High-assurance refactoring via machine-checked formalization

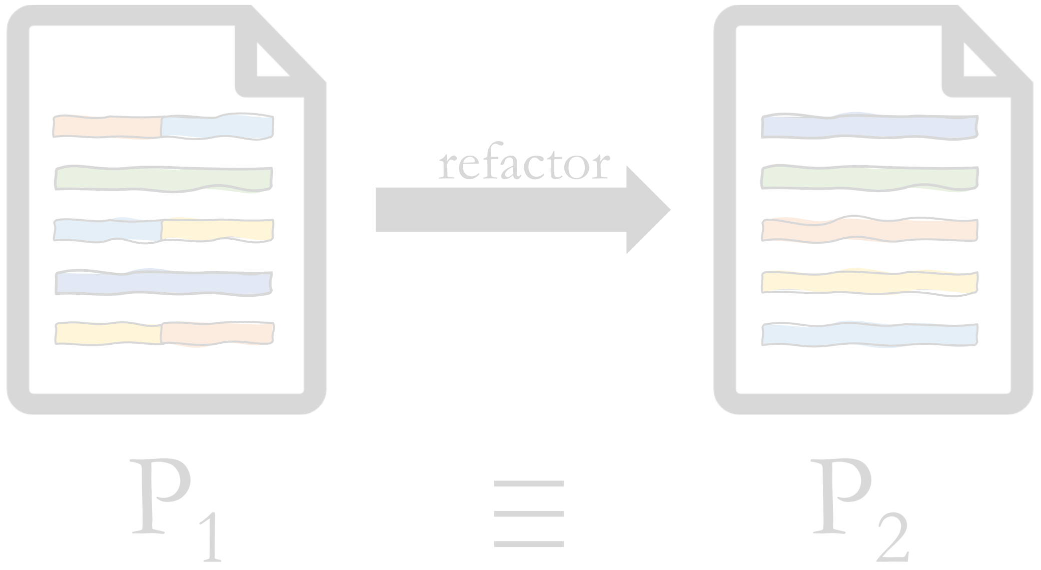
Horpácsi Dániel *et al.*

NATIONAL RESEARCH, DEVELOPMENT AND INNOVATION OFFICE HUNGARY

PROGRAM FINANCED FROM THE NRDI FUND

TKP workshop, 26 May 2022

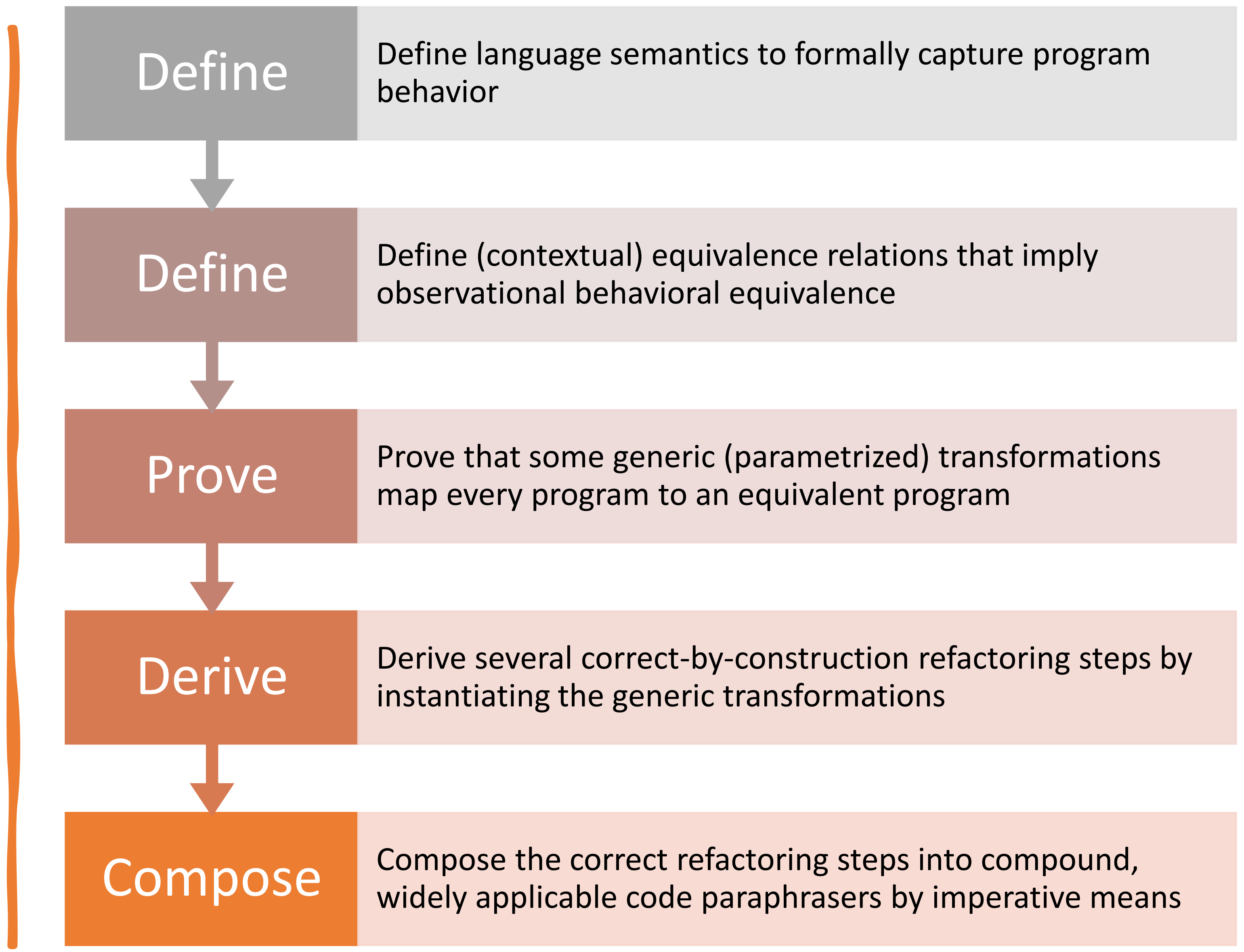
Gentle reminder: refactoring correctness



