VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ STUDIJŲ PROGRAMA

Faktų paieška pagal šabloną TLA+ įrodymų sistemoje

Fact search by pattern in the TLA+ proof system

Kursinis darbas

Atliko:	4 kurso 1 grupės studentas
	Domantas Keturakis
Darbo vadovas:	Doc., Dr. Karolis Petrauskas

Vilnius – 2024

Turinys

Terminai

- ATP (Automated theorem proving) – TODO
- Proof construction – įrodymo rašymas (ranka), i.e.: dalis nuo `Proof` iki `Qed`.
- Proof search – ta dalis, kur ATP atlieka automatiškai.
- Proof synthesis – ???
- Proof script – viskas nuo `Theorem` iki `Qed`.
- Sequent – In mathematical logic, a sequent is a very general kind of conditional assertion. pvz.: $A_1, ..., A_m \vdash B_1, ..., B_n$. [DEFINITION STOLEN FROM WIKI]

Įvadas

Traciškai programų sistemų modeliai aprašomi tekstu ir vizualinėmis diagramomis (UML) – šie yra sunkiai patikrinami automatiniai įrankiais. **[TODO]**.

Formalūs metodai (angl. *formal methods*) yra programinės įrangos kūrimo metodai, kurie naudoja matematine logika pagrįstus modelius, tam kad būtų galima analizuoti ir įrodyti programinės įrangos modelio savybes.

TLA$^+$ yra formaliais metodais pagrįsta modeliavimo sistema, kuria galima patikrinti sistemų modelių savybęs, siekiant užtikrinti jų teisingumą.

Sėkmingi atvejai:

- (AWS) ... [New+15]
- Safety-crictical systems:
  - TAS [RP17]
  - A train status alert system [Sal+20]
  - Avation systems [Das+24]

Čia nežinau, gal per mažai faktiška: IDEs padidina prieinanumą ir gali padidini TLA+ naudojamumą. (Todėl verta/relevant tobulinti TLA+ LSP)

Search engines [probably irrelevant]:

- Isabelle [HK22],
- Loogle for theorems in Lean

Background

Basically, šitai: „In this section, we describe the fundamental technologies for this paper. We describe the proof assistant Coq, giving an overview of how it works, as well as describe Visual Studio Code (VS Code), the Language Server Protocol (LSP), and the concept of code completion.“ bet skirtą šitam darbui.

TLA+

Čia trumpai apie TLA+

- TLA in Isabelle: [Mer99], [GM11]
- TLA checksheet
- The TLA+ Toolbox [KLR19]
  - This paper by Kuppe et al presents the technical architecture of the toolbox
  - Nalabai paaiškina kaip vekia pats TLAMP.

TLAMP

paaiškinti kas yra ir kaip veikia TLAPM / TLAPS.

- TLA+ Proofs [Cou+12]
  - „We describe how to write TLA+ proofs and check them with TLAPS, the TLA+ Proof System.“

TLAMP <-> SMT

Encoding

- „Encoding TLA$^+$'s Set Theory for Automated Theorem Provers“ [Def23]
  - Very lonk phd dissertation
  - „In Chapter 4, we define a direct encoding of TLA+ into HOL. A higher-order logic is necessary to encode the second-order constructs of TLA+ and axiom schemas. For instance, set comprehension $\{x \in S : e\}$ is viewed as a second-order application $\mathrm{setst}(S, \lambda x : e)$, and the full schema of comprehension from ZF is encoded as a single axiom.“
- „Encoding TLA+ Proof Obligations Safely for SMT“ [Def25]
  - „To increase trust in TLAPS, we revisited the encoding of TLA+ for SMT, whose implementation had become too complex. Our approach is based on a first-order axiomatization with E-matching patterns. The new encoding is available with TLAPS and achieves performances similar to the previous version, despite its simpler design.“

Isabelle

The Isabelle/Isar Reference Manual

- TO READ: Automatic Proof and Disproof in Isabelle/HOL [BBN11]
- TO READ: Automatic verification of non-recursive algorithm of Hanoi Tower by using Isabelle Theorem Prover [XYX16]

Zenon

- Zenon: an Extensible Automated Theorem Prover Producing Checkable Proofs [BDD07]

    - „Zenon can directly generate Coq proofs (proof script or proof terms)“
    - „we present Zenon, an automated theorem prover for first or- der classical logic (with equality), based on the tableau method.“

Faktų paieška

Gal tinkami raktažodžiai: metaheuristic search, proof script synthesis, automated proof script synthesis

