Aesop: White-Box Best-First Proof Search for Lean

Jannis Limperg\textsuperscript{1}  Asta Halkjær From\textsuperscript{2}

\textsuperscript{1}Vrije Universiteit Amsterdam, jannis@limperg.de
\textsuperscript{2}Technical University of Denmark, ahfrom@dtu.dk

TU Delft, 15 February 2023
White-Box and Black-Box Proof Search

Proof Search Without Metavariables

Proof Search With Metavariables
White-Box and Black-Box Proof Search

Proof Search Without Metavariabes

Proof Search With Metavariabes
Interactive Theorem Proving Is Great

• No functional bugs
Interactive Theorem Proving Is Great

• No functional bugs
• Small kernel, big trust
Interactive Theorem Proving Is Great

- No functional bugs
- Small kernel, big trust
- Can verify everything*
Interactive Theorem Proving Is Great

• No functional bugs
• Small kernel, big trust
• Can verify everything*
• No functional bugs!
Interactive Theorem Proving Is Annoying

• Have to actually understand the code I wrote yesterday.
Interactive Theorem Proving Is Annoying

• Have to actually understand the code I wrote yesterday.

• Have to prove lots of trivialities.
Interactive Theorem Proving Is Annoying

• Have to actually understand the code I wrote yesterday.

• Have to prove lots of trivialities.

\[ a \in [b] \iff a = b \]
Interactive Theorem Proving Is Annoying

• Have to actually understand the code I wrote yesterday.

• Have to prove lots of trivialities.

\[ a \notin xs \rightarrow a \notin ys \rightarrow a \notin xs ++ ys \]
Interactive Theorem Proving Is Annoying

• Have to actually understand the code I wrote yesterday.

• Have to prove lots of trivialities.

\[
\text{map } f \; xs \subseteq \text{map } f \; ys \leftrightarrow xs \subseteq ys \quad \text{if } f \text{ is injective}
\]
How Do We Make It Less Annoying?

1. Convince lots of mathematicians that they should formalise boring stuff for the greater good.
How Do We Make It Less Annoying?

1. Convince lots of mathematicians that they should formalise boring stuff for the greater good.

2. Let the computer do the boring stuff.
## White-Box and Black-Box Proof Search

<table>
<thead>
<tr>
<th><strong>Black-box</strong></th>
<th><strong>White-box</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>hammers</td>
<td>Coq eauto</td>
</tr>
<tr>
<td>SMT solvers</td>
<td>Matita auto</td>
</tr>
<tr>
<td>ML-based provers</td>
<td>Isabelle auto</td>
</tr>
<tr>
<td>Coq sauto</td>
<td>Isabelle auto2</td>
</tr>
<tr>
<td>Agsy</td>
<td>PVS grind</td>
</tr>
<tr>
<td></td>
<td>ACL2 waterfall</td>
</tr>
<tr>
<td></td>
<td>Aesop</td>
</tr>
</tbody>
</table>
White-Box and Black-Box Proof Search

Black-box
fully automatic
powerful
the future

White-box
needs configuration
weak
boring old tech
White-Box and Black-Box Proof Search

**Black-box**
- fully automatic
- powerful
- the future
- complex
- unpredictable
- opaque
- fixed performance
- proof export is hard

**White-box**
- needs configuration
- weak
- boring old tech
- simple
- predictable
- transparent
- customisable performance
- proof export is easy(ish)
White-Box and Black-Box Proof Search

Proof Search Without Metavariabes

Proof Search With Metavariabes
Basic Procedure

\[ \vdash A \rightarrow C \rightarrow A \land (B \lor C) \]
Basic Procedure

⊢ $A \rightarrow C \rightarrow A \land (B \lor C)$

→i

$A, C \vdash A \land (B \lor C)$
Basic Procedure

⊢ \( A \rightarrow C \rightarrow A \land (B \lor C) \)

\(\rightarrow i\)

\(A, C \vdash A \land (B \lor C)\)

\(\land i\)

\(A, C \vdash A\)
\(A, C \vdash B \lor C\)

\(A\)
Basic Procedure

⊢ 𝐴 → 𝐶 → 𝐴 ∧ (𝐵 ∨ 𝐶)

→i

𝐴, 𝐶 ⊢ 𝐴 ∧ (𝐵 ∨ 𝐶)

∧i

𝐴, 𝐶 ⊢ 𝐴

𝐴, 𝐶 ⊢ 𝐵 ∨ 𝐶

∨i-left

𝐴, 𝐶 ⊢ 𝐵

∨i-right

𝐴, 𝐶 ⊢ 𝐶

∧i

𝐴, 𝐵, 𝐶 ⊢ 𝐴 ∧ (𝐵 ∨ 𝐶)
Basic Procedure

\[ \vdash A \rightarrow C \rightarrow A \land (B \lor C) \]

\[ \rightarrow i \]

\[ A, C \vdash A \land (B \lor C) \]

\[ \land i \]

\[ A, C \vdash A \]

\[ \vdash \]

\[ A, C \vdash B \lor C \]

\[ \lor i - \text{left} \]

\[ A, C \vdash B \]

\[ \lor i - \text{right} \]

\[ A, C \vdash C \]
Basic Procedure

\[
\vdash A \rightarrow C \rightarrow A \land (B \lor C)
\]

