Sequoia: a Playground for Logicians

Proof Tree Building and Unification

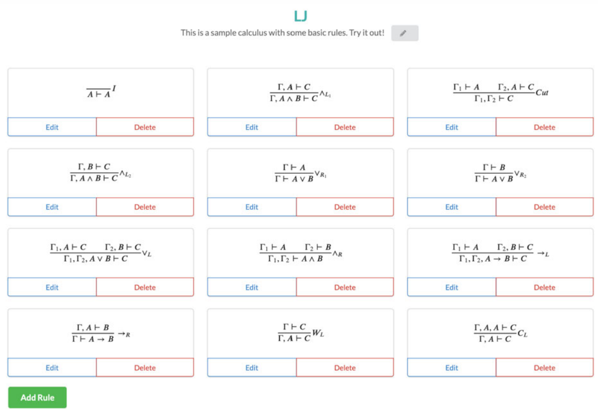
Problem

Sequent calculus is a pervasive technique for studying logics and their properties due to the regularity of rules, proofs, and meta-property proofs across logics. However, composing these proofs brings up a variety of issues:

- Simple proofs can be large and writing them by hand is often messy
- The combinatorial nature of a calculus makes it easy for humans to make mistakes or miss cases
- Formatting the proofs to LaTeX can be tedious and confusing

Solution

Sequoia is a web-based application for specifying sequent calculi and building proofs with theses defined specifications. We provide an intuitive interface where inference rules can be input in LaTeX and are immediately rendered with the corresponding symbols. Neat interactable proof trees can then be built up and transformed based on these user defined input rules.



In the applicative sense we have created a robust automated tree builder by producing valid proof trees from a rule application on a sequent. The problem is how to systematically find all ways to match a rule with a tree sequent so that we can correctly apply the rule to the tree sequent and produce all the valid trees as a result, regardless of the calculus.

We need to substitute certain variables on both the rule and the tree sequent to make them equivalent to a degree. Simple pattern matching will not work since it relies on the substitution to be applied on only one side. However, unification allows substitutions between both sides. Therefore to achieve automated rule application and tree building, our solution involves solving an underlying unification problem.

Core Methods

When a rule is applied to a sequent in a proof tree, for it to apply correctly there must exist at least one valid substitution that transforms the conclusion sequent of the rule to the tree sequent it is applied on.
Our core algorithm for rule application is as follows; consider the following example rule and simple tree:

$$S1 \;\Rightarrow\; \frac{\Gamma_1, A \vdash C \qquad \Gamma_2, B \vdash C}{\Gamma_1, \Gamma_2, A \vee B \vdash C}\vee_L \qquad S2 \;\Rightarrow\; \Theta, Q \vee R, S \vdash R \rightarrow T$$

1) Find all valid unifiers between $S1$ and $S2$:*

$$Unif(\Gamma_1, \Gamma_2, A \vee B \vdash C \;,\; \Theta, Q \vee R, S \vdash R \rightarrow T) \;=$$

$$\left\{ \begin{array}{lllll} \sigma_1: \{A \Rightarrow Q & B \Rightarrow R & C \Rightarrow R \rightarrow T & \Gamma_1 \Rightarrow \{\Theta^1, S\} & \Gamma_2 \Rightarrow \{\Theta^2\}\} \\ \sigma_2: \{A \Rightarrow Q & B \Rightarrow R & C \Rightarrow R \rightarrow T & \Gamma_1 \Rightarrow \{\Theta^1\} & \Gamma_2 \Rightarrow \{\Theta^2, S\}\} \\ \sigma_3: \{S \Rightarrow A \vee B & & C \Rightarrow R \rightarrow T & \Gamma_1 \Rightarrow \{\Theta^1, Q \vee R\} & \Gamma_2 \Rightarrow \{\Theta^2\}\} \\ \sigma_4: \{S \Rightarrow A \vee B & & C \Rightarrow R \rightarrow T & \Gamma_1 \Rightarrow \{\Theta^1\} & \Gamma_2 \Rightarrow \{\Theta^2, Q \vee R\}\} \end{array} \right\}$$
$$\Theta = \Theta^1, \Theta^2$$

