haskell-tools: A Haskell refactoring framework

Boldizsár Németh

Eötvös Loránd University, Budapest

nboldi@elte.hu

August 7, 2016

Introduction

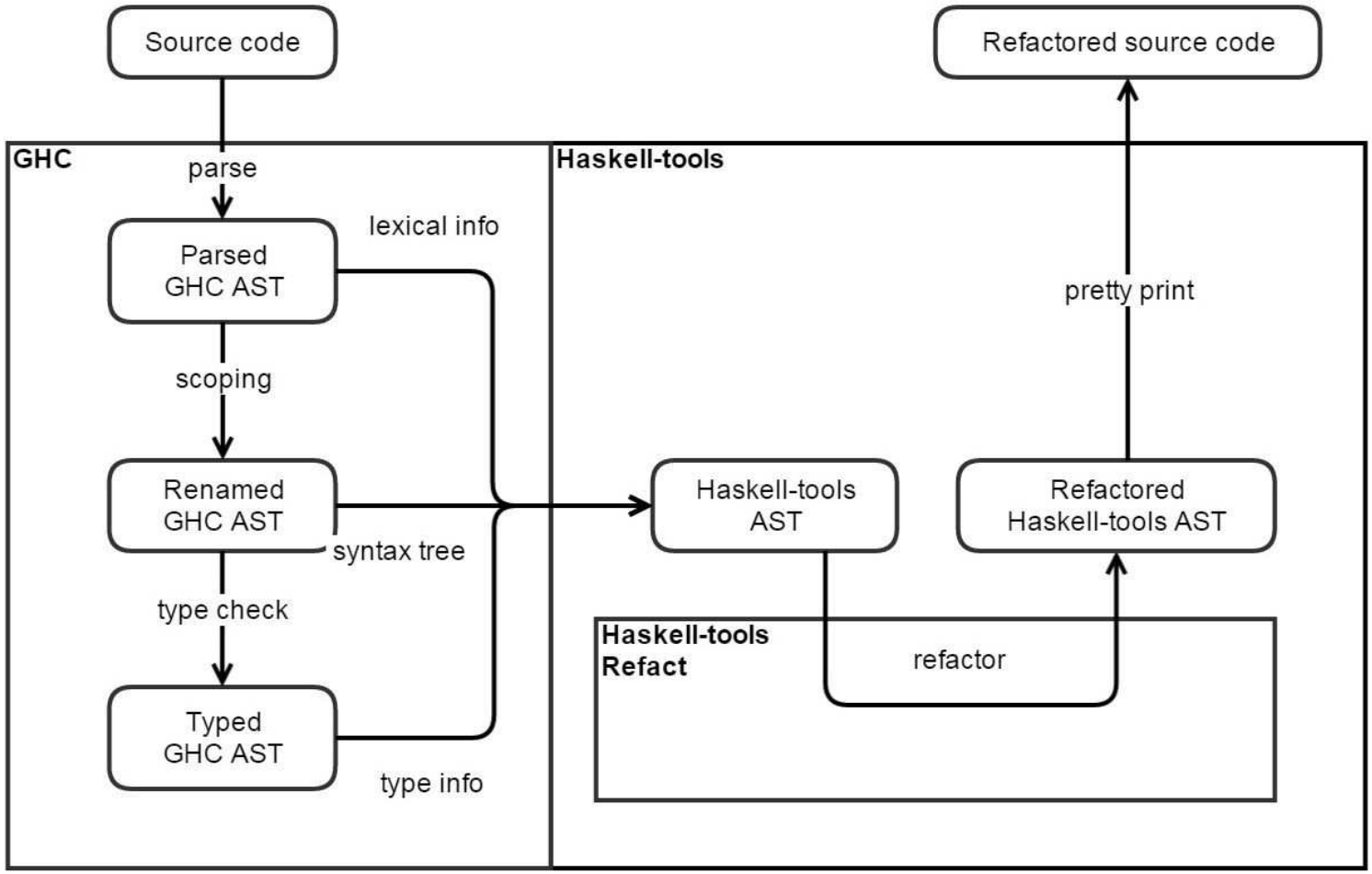
Haskell

- Novel and constantly evolving language
- High abstraction level
- Tooling support could be better

Approaches to refactoring

- Analyze the AST, perform rewrite on source (tokens/text). Pro: fast, keeps layout. Con: Rewrites can interfere, harder to write refactorings.
- Transform the AST, synchronize with the source. Current version of HaRe uses this.
- Transform the AST, pretty print it. Pro: easy to write refactorings. Con: no default mechanizm for keeping formatting.

