Towards Complete Tree-Based Proof Search with Metavariables

Asta Halkjær From
Jannis Limperg

Technical University of Denmark
Vrije Universiteit Amsterdam

TCS Seminar, VU Amsterdam
19th May 2022

Tree-Based Proof Search

…without Metavariables

…with Metavariables

Tree-Based Proof Search

…without Metavariables

…with Metavariables

Underlying Logic

- Arbitrary underlying logic with set $\mathbb{G}$ of goals
  - E.g. $A \vdash A \vee B$.
- Arbitrary set $\mathbb{R}$ of rules $R : \mathbb{G} \nrightarrow \mathcal{P}(\mathbb{G})$.

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}$$

```
apply or.intro_left
```

- Rules perform backward reasoning: "to prove G it suffices to prove $R(G)$".

Problem

- Search for proofs involving only rules in $\mathbb{R}$.
- Complete wrt. $\mathbb{R}$: if there is a proof, it will be found.
- Motivation: search tactics like Isabelle's `auto`, Coq's `auto`, Lean's `finish` and soon our `Aesop`, etc.

Tree-Based Proof Search

…without Metavariables

…with Metavariables

Search Trees

- And/or-tree: goal nodes and rule nodes.
- To prove a goal node, prove *one* child rule node.
- To prove a rule node, prove *all* child goal nodes.
  - If zero child goals: rule proves the goal outright.

Search

- Expansion: select a goal node, apply a rule, add rule node and goal nodes.
- Search strategy determines:
  - which node to expand first (e.g. depth-first, breadth-first, best-first);
  - which rule to apply (e.g. by a user-specified priority).

Node Properties

Nodes can be in one of two final states:

- proven: we have a proof
- stuck: we'll never find a proof

Proven and stuck nodes, and their descendants, are irrelevant: we don't need to expand them any more.

Completeness

Definition

An $\mathbb{R}$-derivation is a proof using only rules in $\mathbb{R}$.

Definition

A search strategy is fair if every rule is eventually applied to every goal.

Theorem (Completeness)

Assuming a fair search strategy, if an $\mathbb{R}$-derivation exists for a goal G, the search will prove G.

Completeness

Theorem (Completeness)

Assuming a fair search strategy, if an $\mathbb{R}$-derivation exists for a goal G, the search will prove G.

Proof Outline.

- Let D be the $\mathbb{R}$-derivation of G.
- From D we can generate a sequence of expansions S that apply exactly the rules in D.
- Since the search strategy is fair, every expansion in this sequence will eventually be applied.
  - Except if the expansion is already irrelevant, but then the parent goal must be proven.

$\square$

Tree-Based Proof Search

…without Metavariables

…with Metavariables

Overview

- Goals may contain metavariables $?x$, $?y$, …
- Metavariables stand for arbitrary terms and are solved by unification.
- Allows us to express important rules:

$$\frac{P(?x)}{\exists x, P(x)} \qquad\qquad \frac{R(x, ?y) \qquad R(?y, z)}{R(x, z)}$$

- Key difficulty: goals are not independent any more.
- Solution: when a metavariable is assigned, copy related goals.

Expansion

When a goal node g is expanded with a rule R which assigns metavariables $?x_1, \ldots, ?x_n$:

- Add a rule node r for R.
- Add the subgoals generated by R as children of r.
- For each sibling g' of a goal on the path from g to the root, if g' contains any of the $?x_i$, copy g' as a child of r.

Metavariable Clusters

- Two child goals g_1, g_2 of a rule node r are directly related if they share an unassigned metavariable.
- g_1 and g_2 are related if they are in the equivalence closure of this relation.
- Call this equivalence closure a meta cluster of r.

Proven

- Goal node g is proven if at least one child rule node of g is proven.
- Rule node r is proven if all meta clusters of r are proven.
- Meta cluster c is proven if any of c's goal nodes are proven.

Stuck

- Goal node g is stuck if
  - all child rule nodes of g are stuck and
  - we've applied every possible rule.
- Rule node r is stuck if at least one meta cluster of r is stuck.
- Meta cluster c is stuck if all of c's goal nodes are stuck.

Irrelevant

- A goal node or rule node or meta cluster n is irrelevant if an ancestor of n (including n itself) is proven or stuck.

Soundness and Completeness

- very WIP
- Soundness not trivial any more: need to account for copied goals; metavariable assignments from different branches need to be consistent.
- $\mathbb{R}$-derivation now models an interactive proof, i.e. we transition between partial proofs and rules may assign metavariables that affect arbitrary goals.
- Confluence is probably similar.

Implementation

- Implemented in `Aesop`, a new proof search tactic for Lean.
- Performance seems acceptable on typical (small) examples.
- Enables best-first search without any compromises.

Example

```
variable
  (R : α → α → Prop)
  (R_trans : ∀ x y z, R x y → R y z → R x z)

example : R a b → R b c → R c d → R a d := by
  aesop
```