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

Įvadas

.

Traciškai programų sistemų modeliai aprašomi tekstu ir vizualinėmis diagramomis – sunkiai patikrinami automatiniai įrankiais, t.t. **[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

TLA+

Čia trumpai apie TLA+

TLA in Isabelle:
- [Mer99], [GM11]

TLAMP

Something something, čia paaiškinti kas yra ir kaip veikia TLAPM / TLAPS.

Faktų paieška

Theorem search:
- Isabelle/HOL `find_theorems`, `find_consts` komandos
- Coq `Search` komanda

LSP stuff:
- Coq auto-completion [DK23] [probably very relevant]
- Coq search by type inhibition [Cza20] [probably irrelevant]

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 <-> Isabelle

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

Pavyzdys

TLA+ pavyzdys, kuriame matosi kaip veikia faktų paieška.

Trūkumai

Rezultatai

Išvados

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, doi: 10.1145/2699417.

[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. doi: 10.1109/ISSREW.2017.43.

[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. doi: 10.1109/WETICE49692.2020.00048.

[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. doi: 10.1109/ICCCNT61001.2024.10725489.

[HK22] F. Huch ir A. Krauss, „FindFacts: A Scalable Theorem Search“. [Interaktyvus]. Adresas: https://arxiv.org/abs/2204.14191

[Mer99] S. Merz, „An Encoding of TLA in Isabelle“, 1999. [Interaktyvus]. Adresas: https://api.semanticscholar.org/CorpusID:10435651

[GM11] G. Grov ir S. Merz, „A Definitional Encoding of TLA* in Isabelle/HOL“, *Arch. Formal Proofs*, t. 2011, 2011, [Interaktyvus]. Adresas: https://api.semanticscholar.org/CorpusID:7524763

[DK23] Hjalte Dalland Jakob Israelsen ir S. Kristensen, „Expanding Coq With Type Aware Code Completion“. [Interaktyvus]. Adresas: https://github.com/Jakobis/vscoqComparison/blob/a181fe074cdca4ee0b13c2c71ab907bea73a5432/Expanding_Coq_with_Type_Aware_Code_Completion.pdf

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