# seqcalc

A structured sequent calculus wrapper for `bussproofs`

Julian

Version 1.0 — January 31, 2026

**Abstract**

`seqcalc` is a structured wrapper around the `bussproofs` package. It provides a declarative interface for sequent calculus proofs, including rule declaration, rule application, formula normalization, shortcuts, and optional standard rule sets. The goal is to simplify the construction of proof trees while keeping the underlying `bussproofs` layout untouched.

Project page: https://github.com/lambdaphoenix/seqcalc

# Contents

# 1 Introduction

The bussproofs package by Sam Buss provides a flexible system for typesetting proof. The seqcalc package builds on top of it by offering a structured, declarative interface for sequent calculus proofs. Its goal is to simplify the construction of proof trees while preserving the layout and behaviour of bussproofs. seqcalc provides:

- a declarative rule system with automatic macro generation,
- optional formula normalization,
- optional debug output,
- optional shortcut commands,
- optional built-in standard rules,
- wrapper environments for proof trees.

seqcalc does not modify the layout logic of bussproofs. It only adds a structured interface on top. For exapmle usage see Section 8.9.

# 2 Installation

## 2.1 Files

This work currently consists of a single file:

- seqcalc.sty

Place this file somewhere in your TEX tree where LATEX can find it, for example, in a local directory searched by your distribution or in the same directory as your document.

## 2.2 Loading the package

Load seqcalc:

```
\usepackage{seqcalc}
```

You may also pass options to seqcalc, as described in Section 3.

# 3 Package options

Package options are implemented using l3keys2e. All options are optional and have sensible defaults. Options may be passed when loading the package:

## 3.1 normalize-formulas

normalize-formulas=<true|false> Enables formula normalization. Default: true.

Normalization automatically removes redundant separators and whitespaces.

| Normalization example | |
|---|---|
| \seq{A, , , B;;C}{D} | $A, B, C \Rightarrow D$ |

### 3.2 debug

`debug=<true|false>` Prints debug messages to the terminal. Default: `false`.

Example output:

```
Package seqcalc [formula]: Normalized 'A;;B' to 'A,B'
Package seqcalc [rule]: Declaring rule AndR
```

### 3.3 shortcuts

`shortcuts=<true|false>` Enables compact aliases for proof constructs. Default: `false`.

Shortcuts may also be enabled at any point using `\EnableSeqShortcuts`. A full list of shortcuts is given in Section 4.4.

### 3.4 standard-rules

`standard-rules=<true|false>` Loads a built-in set of sequent calculus rules. Default: `false`.

Standard rules may also be enabled at runtime using `\EnableSeqStandardRules`. The predefined rules are listed in Section 6.3.

## 4 Runtime commands

### 4.1 Sequents

`\seq{A}{B}` Typesets the sequent $A \Rightarrow B$.

`\seqL{A}` Typesets $A \Rightarrow$.

`\seqR{A}` Typesets $\Rightarrow A$.

The arguments `A` and `B` represent comma-separated lists of formulas. If formula normalization is enabled, redundant separators, multiple commas, and superfluous whitespace are automatically removed.

### 4.2 Axioms and premises

`\SeqAxiom{A}` Inserts an axiom.

`\SeqPremise{A}` Inserts a premise.

Axioms and premises both introduce leaves in a proof tree. Axioms are closed leaves (overlined), while premises remain open. See Section 4.3 on conclusions for how axioms and premises are combined into inference steps.

### 4.3 Conclusions

`\SeqConclusion[Label]{Cmd}{Formula}[Hint]` General conclusion command.

`\SeqConclusionU[Label]{Formula}[Hint]` Unary inference.

`\SeqConclusionB[Label]{Formula}[Hint]` Binary inference.

A conclusion introduces an inference step.
Arguments:

- `Label` (optional): text placed to the left of the inference bar, typically the name of a rule;
- `Cmd`: a `bussproofs` inference command (`UnaryInfC`, `BinaryInfC`, `TrinaryInfC`);

- `Formula`: the conclusion formula;
- `Hint` (optional): text placed to the right of the inference bar, often used for instantiation information or side conditions.

