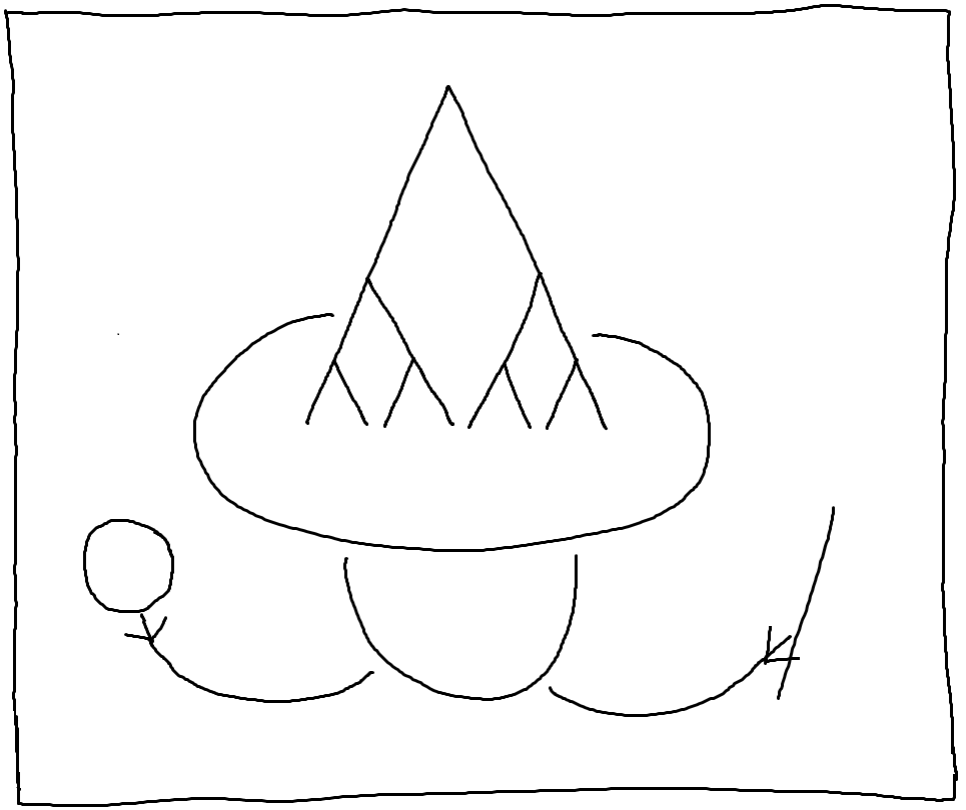
Treehehe

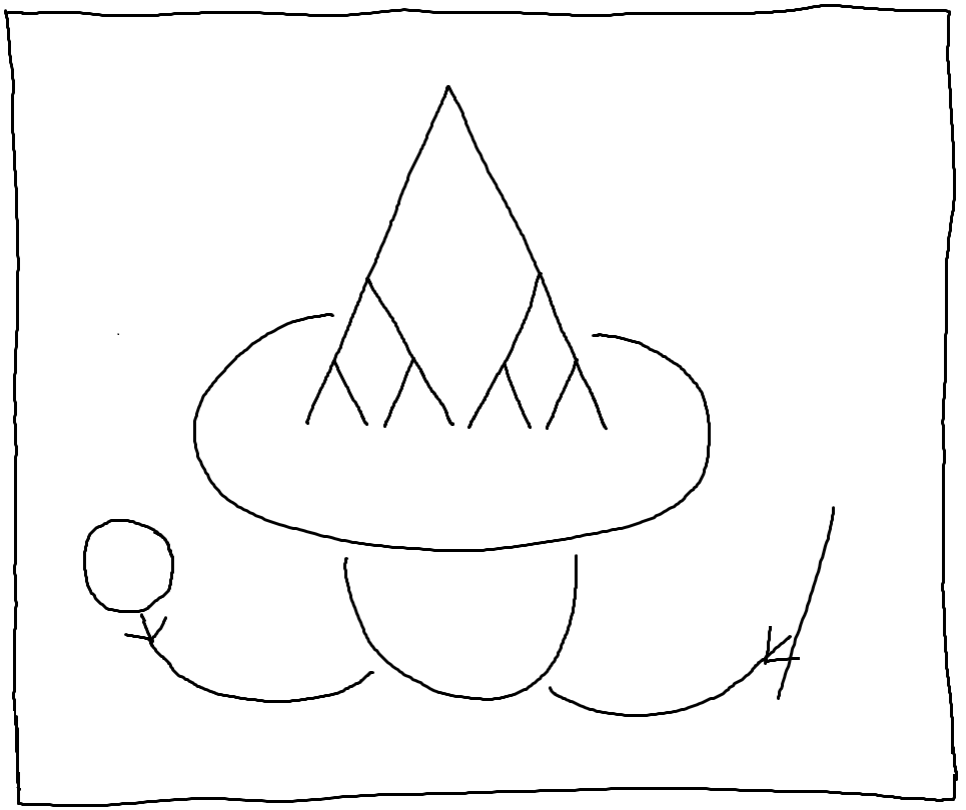
An interactive visualization of proof trees



