6J-03

定理証明支援系 Coq と連携した証明木表示機能の設計と実装

木村 麻衣† 田中 雄太* 川端英之‡ 弘中哲夫‡
広島市立大学情報科学部情報工学科† 関電システムソリューションズ株式会社* 広島市立大学大学院情報科学研究科‡

1 はじめに

数学の定理証明やソフトウェア検証において，ある性質を形式的に証明するために，定理証明支援系 Coq[1]が利用されている．Coq を用いた証明スクリプトは手続き的な記述であるため，可読性が高いとは言い難い．これに対して，証明の過程をグラフィカルに表示する機能を導入することは，対話的な証明手続きの各時点での状況把握や完了した証明の俯瞰の補助に有用であると言える．

Coq は証明支援ツール ProofGeneral[2]を介して，対話的な証明の過程をグラフ構造で描画するツール Prooftree[3]と連携することができる．しかし，Prooftree は適用した推論規則の流れのみを簡易的に描画するため，証明の詳細を把握したい場合の使用には向いていない．

本研究では，Prooftree を改造して証明過程の詳細を証明木でグラフィカルに表示する新たな証明支援ツール Traf[4]を作成した．本発表では，描画する情報量を適切に抑え，自然演繹の簡略表記で用いられる読みやすいスタイルで証明を提示できる機能の設計と実装について報告する．

2 Prooftree

Prooftree は LablGtk バインディング[5]を用いて OCaml[6]で記述されている．また，Coq が標準出力する各ステップの証明情報を ProofGeneral を介してほぼそのままの形で文字列として受ける取る．受け取った情報は，現ステップまでの証明をグラフ構造で表示できるように木構造のデータを Prooftree 内で構築して描画している．Prooftree で描画される導出過程は図１に示すように各証明ステップにおけるサブゴールと仮定をひとつのノードとして扱い，極度に簡略化した形での提示である．したがって，Prooftree では証明の詳細を一目で確認することが難しい．

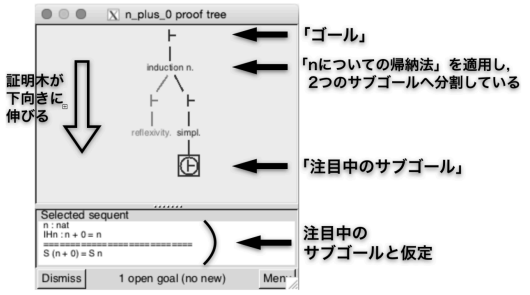


図１：Prooftree を用いた表示

3 Traf：Prooftree の拡張

3.1 証明木と証明スクリプトのギャップ

形式的な証明[7]を紙面で行う場合は，証明をいくつかの規則に基づいて構成される有限の木（証明木）として表現することが多い．証明木は証明そのものである．しかし，Coq を用いた証明スクリプトは証明を適用する推論規則のみで記述しているため，証明の詳細が分からない．これに対して，紙面で行う証明と同様な表現で証明を提示することで，現ステップまでの証明を詳しく確認することができる．さらに，推論規則適用前後のサブゴールを同じ画面で比較できるため，推論規則の適用効果も確認しやすい．また，完了した証明に対する可読性の高い補助資料として保管することができる．

3.2 自然演繹の簡略表記で用いられる読みやすい証明スタイルの導入

証明過程を証明木で表示する場合，いくつかのパターンの証明木表示スタイルが考えられる．例えば，図２のように「仮定 ⊢論理式」の形のノードの木は各証明ステップの詳細を厳密に表示することができる．しかし，全て正確に表現した証明木は，その証明ステップで必要ない仮定の情報を含む場合が多い．そこで，描画する情報量を適切に抑えて可読性の高い証明木を表示するために，図３のような自然演繹の簡略表記で用いられる読みやすい証明スタイルを採用し，実装した．

図３の証明では，Prooftree が ProofGeneral から受け取る各証明ステップの情報を踏まえ，推論規則の適用時に使用された仮定を明記している．これにより，仮定のディスチャージの様子

Design and Implementation of a Functionality to Cooperate with the Theorem Prover Coq for Displaying Proof Trees
†Mai Kimura *Yuta Tanaka
‡Hideyuki Kawabata ‡Tetsuo Hironaka
†Department of Computer and Network Engineering, Hiroshima City University * Kanden System Solutions Co.,Inc.
‡Graduate School of Information Sciences, Hiroshima City University

Copyright ©2018 Information Processing Society of Japan.
All Rights Reserved.