### 4.4 Shortcuts

When shortcut support is enabled (either via the package option `shortcuts=true` or by calling `\EnableSeqShortcuts`), `seqcalc` defines a set of compact aliases for proof-tree constructs. These shortcuts behave exactly like their long forms:

| Shortcut | Expands to |
|---|---|
| `\SeqAx` | `\SeqAxiom` |
| `\SeqPr` | `\SeqPremise` |
| `\Cc` | `\SeqConclusion` |
| `\CU` | `\SeqConclusionU` |
| `\CB` | `\SeqConclusionB` |

## 5 Environments

### 5.1 seqproof

A wrapper around `prooftree`:

**seqproof**

```
\begin{seqproof}
    \SeqPremise{A}
    \SeqConclusionU{B}
\end{seqproof}
```

$$\frac{A}{B}$$

### 5.2 seqproofinline

A compact inline proof environment:

**seqproofinline**

```
Some text
\begin{seqproofinline}
    \SeqPremise{A}
    \SeqConclusionU{B}
\end{seqproofinline}
and some more text...
```

Some text $\frac{A}{B}$ and some more text...

Inline proofs are useful when a full proof tree would interrupt the flow of text.
For complete examples demonstrating these environments in practice, see Section 7.

## 6 Rule system

A rule in `seqcalc` is a named inference pattern specifying its arity (one, two, or three premises) and an optional display label. Once declared, a rule can be applied using `\SeqRule` or via its automatically generated macro.

### 6.1 Declaring rules

Declare rules using:

```
\SeqCalcDeclareRule{Name}{Premises}[Label]
```

### 6.2 Applying rules

Apply rules with:

```
\SeqRule{Name}{Formula}[Hint]
```

Alternatively, once a rule has been declared, its name becomes a macro of its own:

```
\Name{Formula}[Hint]
```

### 6.3 Standard rules

When the option `standard-rules=true` is used, or when `\EnableSeqStandardRules` is called, the following rules are predeclared:

| Rule | Arity | Label |
|---|---|---|
| NegL | 1 | $(\neg \Rightarrow)$ |
| NegR | 1 | $(\Rightarrow \neg)$ |
| AndL | 1 | $(\wedge \Rightarrow)$ |
| AndR | 2 | $(\Rightarrow \wedge)$ |
| OrL | 2 | $(\vee \Rightarrow)$ |
| OrR | 1 | $(\Rightarrow \vee)$ |
| ImpL | 2 | $(\rightarrow\Rightarrow)$ |
| ImpR | 1 | $(\Rightarrow\rightarrow)$ |
| ForallL | 1 | $(\forall \Rightarrow)$ |
| ForallR | 1 | $(\Rightarrow \forall)$ |
| ExistsL | 1 | $(\exists \Rightarrow)$ |
| ExistsR | 1 | $(\Rightarrow \exists)$ |
| SubL | 1 | $(S \Rightarrow)$ |
| SubR | 1 | $(\Rightarrow S)$ |
| EqL | 1 | $(\Rightarrow=)$ |

All standard rules can be applied either via `\SeqRule` or directly through their shortcut macro, e.g. `\AndR{B}`.

## 7 Examples

This section demonstrates the functionality of `seqcalc` using examples.

## 8 Minimal working example

The following example demonstrates the basic usage of `seqcalc` with a simple unary inference.

## 8.1 Basic axiom and unary conclusion

**Axiom and unary inference**

```
\begin{seqproof}
    \SeqAxiom{A}
    \SeqConclusionU{B}
\end{seqproof}
```

$$\dfrac{\dfrac{}{A}}{B}$$

## 8.2 Binary conclusion

**Binary conclusion**

```
\begin{seqproof}
    \SeqPremise{A}
    \SeqPremise{B}
    \SeqConclusionB{A,B}
\end{seqproof}
```

$$\dfrac{A \qquad B}{A, B}$$

## 8.3 Using shortcuts

**Shortcuts enabled**

```
\EnableSeqShortcuts

\begin{seqproof}
    \SeqAx{A}
    \CU{B}
\end{seqproof}
```

$$\dfrac{\dfrac{}{A}}{B}$$

## 8.4 Declaring and applying a custom rule

**Custom rule**

```
\SeqCalcDeclareRule{MyRule}{2}[\star]

\begin{seqproof}
    \SeqPremise{A}
    \SeqPremise{B}
    \MyRule{A,B}[C]
\end{seqproof}
```

$$(\star)\ \dfrac{A \qquad B}{A, B}\ C$$

## 8.5 Using standard rules

**Standard rule: AndR**

```
\EnableSeqStandardRules

\begin{seqproof}
    \SeqPremise{A\Rightarrow B}
    \SeqPremise{A\Rightarrow C}
    \AndR{A\Rightarrow B\land C}
\end{seqproof}
```

$$(\Rightarrow \land)\ \frac{A \Rightarrow B \qquad A \Rightarrow C}{A \Rightarrow B \land C}$$