by Chelsea Battell

Treehehe

An interactive visualization of proof trees



by Chelsea Battell

Application Domain

- Proof: argument obeying logic rules (assumptions ↔ goal)
- Objective: gain insight on proofs and proof systems by exploring proof trees

Super-fast logic introduction:

Formula – an expression we build that is true or false

$$(p \wedge q) \rightarrow r$$

Inference rule – how to build a formula or take it apart, giving meaning to connectives

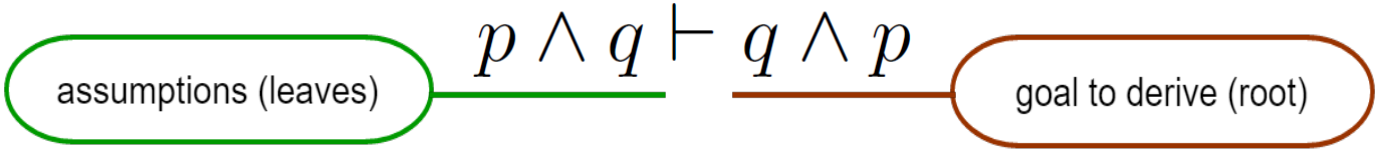
$$\frac{\text{If I have these things...}}{\text{... then I can make this thing}}\,(\text{rule name})$$

$$\frac{P_1 \quad P_2}{P_1 \wedge P_2}\wedge_I \qquad \frac{P_1 \wedge P_2}{P_i}\wedge_{E_i}$$

Proof tree – stick rules together, filling in variables consistently, with assumptions as leaves and goal as root

User Tasks

- Choose an example *sequent*



- Exploring a proof, optionally guided

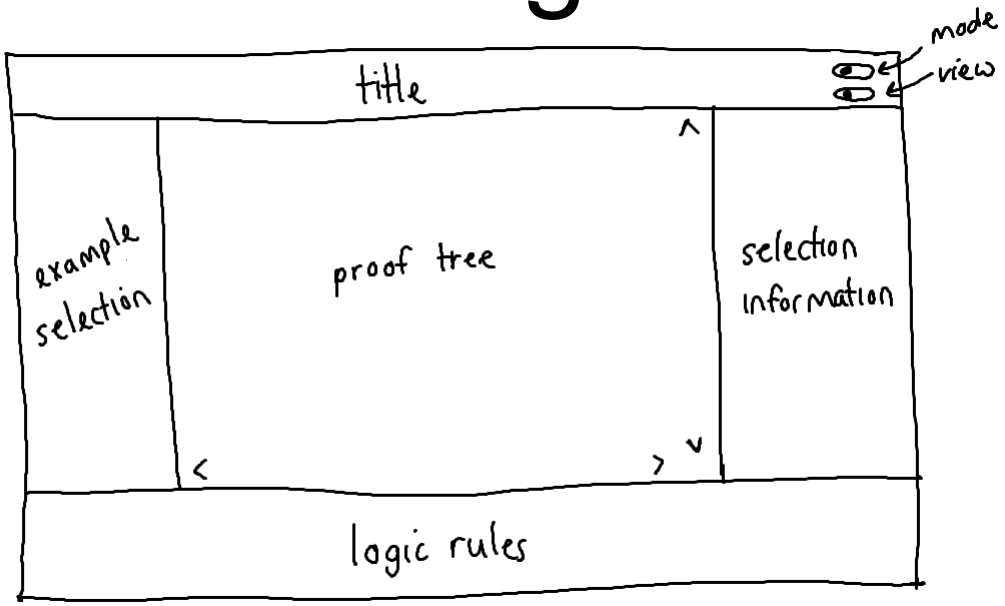
$$\dfrac{\dfrac{\boxed{p \wedge q}}{q}\wedge E_2 \quad \dfrac{\boxed{p \wedge q}}{p}\wedge E_1}{\boxed{q \wedge p}}\wedge I$$

