High-assurance refactoring project
Interactive proofs in matching logic

Horpácsi Dániel *et al.*

TKP workshop, 18 Jan 2023

Publication results since 2022Q3

Accepted or appeared

Under submission or review

CORE ERLANG

Simon Thompson, **DH** : **Refactoring = Substitution + Rewriting**
Open Access Series in Informatics (OASIcs)

Balázs Szalontai, Péter Bereczky, **DH** : **Deep learning based refactoring with formally verified training data**
Annales Mathematicae et Informaticae

Péter Bereczky, **DH**, Simon Thompson : **A Comparison of Big-step Semantics Definition Styles**
Annales Universitatis Scientiarium Budapestinensis de Rolando Eötvös Nominatae

DH, Péter Bereczky, Simon Thompson : **Program Equivalence in an Untyped** **Language with Uncurried Functions**
Journal of Logical and Algebraic Methods in Programming (JLAMP)

Péter Bereczky, **DH**, Simon Thompson : **A Formalisation of Core Erlang, a C**
Acta Cybernetica

An alternative metatheory for semantics and refactoring

LOGIC

Péter Bereczky, Xiaohong Chen, **DH**, Lucas Peña, Jan Tušil : **Mechanizing Matching Logic In Coq**
Electronic Proceedings in Theoretical Computer Science (EPTCS)

Jan Tusil, Péter Bereczky, **DH** : **Interactive Matching Logic Proofs in Coq.**

Matching Logic in Coq

What is the right level of embedding for the matching logic in Coq?

Shallow or deep embedding? How to represent binders?

Can we prove the soundness of the matching logic proof system in Coq?

Including the fixed-point reasoning rules.

How to encode theories so that they are modular and practically usable?

Standard theories like definedness and sorts must be easily included.

What is the way of supporting matching logic reasoning within Coq?

We need ML-specific proof tactics and tautology solver.

How to import K semantics definitions into the Coq formalization?

This is needed for our ultimate goal of reasoning about PLs.

Proofs in matching logic: $A \rightarrow B \rightarrow A \wedge B$

```
Lemma ex1_pm2 : forall {Σ : Signature} (Γ : Theory) (A B : Pattern),
  well_formed A = true ->
  well_formed B = true ->
  Γ ⊢i (A ---> B ---> (A and B))
  using BasicReasoning.
Proof.
  intros Σ Γ A B WFA WFB.
  do 2 mlIntro; mlSplitAnd; mlAssumption.
Defined.
```

```
  mlAssert ("H0" : B). { wf_auto2. } { mlAssumption. }
  mlRevertLast.
  mlApply "H".
  mlIntro "H0". mlApply "H0". mlAssumption.
Defined.
```

```
                      (MP t1
                                  (P2 Γ B ((!(!A) ---> !B) ---> !A)
                                      ((((!(!A) ---> !B) ---> A) ---> !(!(!A) ---> !B)) _ _ _))).
  epose proof (tA := (P1 Γ A B) _ _).
  epose proof (tB' := MP tB (P1 _ (B ---> B) A _ _)).
  epose proof (t3' := MP t3'' (P2 _ B A ((!(!A) ---> !B) ---> A) _ _ _)).
  epose proof (t3 := MP t3' (P1 _ ((B ---> A) ---> B ---> (! (! A) ---> ! B) ---> A) A _ _)).
  epose proof (t5' := MP t5''
                         (P2 _ B ((!(!A) ---> !B) ---> A) (!(!(!A) ---> !B)) _ _ _)).
  epose proof (t5 := MP t5'
                        (P1 _ ((B ---> (! (! A) ---> ! B) ---> A) ---> B ---> ! (! (! A) ---> ! B))
                           A _ _)).
  epose proof (t6 := MP tA
                        (MP t3
                                    (P2 _ A (B ---> A) (B ---> (!(!A) ---> !B) ---> A) _ _ _))).
  epose proof (t7 := MP t6
                        (MP t5
                                    (P2 _ A (B ---> (!(!A) ---> !B) ---> A) (B ---> !(!(!A) ---> !B)) _ _ _))).
  apply t7.
  Unshelve.
    (* 43 well-formedness goals *)
    all: wf_auto2.
Defined.
```

- Automatic handling of well-formedness constraints on patterns
- Derived proof rules for natural deduction proofs
- Customized proof tactics for interactive reasoning in matching logic
- Object-level (embedded) proof state with named local (virtual) hypotheses (“virtual deduction theorem”)

Sequent natural deduction calculus for AML

- Single-conclusion sequents with local hypotheses and subproof constraints:

$$\Gamma \blacktriangleright_c \Delta \vdash_{\mathcal{N}} \psi$$

- Correspondence to the (default) Hilbert-style calculus:

$$\Gamma \blacktriangleright_c \varphi_1, \ldots, \varphi_k \vdash_{\mathcal{N}} \psi \iff \Gamma \vdash^c_{\mathcal{H}} \varphi_1 \to \ldots \to \varphi_k \to \psi$$

$$\Gamma \vdash_{\mathcal{N}} \psi \iff \Gamma \vdash_{\mathcal{H}} \psi$$

- The virtual deduction theorem (describing the “intro” tactic formally):

$$\frac{\Gamma \blacktriangleright_c \Delta, \varphi \vdash_{\mathcal{N}} \psi}{\Gamma \blacktriangleright_c \Delta \vdash_{\mathcal{N}} \varphi \to \psi}$$

And a lot more… we are now working on rules for FO and FIX reasoning.

/harp-project

Thank you for your attention!

Contact:

daniel-h@elte.hu

Az Alkalmazásiterület-specifikus nagy megbízhatóságú informatikai megoldások című projekt a Nemzeti Kutatási Fejlesztési és Innovációs Alapból biztosított támogatással, a Tématerületi kiválósági program (TKP2020-NKA-06, Nemzeti Kihívások Alprogram) finanszírozásában valósult meg.

NATIONAL RESEARCH, DEVELOPMENT AND INNOVATION OFFICE HUNGARY

PROGRAM FINANCED FROM THE NRDI FUND