- TO READ: ProofCloud: A Proof Retrieval Engine for Verified Proofs in Higher Order Logic [Wan24]
    -

RocQ (Coq)

- [3110 Coq Tactics Cheatsheet]()

- „Expanding Coq With Type Aware Code Completion“ [DK23]
    - Kaip suprantu veikia tik su `rewrite` ir `apply` taktikomis: „For every file, at every location where the tactics apply and rewrite are used, request for a list of completion items.“
    - „Discovery is handled through the internal function that is also used by the Search command in Coq“
    - „For every file, at every location where the tactics `apply` and `rewrite` are used, request for a list of completion items.“
    - Realiai čia aprašo kaip geriausia sort'inti rezultatus, naudinga, bet čia pirma reikia turėti, ką sort'inti: „The only difference in how the algorithms are implemented is the sorting step.“

- TO READ: TacTok: semantics-aware proof synthesis [FBG20]
    - Machine learning
    - Gal turi gerų reference'ų: „Recent research has shown that the proof state can help predict the next step.“
    - „In this paper, we present TacTok, the first technique that attempts to fully automate proof script synthesis by modeling proof scripts using both the partial proof script written thus far and the semantics of the proof state“

- TO READ: Coq search by type inhibition [Cza20a] [probably irrelevant]

- TO READ: PRoofster: Automated Formal Verification [Agr+23]

- TO READ: Coq `Search` komanda

- Hammer for Coq: Automation for dependent type theory [CK18]

  „This paper is the main reference for the hammer tactic.“

    - „Practical proof search for Coq by type inhabitation“ [Cza20b]

      „This paper is the main reference for the sauto tactic.“

    - „A Shallow Embedding of Pure Type Systems into First-Order Logic“ [Cza18]

      „This paper proves soundness of a core version of the translation used by the hammer tactic.“

- Kiti Coq įrankiai:
    - CoqHammer
    - ASTactic [Irrelevant]

- Machine learning based

Jeigu kalbėsiu apie Rocq (dar žinomą kaip Coq) [Pau15]

Others

- Let AI/LLMs do it [Zho25] [outside the scope] Galimai naudinga:
  - „We present a novel approach to automated proof generation for the TLA+ Proof System (TLAPS) using Large Language Models (LLMs). Our method combines two key components: a sub-proof obligation generation phase that breaks down complex proof obligations into simpler sub-obligations, <...>“
- Isabelle/HOL `find_theorems`, `find_consts` komandos
- Pattern maching proofs in `Lean` & `Idris`.

Previous art/work

Dalykai, ant kurių galiu pagrįsti savo darbą.

Isabelle `find_theorems` ir `find_consts` komandos, kaip veikia Coq `Search` komanda. Kas ir kokie metodai iš kiekvieno buvo pasirinkti, paaiškinti kodėl pasirinkti.

Implementacija / kaip susijuniga su lsp ir TLAMP

Planas xuliganas:

vscode-lsp <-> tlamp_lsp <-> tlamp <-> Isabelle

Galutinis rezultatas, kaip veikia, kaip naudotis, kaip pritaikyti.

Pavyzdys

TLA+ pavyzdys, kuriame matosi kaip veikia faktų paieška.

Trūkumai

Rezultatai

Išvados

Notes

- **The E Theorem Prover (https://github.com/eprover/eprover/blob/master/DOC/E-3.1.html)**

  Kaip suprantu gan paprasta first-order logic theorem prover implementacija.

  1.2 versija turi ~157k C kodo eilučiu, tai tikriausiai ne tokia ir paprasta implementacija.

  Tikriausiai irrelevant

  - The CADE ATP System Competition (https://tptp.org/CASC/)
  - „E – a brainiac theorem prover“ [Sch02]

Šaltiniai

[New+15] C. Newcombe, T. Rath, F. Zhang, B. Munteanu, M. Brooker, ir M. Deardeuff, „How Amazon web services uses formal methods“, *Commun. ACM*, t. 58, nr. 4, p. 66–73, kovo 2015.

[RP17] S. Resch ir M. Paulitsch, „Using TLA+ in the Development of a Safety-Critical Fault-Tolerant Middleware“, *2017 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, 2017, p. 146–152.

[Sal+20] G. Salierno, S. Morvillo, L. Leonardi, ir G. Cabri, „Specification and verification of railway safety-critical systems using TLA+: A Case Study“, *2020 IEEE 29th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, 2020, p. 207–212.

[Das+24] M. Das *ir kt.*, „TLA+ Specification of Aviation System with Time Analysis“, *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2024, p. 1–6.