2) Discard all substitutions that either map formula terms from $S2$ to $S1$ or map context variables from either sequent to those in $S1$:

$$\left\{ \begin{array}{lllll} \sigma_1: \{A \Rightarrow Q & B \Rightarrow R & C \Rightarrow R \rightarrow T & \Gamma_1 \Rightarrow \{\Theta^1, S\} & \Gamma_2 \Rightarrow \{\Theta^2\}\} \\ \sigma_2: \{A \Rightarrow Q & B \Rightarrow R & C \Rightarrow R \rightarrow T & \Gamma_1 \Rightarrow \{\Theta^1\} & \Gamma_2 \Rightarrow \{\Theta^2, S\}\} \end{array} \right\}$$
$$\Theta = \Theta^1, \Theta^2$$

3) Extend the tree by adding premises above $S1$ as the new premises atop $S2$ and apply each remaining unifier to the tree independently:

$$\sigma_1\left(\frac{\Gamma_1, A \vdash C \qquad \Gamma_2, B \vdash C}{\Theta, Q \vee R, S \vdash R \rightarrow T}\vee_L\right)$$
$$\sigma_2\left(\frac{\Gamma_1, A \vdash C \qquad \Gamma_2, B \vdash C}{\Theta, Q \vee R, S \vdash R \rightarrow T}\vee_L\right)$$

4) Each new set of premises is a valid set of premises that result from a correct application of the rule to $S2$:

$$\frac{\Theta^1, Q, S \vdash R \rightarrow T \qquad \Theta^2, R \vdash R \rightarrow T}{\Theta, Q \vee R, S \vdash R \rightarrow T}\vee_L$$
$$\frac{\Theta^1, Q \vdash R \rightarrow T \qquad \Theta^2, R, S \vdash R \rightarrow T}{\Theta, Q \vee R, S \vdash R \rightarrow T}\vee_L$$

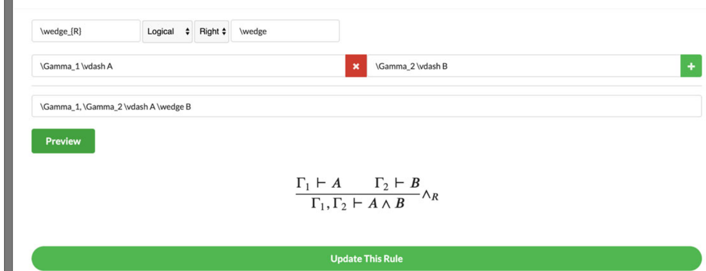
Through this algorithm, one can compute all possible trees that result from a rule application. This algorithm is at the heart of the backend of Sequoia and is used not only for the user tree building, but also in the automated properties testing.

*http://www.gisellereis.com/papers/mset-unif.pdf

Web Application Modules

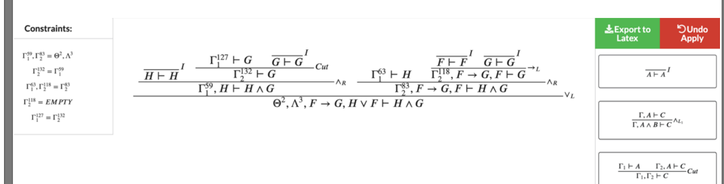
Rule Specification

Users define and edit calculus rules in LaTeX, which are rendered in a preview before they are added to their calculus.
Rules are also specified by important details such as side, type, and main connective if applicable.



Tree Building

Users can build proof valid proof trees using their defined calculus rules and see how these rules interact and affect the structure of the tree as the tree is built.
Proof trees can be concrete or schematic; for the latter there is a context constraint to track the changes in contexts as rules are applied. These trees can then be exported to the LaTeX for simple formatting.



The web application can be accessed at:
https://logic.qatar.cmu.edu/sequoia/
The project source code is available on GitHub:
https://github.com/meta-logic/sequoia

Advisor: Giselle Reis

Zan Naeem & Mohammad Hashim

Carnegie Mellon University Qatar