Sequent Calculus Prover: User Guide

Documentation - Abdulrahman Al-Taweel - 15-316

January 14, 2026

Disclaimer: AI-Generated Software

This application was developed with the assistance of Artificial Intelligence. While the logic engine adheres to standard Sequent Calculus rules, errors or edge cases may exist.
We are **not responsible** for any mistakes in logic, parsing, or proof generation. If you encounter a bug or a logical inconsistency, please let us know so we can fix it manually. **Always verify your proofs.**

1 Introduction

The Sequent Calculus Prover is a graphical tool designed to help students and logicians verify propositional formulas. It features a modern interface, real-time proof tree visualization, and automatic LaTeX code generation.

2 Installation

Ensure you have **Python 3** installed.

```
python sequentGen.py
```

3 Input Syntax

The engine accepts both English keywords and standard ASCII symbols.

Connective	**Word Input**	**Symbol Input**
Implication ($\to$)	`implies`	`->`
Conjunction ($\wedge$)	`and`	`&`
Disjunction ($\vee$)	`or`	`|`
Negation ($\neg$)	`not`	`~`
Turnstile ($\vdash$)	`entails`	`|-`

Example: `p -> q, p |- q`

4 Step-by-Step Workflow

4.1 1. Start the Proof

Enter your formula in the top bar and press **Start Proof**. The root node will appear in the tree view on the left.

4.2 2. Select a Target Formula

The application requires explicit user interaction to apply rules:

1. Select the active branch in the **Proof Tree** (left pane).
2. Look at the **Antecedent (LHS)** and **Succedent (RHS)** lists on the right.
3. Click to highlight the specific formula you wish to decompose.

4.3 3. Apply a Rule

Navigate to the **Rule Application** grid at the bottom right. Click the button corresponding to the main connective of your selected formula (e.g., if you selected $A \wedge B$ on the left, click **And Left**).

4.4 4. Closing a Branch

When an atomic formula appears on both the left and right sides of the turnstile (e.g., $P, \Gamma \vdash P, \Delta$), the branch is logically valid. Click **Check Identity (Axiom)** to mark the branch as closed.

5 LaTeX Export

To export your proof:

1. Click the **Export LaTeX** button.
2. Copy the generated code.
3. Paste it into a document that includes the `proof` package.