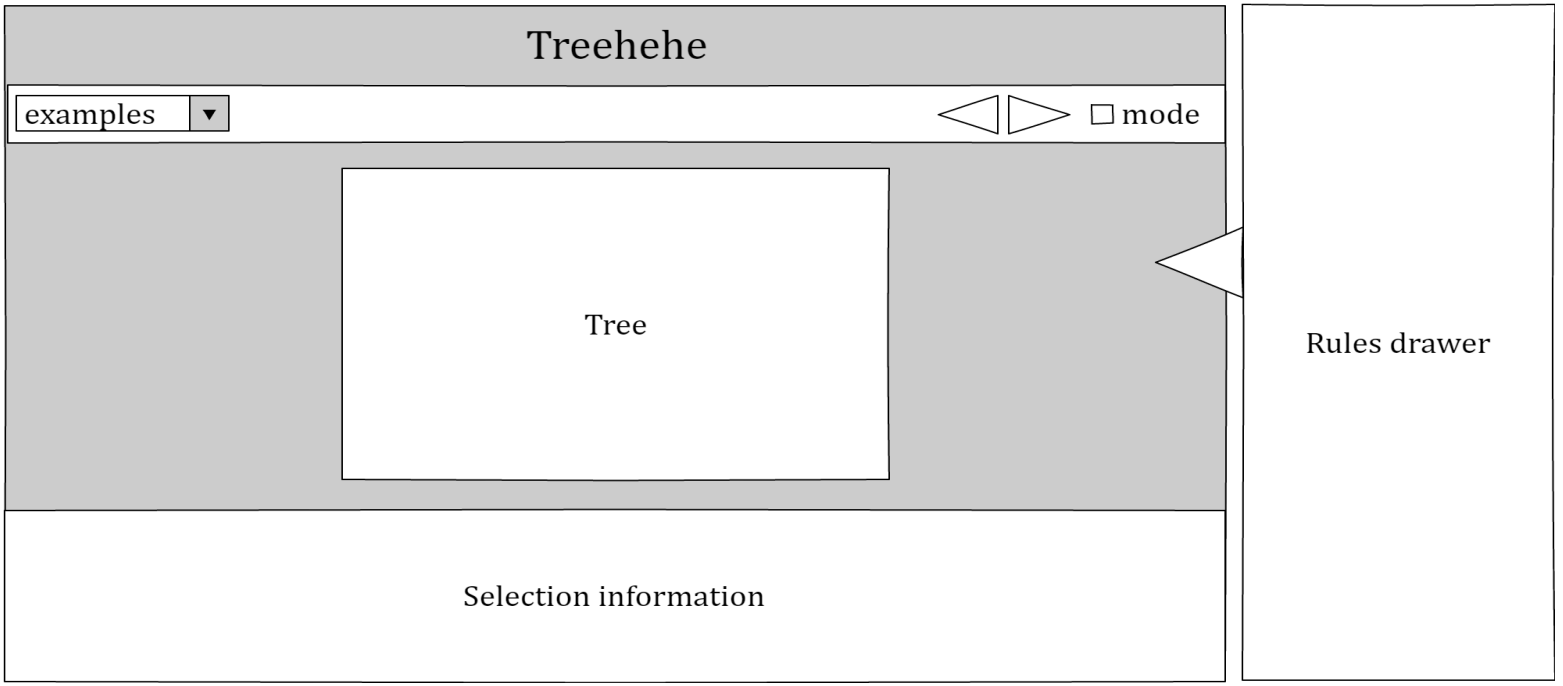
- Viewing supplementary info on selected node
- Reviewing rule sets