が把握できるように描画されている．

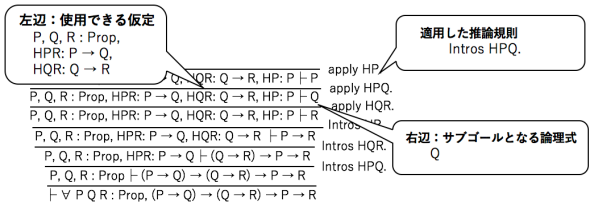


図２：証明木表示スタイルの例

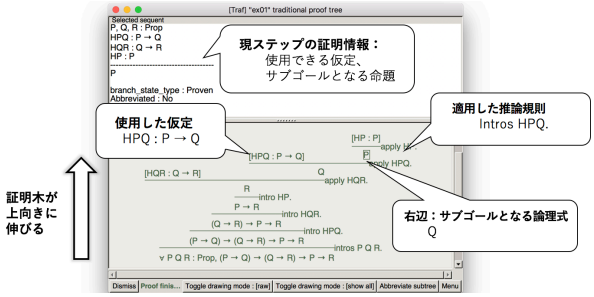


図３：自然演繹を用いた証明スタイル

3.3 推論規則適用前後の論理式の差異を強調表示

サブゴールを導き出す手続きは推論規則に従って遂行される．したがって，推論規則を適用したことによるサブゴールの変化を理解することは，証明の正確な把握のために重要である．

Coq を用いた証明では，ある推論規則を適用した後のサブゴールとなる論理式は適用する前と比べて，論理式の一部のみが変化している場合が多い．特に論理式の記述が長い場合は，推論規則を適用した効果がどこに現れているか分かりにくい．そこで，図４のように推論規則適用前後の論理式の差異を強調した証明木のスタイルでも表示できる機能を次のように実装した．

まず，Coq で標準的に用いられる命題記述構文を踏まえて，推論規則適用前後のサブゴールの文字列表記をそれぞれ構文解析し，構文木ベースで差異を検出するようにした．次に，差異に現れていない部分木はひとまとめにして省略表示するようにした．枝の一部に差異が認められない場合は，要となる記号は省略せず，変化した部位を一目で把握できるように配慮した．

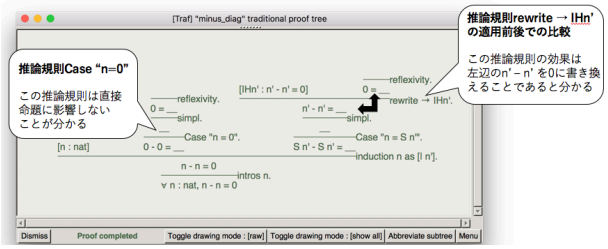


図４：推論規則適用前後の省略

3.4 各ノードの簡略化

画面内の情報量が増えて煩雑になった場合や複数の分岐が現れた場合に，ユーザが現在注目しているゴール情報のみを詳細に表示し，他の部分を簡略化した状態で表示できる機能を実装した．さらには，ユーザが選択したノード以降を詳細に表示する機能も実装した．

3.5 考察

自然演繹の簡略表記で用いられる証明スタイルは，数理論理学で利用される証明と同じ形式で表示するため，Coq の使用経験が浅い人でも親しみやすいスタイルである．また，論理式の差異を抽出することで，推論規則の効果を正確に理解できていない初学者も容易に効果を把握することができる．

証明過程における場合分けの数は，各ノードの簡略化機能を用いて容易に把握ができるため Prooftree と同様な役割を持っている．この機能を用いて場合分けが多いと判断したときには，現在の証明方法に改善の余地がある可能性があり，複数の推論規則組み合わせたものを適用するなど試行錯誤して，もう少し工夫した証明を行うべきであることがわかる．

4 まとめと今後の課題

本研究では，証明手続きの各時点での状況把握や完了した証明の俯瞰の補助など証明の詳細な把握できるような証明木を Prooftree の改造により設計・実装した．今後の課題として，Coq はユーザが記号を定義して扱うことができるので，その記号が入力として与えられた場合にエラーが生じることが挙げられるため，その改善が挙げられる．

参考文献

[1] The Coq Proof Assistant, https://coq.inria.fr/.
[2] Proof General, http://proofgeneral.github.io/.
[3] Prooftree, http://askra.de/software/prooftree/.
[4] 田中 雄太，川端 英之，弘中 哲夫:定理証明支援系を証明木ベースで利用するための Proof General の外部ツールの開発，日本ソフトウェア科学会第 33 回大会，2016
[5] LablGtk, http://lablgtk.forge.ocamlcore.org/ .
[6] OCaml, http://ocaml.org/ .
[7] IPA, 「なぜ形式手法か A」, 2012

Copyright ©2018 Information Processing Society of Japan.
All Rights Reserved.