



Goal:

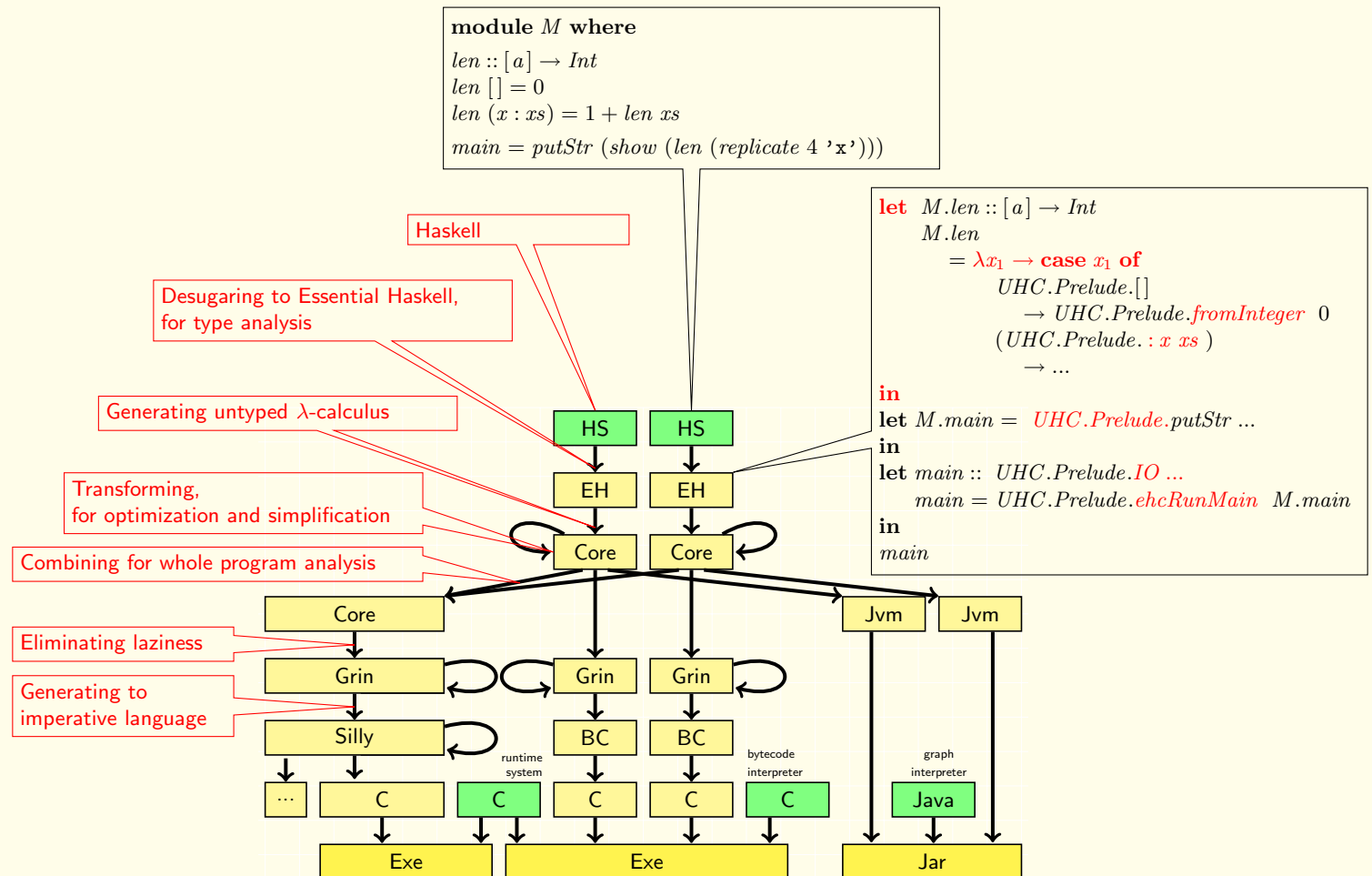
Experimentation platform for
Haskell language and compiler

Design:

Organize by partitioning:

Transformations, Aspects, Domain Specific Languages, and Tools

UHC pipeline: *languages and transformations*



Specification paradigms:

- Functional programming
GHC: Haskell compiler
- Tree-oriented programming
UUAG: Attribute Grammar preprocessor
- Rule-oriented programming
Ruler: Type rule preprocessor
- Aspect-oriented programming
Shuffle: Source fragmentation preprocessor

Transformation examples:

>60 *small & simple transformations*

- Core: η -reduction, renaming to unique identifiers, elimination of unnecessary recursive lets, elimination of name aliases, λ -lifting, ...
- Grin (whole program):
 - remove unused bindings, inline function bodies,
 - heap points to analysis,
 - inline evaluation and application, eliminate unused case alternatives, eliminate single case alternatives, merge case expressions, ...