Keeping source information

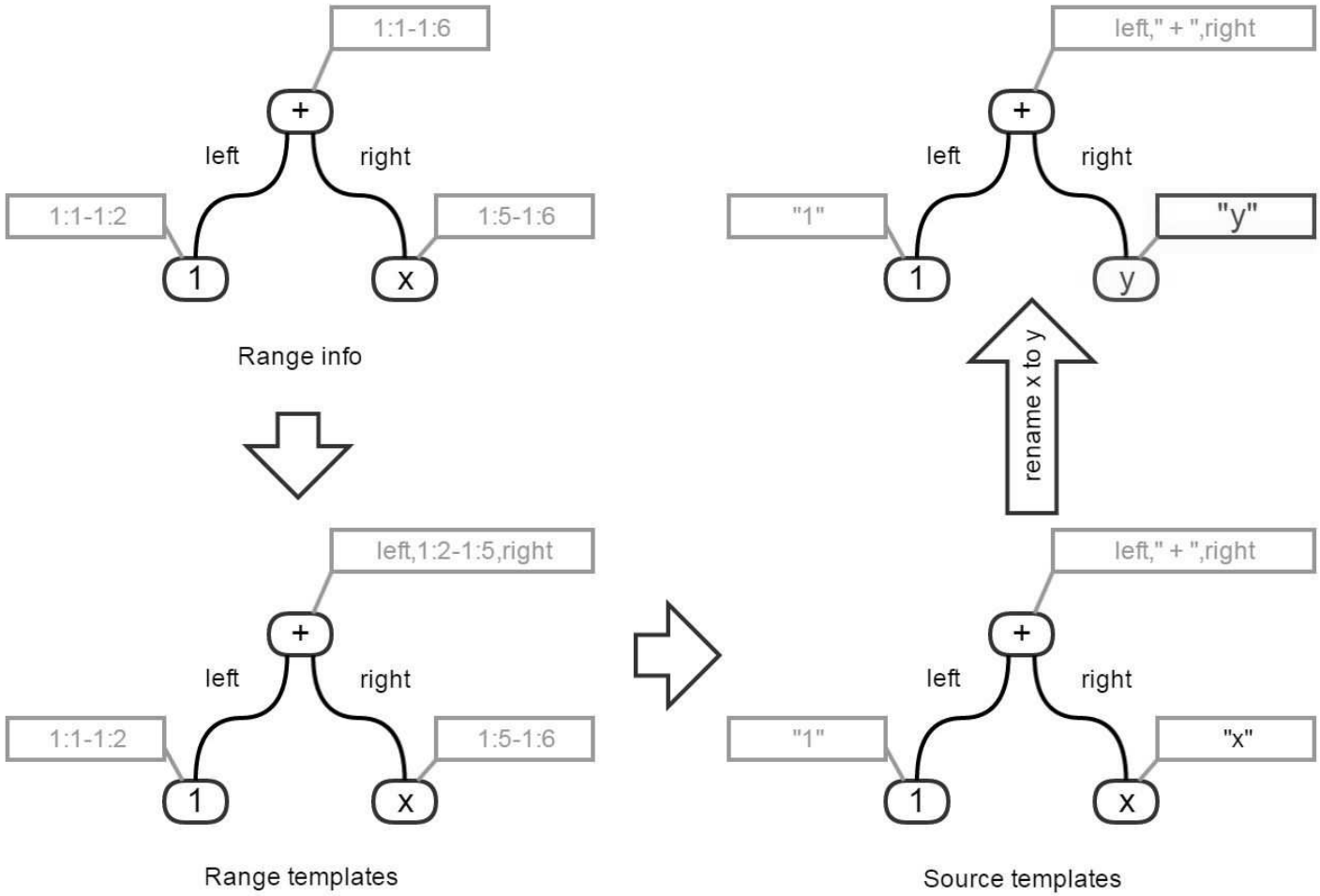


Basic idea

How to write a refactoring on AST

- Use GHC to compile the source file and get the AST.
- Convert GHC's AST to another AST representation better suited for rewriting.
- Store semantic infos and the original format of the source code in the new AST.
- Perform the refactorings as an `AST -> AST` function (monadic).

Keeping source information



Support refactorings

Our goal is to maximize the support of refactorings. Currently major refactorings can be implemented in 100-200 lines.

- Combine the information to a single representation
- A representation suited for transformation
- Semantic infos on nodes
- References (a lens-like property abstraction) for accessing elements.
- Uniplate generics
- Traversals

Current status

- GHC 8.0.1 support
- 5 complex refactorings implemented
- All comments and formatting kept
- Layout sensitive

Implemented refactorings

- Extract binding
- Rename
- Generate type signature
- Organize imports
- Generate exports

Future plans

- 2016 - Refactorings with multiple modules. Start supporting editor integration.
- early 2017 - Editor support and a wider variety of refactorings.

Planned refactorings

- Reorder arguments
- Move bindings up-down
- Move definition between modules
- Organize extensions
- Integration with hlint to replace parts of the source with better variants
- Quick fixes for common programming mistakes

Demo

haskelltools.org