InfoVis Elements

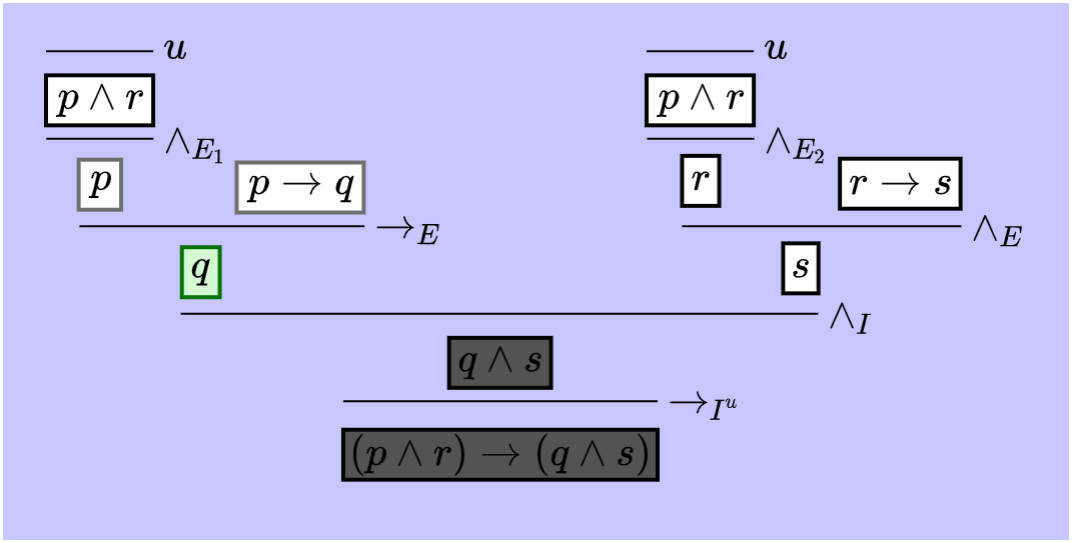
- Tilford-Reingold-Walker algorithm for layout
- Focus+context: children ← focus → parent
- Interaction:
  - Overview/explore: full tree view and navigation
  - Abstract/elaborate: detail view vs structure view
  - Select: display information on focused node
- Colour
  - To highlight a focused node and related nodes
  - Categorizing nodes (e.g. children, visited)

Design Sketches: Page





Design Sketches: Proof



Rules

$$\frac{P_1 \qquad P_2}{P_1 \wedge P_2}\wedge_I$$

$$\frac{P_1 \wedge P_2}{P_1}\wedge_{E_1}$$

$$\frac{P_1 \wedge P_2}{P_2}\wedge_{E_2}$$

$$\frac{\begin{array}{c}[P_1]^u \\ \vdots \\ P_2\end{array}}{P_1 \to P_2}\to_{I^u}$$

$$\frac{P_1 \qquad P_1 \to P_2}{P_2}\to_E$$

Selection

Selected proposition: q

Derived by $\to_E$ applied to $p, p \to q$

Technology

- Main languages: HTML, CSS, JavaScript
  - Fast prototyping, easy sharing
- Laying out math: MathJax
- Tree structure: D3 trees
  - Handles most of layout computation
  - Modifications:
    - Bottom rooted, no links, add inference lines & side content
  - Challenges:
    - Unconventional tree components
    - Varied node widths
    - Sequencing layout using LaTeX

Demo

- Trees for different logics
  - Natural deduction
  - Logic programming
- Remaining tasks:
  - Structure view
  - Minimizing subtrees

Discussion

- Consequences of logic choice
  - Simpler logics see fewer benefits in documentation and insight, but are easier to understand
  - Other visualizations may be better for natural deduction
- Reasoning direction
  - Forward (assumptions to goal) more natural
  - Backward (goal to assumptions) more conducive to automatic reasoning
- Impact of different traversals on cognition
- Remaining tasks:
  - Hiding subtrees
  - Structure view