[HK22] F. Huch ir A. Krauss, „FindFacts: A Scalable Theorem Search“. arXiv, 2022 m.

[Mer99] S. Merz, „An Encoding of TLA in Isabelle“, 1999.

[GM11] G. Grov ir S. Merz, „A Definitional Encoding of TLA* in Isabelle/HOL“, *Arch. Formal Proofs*, t. 2011, 2011.

[KLR19] M. A. Kuppe, L. Lamport, ir D. Ricketts, „The TLA+ Toolbox“, 2019.

[Cou+12] D. Cousineau, D. Doligez, L. Lamport, S. Merz, D. Ricketts, ir H. Vanzetto, „TLA+ proofs“, *FM 2012: Formal Methods: 18th International Symposium, Paris, France, August 27-31, 2012. Proceedings 18*, 2012, p. 147–154. [Interaktyvus]. Adresas: https://arxiv.org/pdf/1208.5933

[Def23] R. Defourné, „Encoding TLA$^+$'s Set Theory for Automated Theorem Provers“, 2023. [Interaktyvus]. Adresas: http://www.theses.fr/2023LORR0263/document

[Def25] R. Defourné, „Encoding TLA+ proof obligations safely for SMT“, *Science of Computer Programming*, t. 239, p. 103178, 2025, doi: https://doi.org/10.1016/j.scico.2024.103178.

[BBN11] J. C. Blanchette, L. Bulwahn, ir T. Nipkow, „Automatic proof and disproof in Isabelle/HOL“, *Frontiers of Combining Systems: 8th International Symposium, FroCoS 2011, Saarbrücken, Germany, October 5-7, 2011. Proceedings 8*, 2011, p. 12–27.

[XYX16] H. Xu, Z. You, ir J. Xue, „Automatic verification of non-recursive algorithm of Hanoi Tower by using Isabelle Theorem Prover“, *2016 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 2016, p. 13–18.

[BDD07] R. Bonichon, D. Delahaye, ir D. Doligez, „Zenon: An Extensible Automated Theorem Prover Producing Checkable Proofs“, *Logic for Programming, Artificial Intelligence, and Reasoning*, N. Dershowitz ir A. Voronkov, Sud., Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, p. 151–165.

[Wan24] S. Wang, „ProofCloud: A Proof Retrieval Engine for Verified Proofs in Higher Order Logic“, *arXiv preprint arXiv:2412.20947*, 2024.

[DK23] Hjalte Dalland Jakob Israelsen ir S. Kristensen, „Expanding Coq With Type Aware Code Completion“. 2023 m.

[FBG20] E. First, Y. Brun, ir A. Guha, „TacTok: semantics-aware proof synthesis“, *Proc. ACM Program. Lang.*, t. 4, nr. OOPSLA, lapkr. 2020.

[Cza20] Ł. Czajka, „Practical Proof Search for Coq by Type Inhabitation“, *Automated Reasoning*, N. Peltier ir V. Sofronie-Stokkermans, Sud., Cham: Springer International Publishing, 2020a, p. 28–57.

[Agr+23] A. Agrawal *ir kt.*, „Proofster: Automated formal verification“, *2023 IEEE/ACM 45th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, 2023, p. 26–30.

[CK18] Ł. Czajka ir C. Kaliszyk, „Hammer for Coq: Automation for dependent type theory“, *Journal of automated reasoning*, t. 61, p. 423–453, 2018.

[Cza20] Ł. Czajka, „Practical proof search for coq by type inhabitation“, *Automated Reasoning: 10th International Joint Conference, IJCAR 2020, Paris, France, July 1–4, 2020, Proceedings, Part II 10*, 2020b, p. 28–57.

[Cza18] L. Czajka, „A Shallow Embedding of Pure Type Systems into First-Order Logic“, *22nd International Conference on Types for Proofs and Programs (TYPES 2016)*, S. Ghilezan, H. Geuvers, ir J. Ivetic, Sud., Leibniz International Proceedings in Informatics (LIPIcs), vol. 97. Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018, p. 1–39.

[Pau15] C. Paulin-Mohring, „Introduction to the calculus of inductive constructions“, *All about Proofs, Proofs for All*, t. 55, 2015.

[Zho25] Y. Zhou, „Retrieval-Augmented TLAPS Proof Generation with Large Language Models“. 2025 m.

[Sch02] S. Schulz, „E–a brainiac theorem prover“, *Ai Communications*, t. 15, nr. 2–3, p. 111–126, 2002.