$*f)* apply Axiom3
  - If LHS of any statement is of the form (((A $\rightarrow$ *f*)$\rightarrow$*f*) $\rightarrow$ A) apply Axiom3
  - If none of the above conditions can be applied, then apply Brute Force and finally ask for Human Help

Brute Force

- Pick all elements from the proof vector of the form:
  - P
  - $(P \rightarrow Q)$
- Construct a vector 'X' of these elements
- Generate statements by plugging in the elements from Vector 'X' in placeholders in Axioms (using all permutations). Put these statements in another vector 'Y'.
- For every statement in vector 'Y' and proof vector, check if Modus Ponen can be applied.
- If MP can be applied, put the corresponding statement from Y and the result of MP in the proof vector.

Human Help

- Human help can be provided in the form of:
  - Applying transitivity:
    $(A \rightarrow B)$ and $(B \rightarrow C) \Rightarrow (A \rightarrow C)$
  - Applying contraposition:
    $(A \rightarrow B) \Rightarrow (\sim B \rightarrow \sim A)$
  - Apply De Morgan's 1st law:
    $(\sim(P \wedge Q)) \Rightarrow (\sim P \vee \sim Q)$
  - Apply De Morgan's 2nd law:
    $(\sim(P \vee Q)) \Rightarrow (\sim P \wedge \sim Q)$
  - Apply:
    $f \rightarrow P$
  - Apply any user specified statement(if provable)
  - Apply axioms
  - Continue.

Final Result

- Whenever applying any standard result using Human help, we also print the corresponding proof.

- In the end, we print a final proof that consists of only the relevant statements.