1L-08

定理証明支援系 Coq と連携した証明木図示ツールの Visual Sudio Code プラグインとしての実装

深澤貴仁† 川端英之‡ 弘中哲夫‡
広島市立大学情報科学部情報工学科† 広島市立大学大学院情報科学研究科‡

1 はじめに

定理証明支援系である Coq [1] は、数学の定理の形式化やソフトウェアの安全性の検証を補助するツールであり、様々な場面で用いられている。そして Coq の使いやすくするツールとしてユーザ証明記述を補助する ProofGeneral [2] や、証明過程を証明木として描画する Traf [3] などのインタフェースツールがある。

これらのユーザ支援機能は個別に開発されており、現在統合した環境とはなっていない。しかし、統合した環境が実現できれば、ユーザにとっての Coq 使用時の利便性向上を図れる。

本研究では、VSCode [4] 内での統合した Coq 利用環境を設計する一環として、Traf の証明木描画を模した描画ツールを Visual Studio Code プラグインで設計した。本稿では、今回設計した描画ツールのプロトタイプについて述べる。

2 証明木図示ツール Traf

Traf は、定理証明支援系の初学者を対象にした、Prooftree [5] をもとに開発された証明木図示ツールである。Traf の特筆する点として以下が挙げられる。 [6]

- 証明過程の詳細をグラフィカルな証明木で確認できる
- 使用した仮定をその都度表示できる

これらにより Traf は Coq 初学者の証明をより効率的にサポートすることができる。Traf を用いて表示した証明過程の例を図 1 に示す。

3 Visual Studio Code プラグインとしての Traf の実装

3.1 Traf の現状と拡張案

Traf は PG からの標準出力経由で証明の進行情報を受け取り証明木を作成しているため、Traf は PG (emacs) 以外で証明を行う場合には対応していない。そこで、次のような Traf の拡張を検討した。

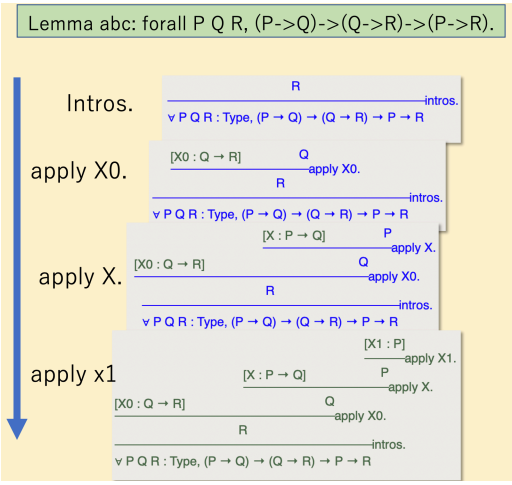

図 1: Traf を用いて表示した証明過程の例

1. PG (emacs) 以外で証明を行う際にも Traf を利用できるようにする
2. PG (Coq インタフェース) から独立したツールである Traf を PG に組み込む

1 について、Traf は Coq 初学者を対象としたツールである。Traf の利用手段が増えれば自分の使い慣れた、もしくは使いたいツール (エディタ) で証明を記述することができるようになるため、より効率的に初学者の証明をサポートすることができる。

2 について、Traf は PG とは独立した外部ツールであるため、Traf インストールのための描画用ライブラリのインストール、Coq や PG のアップデート等に伴うバージョン管理などに手間が掛かる。Traf を PG に組み込むことによってこれらの手間を省くことができるため、ユーザはより Traf を利用しやすくなる。

これらの改良案に対し、例えば VSCode [4] を用いれば Coq のインタフェースと証明木描画ツールを統合でき、スムーズな開発、保守環境が実現できる可能性がある。

3.2 解決のアプローチ

我々は上記の改良案を踏まえた、VSCode プラグインを活用した新しい証明木図示ツールの設計を検討した。この証明木図示ツールは、既存の VSCode 上での Coq のインタフェースである VSCoq [7] を拡張し、証明木描画機能を埋め込むことで設計可能である。VSCoq が Coq とのやりとりを行い、その結果を証明木描画機能が VSCoq から受け取り証明木を描画することを想定している。図 2 にその構造図を示す。

Implementing a Graphical Proof Tree Viewer as a Visual Studio Code Plugin That Cooperates with the Coq Theorem Prover
Takahito Fukasawa† Hideyuki Kawabata‡ Tetsuo Hironaka‡
†Department of Computer and Network Engineering, Hiroshima City University
‡Graduate School of Information Sciences, Hiroshima City University

Copyright ©2022 Information Processing Society of Japan.
All Rights Reserved.

