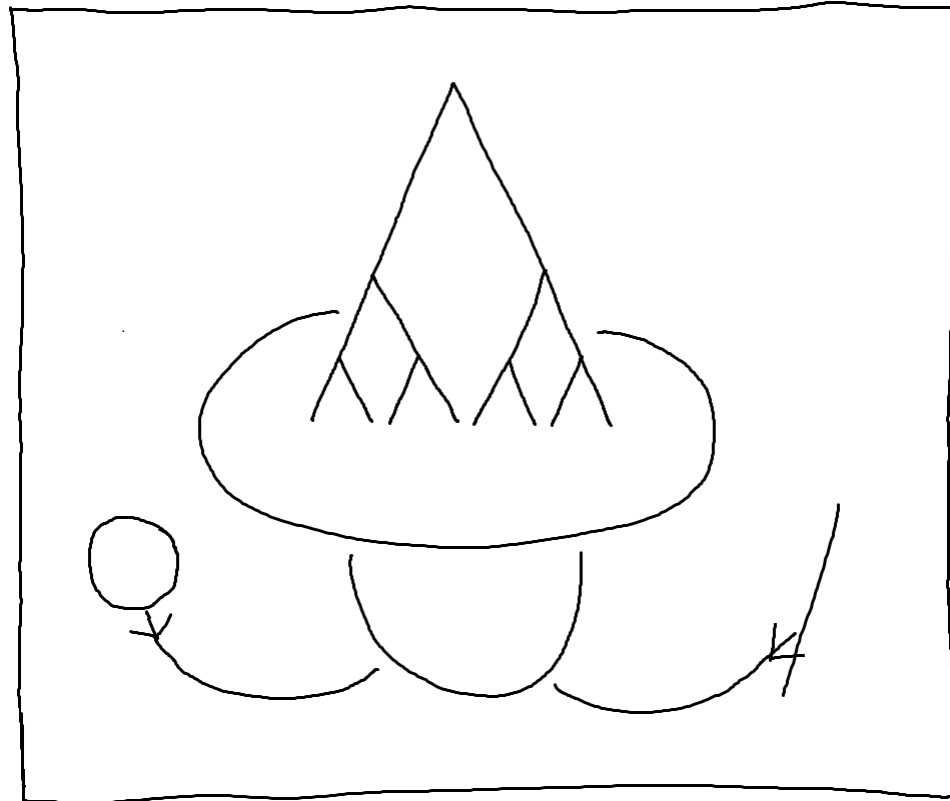


Treehehe

An interactive visualization of proof trees



by Chelsea Battell

Application Domain

- Proof: argument obeying logic rules (assumptions \leftrightarrow goal)
- Objective: explore and gain insight on proofs and proof systems

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 thing or take it apart, giving meaning to connectives

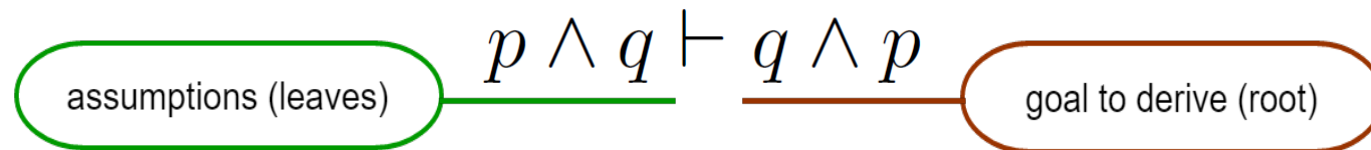
$$\frac{\text{If I have these things...}}{\text{... then I can make this thing}} \text{ (rule name)}$$

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

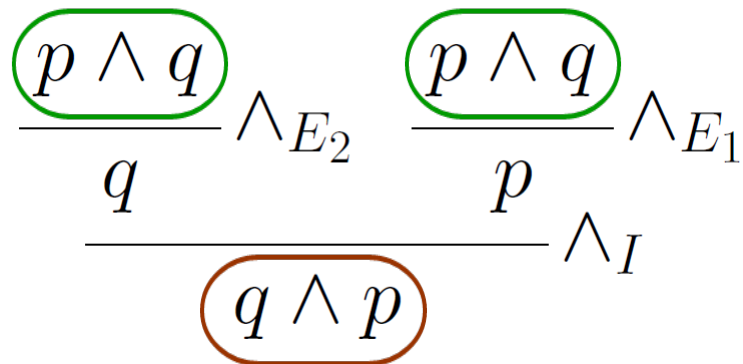
Proof tree – stick rules together, filling in variables consistently, with assumptions as leaves and goal as root