## 8.6 Using sequents in prooftree

**Sequents in prooftree**

```
\begin{seqproof}
    \SeqPremise{\seq{A}{B}}
    \SeqConclusionU{\seqR{B, \lnot A}}
\end{seqproof}
```

$$\frac{A \Rightarrow B}{\Rightarrow B, \neg A}$$

## 8.7 Declaring and applying a rule with sequents

**Custom rule with sequents**

```
% Declare a binary rule "Cut"
\SeqCalcDeclareRule{Cut}{2}[\mathrm{
    Cut}]

\begin{seqproof}
    \SeqPremise{\seq{A}{B}}
    \SeqPremise{\seq{B}{C}}
    \Cut{\seq{A}{C}}
\end{seqproof}
```

$$(\text{Cut})\ \frac{A \Rightarrow B \qquad B \Rightarrow C}{A \Rightarrow C}$$

## 8.8 Inline proof

**Inline proof**

```
We can derive $B$ from $A$ using:
\begin{seqproofinline}
    \SeqPremise{A}
    \SeqConclusionU{B}
\end{seqproofinline}
which completes the argument.
```

We can derive $B$ from $A$ using: $\frac{A}{B}$ which completes the argument.

### 8.9 Full derivation:

**Derivation of $A \land B \Rightarrow B \land A$**

```
\EnableSeqStandardRules

\begin{seqproof}
    % Left branch
    \SeqAxiom{\seq{A,B}{A}}

    % Right branch
    \SeqAxiom{\seq{A,B}{B}}

    % Combine
    \AndR{\seq{A,B}{B\land A}}
    \AndL{\seq{A\land B}{B\land A}}
\end{seqproof}
```

$$(\land\Rightarrow)\ \frac{(\Rightarrow\land)\ \dfrac{\dfrac{}{A, B \Rightarrow A} \quad \dfrac{}{A, B \Rightarrow B}}{A, B \Rightarrow B \land A}}{A \land B \Rightarrow B \land A}$$

## 9 Implementation notes

- The rule system is implemented using global token lists.
- Formula normalization uses `l3regex`.
- Proof environments wrap the original `bussproofs` environments.

## 10 License and maintenance

This work may be distributed and/or modified under the conditions of the *LaTeX Project Public License* (LPPL), version 1.3c or any later version. The latest version of this license is available at:

https://www.latex-project.org/lppl/

This work has the LPPL maintenance status *maintained*. The current maintainer of this work is Julian (lambdaphoenix). This work consists of the following files:

- `source/seqcalc.sty`
- `doc/seqcalc-doc.tex` and the generated PDF
- `testfiles/` (test suite for `l3build`)
- `build.lua` (build configuration)
- `README.md`

## 11 Project Information

The `seqcalc` package is developed openly on GitHub. Users are encouraged to report issues, request features, or contribute improvements.

### Repository

- Project page: https://github.com/lambdaphoenix/seqcalc
- Issue tracker: https://github.com/lambdaphoenix/seqcalc/issues

The repository contains the following components:

- **source**/ - the package source file `seqcalc.sty`
- **doc**/ - the documentation source `seqcalc-doc.tex`
- **testfiles**/ - the l3build test suite (`.lvt` and `.tlg` files)
- **build.lua** - the `l3build` configuration
- **README.md** - project overview

The documentation PDF is generated from the files in `doc/`.

### Bug reports and feature requests

If you encounter unexpected behavior, incompatibilities, or missing features, please open an issue on GitHub. When reporting a problem, include:

- a minimal working example
- your LaTeX distribution and version
- the package version number

### Contributing

Contributions are welcome. Pull requests should follow the existing code style and include examples when appropriate.

## 12 Change log

### v1.0 (2026/01/30)

- Initial public version.
- Rule declaration and application system.
- Formula normalization.
- Debug mode.
- Shortcut system.
- Standard rule set.