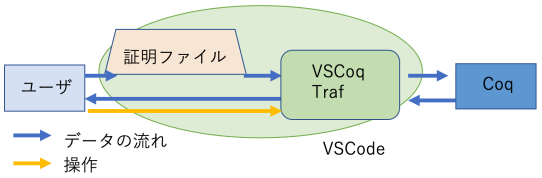

図 2: 設計を検討する新しい証明木図示ツールの構造図

本研究では、図 2 の VSCoq に組み込むための証明木図示ツール実現するため，手始めに実験的に VSCode のプラグインとして実現した VSCode 版 Traf の実装を行った．VSCode 版 Traf は将来的に VSCoq にマージを行う予定である．以下，VSCode 版 Traf の実装について述べる．

3.3 VSCode 版 Traf の概要

作成した VSCode プラグインとしての Traf の構造を図 3 に示す。開発を検討しているツールでは VSCoq が Coq とのやりとりを行うが、(図 4) 今回作成した VSCode 版 Traf では PG の標準出力をファイに出力し、それを読むことで代用した。また、開発言語には TypeScript を使用した。

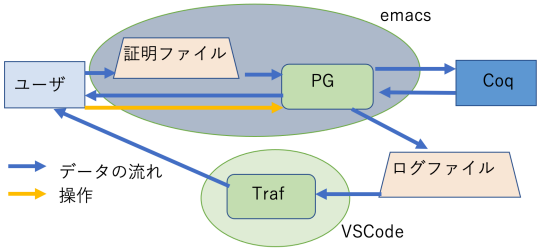

図 3: VSCode 版 Traf の構造図

3.3.1 証明木データ構造の構築

VSCode プラグインである Traf が、外部コマンドである tail コマンドを実行して (図 4) ログファイルを監視する。ログファイルは証明ステップが前後する度に更新され、Traf が内部情報を取得する。取得したデータ解析し、各証明ステップごとに証明木描画に必要な情報 (定理名、仮定、タクティク、サブゴール) を持ったノードを作成する。作成したノードを連結させることで証明木データ構造を構築する。

3.3.2 証明木の描画

VSCode 内に証明木を描画するために、Webview [8] と Canvas [9] を用いた。Webview は VSCode 内にタブの一つとして HTML を表示させる VSCode の API であり、Canvas は HTML 内に図を描画する HTML の要素である。Canvas では描画する図形の座標を指定する必要がある。

作成した証明木データ構造の大きさをもとに、証明木の根の部分である”証明する定理”の位置を決める。それ

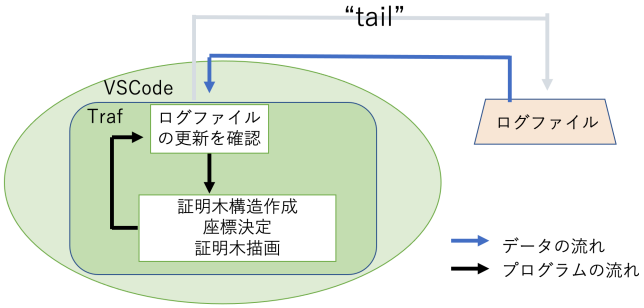

図 4: VSCode 版 Traf の処理の流れ

を基準として証明木全体の座標を決定し、証明木データ構造に組み込む。座標データを持った証明木データ構造を Canvas に渡し、証明木を描画する。

実際に VSCode 版 Traf で証明木を描画している様子を図 5 に示す。

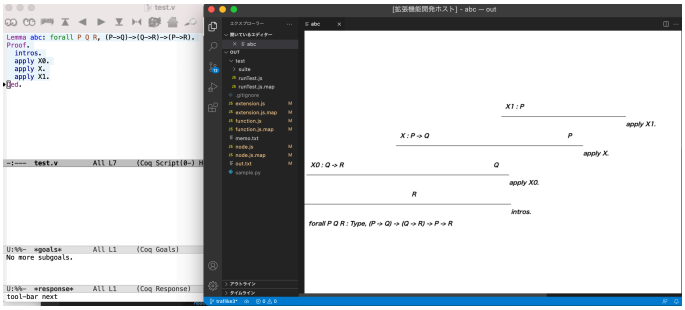

図 5: VSCode 内での証明木描画の様子

4 まとめと今後の課題

本研究では VSCode 内で証明木を描画する機能の大部分を作成することができた。これによって、あとは VSCoq との連携を行うことで VSCode 内での閉じた Coq 証明環境を開発することができる。

今後の課題としては VSCoq との連携すること、証明木にオンマウスした際に詳細情報を提示する機能の実装が挙げられる。

参考文献

[1] The Coq Proof Assistant. https://coq.inria.fr/
[2] Proof General. http://proofgeneral.github.io/
[3] Traf. https://github.com/hide-kawabata/traf
[4] Visual Studio Code. https://code.visualstudio.com/
[5] Prooftree. http://askra.de/software/prooftree/
[6] 古谷ら，定理証明支援系 Coq と連携した証明木図示ツールにおける大域的および局所的な情報把握支援機能の改善, 第 82 回情処全大, 4K-01, pp.1-173–1-174, Mar. 2020.
[7] VSCoq. https://github.com/coq-community/vscoq
[8] WebViewAPI. https://code.visualstudio.com/api/extension-guides/webview/
[9] Canvas. https://html.spec.whatwg.org/multipage/canvas.html/

Copyright ©2022 Information Processing Society of Japan.
All Rights Reserved.