User Tasks

- Choose an example *sequent*



- Exploring a proof, optionally guided

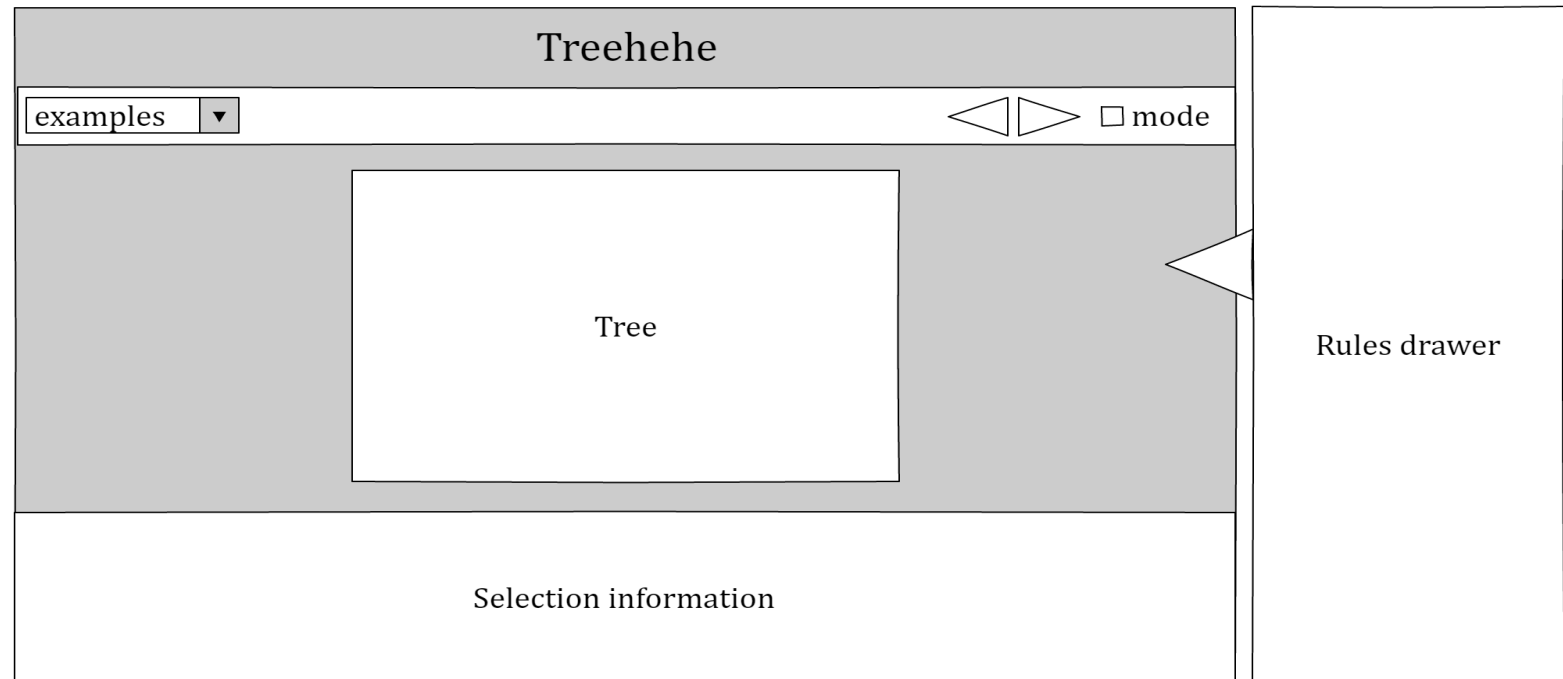
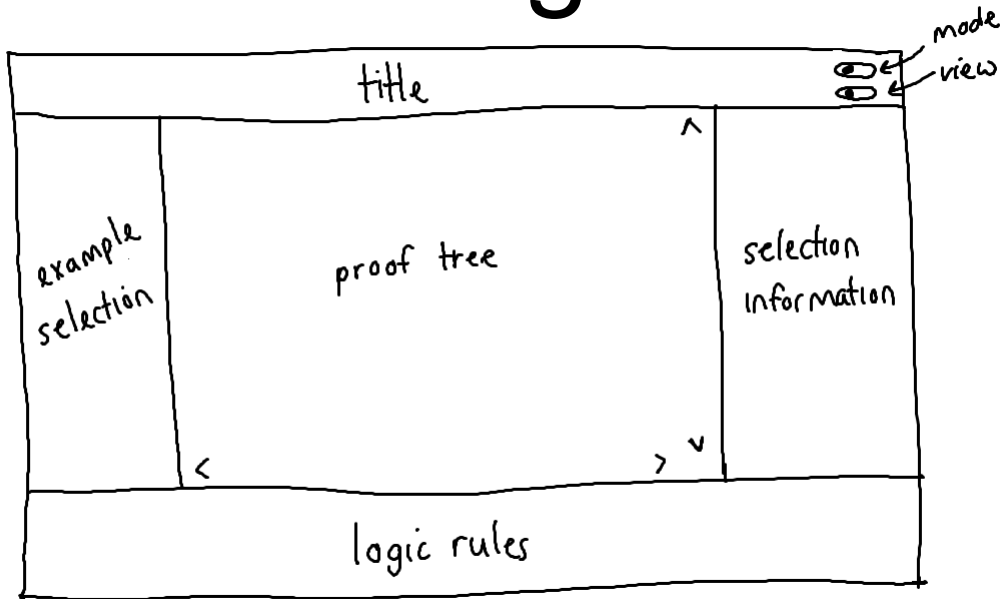