$$P_1 \equiv P_2$$

$$\forall P: (P \overset{?}{\equiv} \text{refactor}(P))$$

Our approach in a nutshell



Define	Define language semantics to formally capture program behavior
Define	Define (contextual) equivalence relations that imply observational behavioral equivalence
Prove	Prove that some generic (parametrized) transformations map every program to an equivalent program
Derive	Derive several correct-by-construction refactoring steps by instantiating the generic transformations
Compose	Compose the correct refactoring steps into compound, widely applicable code paraphrasers by imperative means

Our approach in a nutshell

Define	Define language semantics to formally capture program behavior
Define	Define (contextual) equivalence relations that imply observational behavioral equivalence
Prove	Prove that some generic (parametrized) transformations map every program to an equivalent program
Derive	Derive several correct-by-construction refactoring steps by instantiating the generic transformations
Compose	Compose the correct refactoring steps into compound, widely applicable code paraphrasers by imperative means

Formalizing the Erlang programming language

Is there a reasonably complex sublanguage that is representative?

recursion / pattern matching / container types / exceptions / side-effects

Which formal semantics definition style is the most suitable?

structural / reduction / natural / pretty big-step / denotational

What is the best way of encoding the formal language definition in Coq?

fully named / locally nameless / de Bruijn; scoping metatheory; value representation

How to validate the semantics against the reference implementation?

Extraction; property-based testing with shrinking

How to define a practically useful equivalence relation for the language?

behavioral / contextual / CIU / logical relations

How to expand the preliminary results to the full-featured language?

Matching Logic and its theories 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.

Further topics and future work

Further projects

Modern interfaces for interactive refactoring
aka. the Wrangler Language Server

AI-based Erlang refactoring
with formally verified training data

Future work

Complete formal definition for Erlang in Coq
including concurrent language features

Prove Erlang refactoring schemes correct
and derive some practically useful steps

Results

A successful Software Technology Lab

Bajka Ákos • Bense Viktor • Bereczky Péter • Boros Attila • Xiaohong Chen • Horpácsi Dániel • Horváth Szilárd • Katkó Dominik • Kókai Péter • Kőszegi Judit • Mizsei Tamás • Németh Dávid • Lucas Peña • Piszkor Balázs • Sághi György • Sevella Márton • Simon Thompson • Szalay Bence • Szalontai Balázs • Jan Tušil • Vadász András • Zászlós Márton

Dissemination

- 10 public repositories @ https://github.com/harp-project
- 8 published papers + 3 under review
- 4 theses accepted to be presented at the OTDK

Funding

- €100k grant from Runtime Verification Inc. for matching logic research

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