\rightarrowi

\[
A, C \vdash A \land (B \lor C)
\]

\landi

\[
A, C \vdash A
\]

\[
A, C \vdash B \lor C
\]

\lori-left

\lori-right

\[
A, C \vdash B
\]

\[
A, C \vdash C
\]
Best-First Search

$\vdash A \rightarrow C \rightarrow A \land (B \lor C)$

$\rightarrow i \ 100\%$

$A, C \vdash A \land (B \lor C)$

$\land i \ 100\%$

$A, C \vdash A \land \neg B \lor C$

$\land i \ 100\%

$A, C \vdash A \ 100\%$

$A, C \vdash B \lor C$

$\lor i \ 50\%

$A, C \vdash B \lor C$

$\lor i \ 50\%

$A, C \vdash B \lor C$

$\lor i \ 50\%

$A, C \vdash C \ 100\%$
Best-First Search

\[ \vdash A \rightarrow C \rightarrow A \land (B \lor C) \quad 100\% \]

\[ \rightarrow i \quad 100\% \]

\[ A, C \vdash A \land (B \lor C) \quad 100\% \]

\[ \land i \quad 100\% \]

\[ A, C \vdash A \quad 100\% \quad A, C \vdash B \lor C \quad 100\% \]

\[ A \quad 100\% \quad \lor i \text{-left} \quad 50\% \quad \lor i \text{-right} \quad 50\% \]

\[ A, C \vdash B \quad 50\% \quad A, C \vdash C \quad 50\% \]

\[ C \quad 100\% \]
Extensions of the Basic Procedure

Apply normalisation rules (esp. simplifier) ➔ progress

no progress

Apply safe rules ➔ progress ✓

no progress

Apply unsafe rules

no progress

✓

✓
Apply normalisation rules (esp. simplifier)
Apply normalisation rules (esp. simplifier) → progress

Apply safe rules → progress

\[ \Gamma \vdash A \quad \Gamma \vdash B \]

\[ \Gamma \vdash A \land B \quad \land i \]
Apply normalisation rules (esp. simplifier) → progress

Apply safe rules → ✓

Apply unsafe rules → ✓

no progress

no progress

no progress
White-Box and Black-Box Proof Search

Proof Search Without Metavariabes

Proof Search With Metavariabes
Proof Search With Metavariabes

\[ a < b, \ a < c, \ b < z \vdash a < z \]
Proof Search With Metavariables

\[ a < b, \ a < c, \ b < z \vdash a < z \]

\[ \text{<-trans} \]

\[ \cdots \vdash a < ?x \quad \cdots \vdash ?x < z \]
Proof Search With Metavariables

\[ a < b, a < c, b < z \vdash a < z \]

\[ \text{<-trans} \]

\[ \ldots \vdash a < ?x \quad \ldots \vdash c < z \]

\[ \begin{array}{c}
  a < c \\
  [?x := c]
\end{array} \]
Proof Search With Metavarsiables

\[ a < b, a < c, b < z \vdash a < z \]

\[ a < c [?x := c] \]

\[ \cdots \vdash ?x < z \]

\[ \vdash c < z \]
Proof Search With Metavariables

\[ a < b, a < c, b < z \vdash a < z \]

\(-\text{trans}\)

\[ \cdots \vdash a < ?x \quad \cdots \vdash ?x < z \]

\[ a < c \ [?x := c] \quad a < b \ [?x := b] \]

\[ \cdots \vdash c < z \quad \cdots \vdash b < z \]
Proof Search With Metavariables

\[ a < b, \ a < c, \ b < z \vdash a < z \]

\[ \text{<-trans} \]

\[ \vdash a < ?x \quad \vdash ?x < z \]

\[ a < c \ [?x := c] \]
\[ \vdash c < z \]

\[ a < b \ [?x := b] \]
\[ \vdash b < z \]

\[ b < z \]
Also in the Paper

- Built-in rules
- Rule indexing
- UI for adding common sorts of rules
- Case studies
  - 173 basic lemmas about lists (Aesop + induction proves 94%)
  - Sequent calculus prover
Aesop: White-Box Best-First Proof Search for Lean

Jannis Limperg\textsuperscript{1}   Asta Halkjær From\textsuperscript{2}

\textsuperscript{1}Vrije Universiteit Amsterdam, jannis@limperg.de
\textsuperscript{2}Technical University of Denmark, ahfrom@dtu.dk

TU Delft, 15 February 2023