- Viewing supplementary info on selected node
- Reviewing rule sets

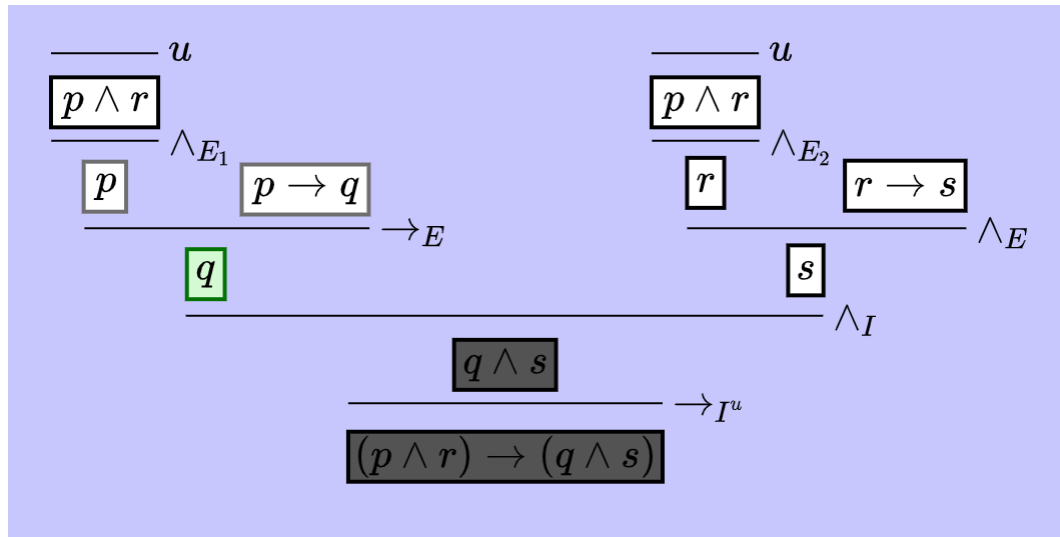
InfoVis Elements

- Tilford-Reingold-Walker algorithm for layout
- Focus+context: children \leftarrow focus \rightarrow 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



Selection

Selected proposition: q

Derived by \rightarrow_E applied to $p, p \rightarrow q$

Rules

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

$$\frac{P_1 \wedge P_2}{P_1} \wedge_{E1}$$

$$\frac{P_1 \wedge P_2}{P_2} \wedge_{E2}$$

$$\frac{\begin{array}{c} \overline{P_1} \\ \vdots \\ \overline{P_2} \end{array}}{P_1 \rightarrow P_2} \rightarrow_{I^u}$$

$$\frac{P_1}{P_2} \frac{P_1 \rightarrow P_2}{\rightarrow_E}$$

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